Hewlett Packard Enterprise

HPE Ezmeral Runtime Enterprise 5.6 Documentation

Contents

last-updated: May 06, 2024

last-updated: May 06, 2024

last-updated: May 06, 2024

last-updated: May 06, 2024

last-updated: May 06, 2024

last-updated: May 06, 2024

last-updated: May 06, 2024

last-updated: May 06, 2024

Home

This site contains documentation for HPE Ezmeral Runtime Enterprise, including installation, configuration, administration, and reference content, and information about related solutions. Examples of related solutions include HPE Ezmeral ML Ops and HPE Ezmeral Runtime Analytics for Apache Spark.

Welcome!

HPE Ezmeral Runtime Enterprise is a unified platform built on open-source Kubernetes and designed for both cloud-native applications and non-cloud-native applications running on any infrastructure; whether on-premises, in multiple public clouds, in a hybrid model, or at the edge.

Transitioning to a container-first approach allows your organization to realize the agility and efficiencies of containerized applications running on either bare-metal or virtualized infrastructure.

Some of the key features of HPE Ezmeral Runtime Enterprise include:

- **Multi-cluster Kubernetes management:** Fast, easy deployment, management, and monitoring of Kubernetes clusters with out-of-the-box configuration of networking, load balancing, and storage.
- **100% open source Kubernetes:** With added innovations, such as the open-source KubeDirector Kubernetes-based controller, to deploy non-cloud-native apps.
- **Accelerate large-scale deployments with containers:** Speed and simplify your container deployment and operations at scale. Best practices and automation help streamline operations and improve SLAs. Hewlett Packard Enterprise delivers highly automated playbooks for Day 0 deployments combined with best practices and configuration automation to setup container HA, backup/restore, security validation and monitoring to minimize manual overheads for customers.
- **Enterprise-grade security and control:** Integrations into enterprise security and authentication services with support for high availability, fault tolerance, and resiliency for mission-critical enterprise applications.
- **On-demand provisioning:** App Store of curated, prebuilt images for a wide range of applications including machine learning (ML), analytics, IoT/edge, CI/CD, and application modernization. Flexibility to deploy on bare-metal or virtualized infrastructure, either on-premises, in the cloud, or at the edge.

Release Notes

The release notes contain information about new and changed features, installation, upgrade, compatibility, and issues and workarounds for HPE Ezmeral Runtime Enterprise 5.6.0.

Description

HPE Ezmeral Runtime Enterprise is a unified container software platform built on open source Kubernetes and designed for both cloud-native applications and non-cloud-native applications running on any infrastructure: on-premises, in multiple public clouds, in a hybrid model, or at the edge.

Supersede Information

Supersedes: The HPE Ezmeral Runtime Enterprise 5.6.0 release supersedes all HPE Ezmeral Runtime Enterprise 5.5.x releases.

last-updated: May 06, 2024

Operating Systems

This release is supported on the operating systems listed in OS Support on page 85

Languages

Languages supported for this release: English

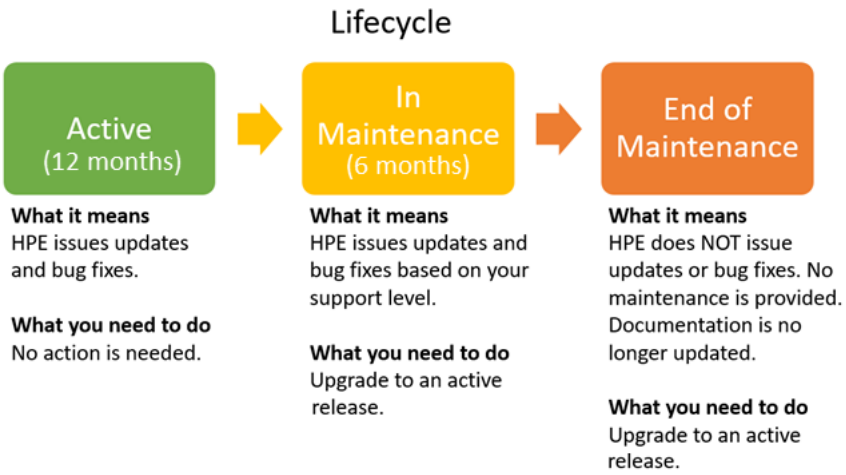
Understand the Software Lifecycle

This page describes the HPE Ezmeral Runtime Enterprise lifecycle and defines the lifecycle stages, which are Active, In Maintenance, and End of Maintenance.

Lifecycle Stages

Hewlett Packard Enterprise periodically releases new software. Each HPE Ezmeral Runtime Enterprise release is supported for an amount of time that can vary depending on the new releases that follow it. When new versions are released, older versions are deprecated or discontinued. Each version therefore has its own lifecycle. As shown in the diagram, a release can transition through three lifecycle stages:

- Active (12 months)
- In Maintenance (6 months)
- End of Maintenance



Typically, within six months after a new release, Hewlett Packard Enterprise issues an advisory to indicate the end of maintenance for older versions. Twelve months after the advisory is issued, the In-Maintenance version reaches the End-of-Maintenance stage and is discontinued.

To view the current lifecycle status for every release, see HPE Ezmeral Runtime Enterprise Version Support and Lifecycle Status on page 13. The following table describes the lifecycle stages:

Support and the Lifecycle Stages

Support Activity	Notes	Lifecycle Stage		
		Active	In Maintenance	End of Maintenance
Proactive Support (Minor, Maintenance)	Includes proactive fixes for security vulnerabilities, critical bugs, and other issues.	Yes	No	No

last-updated: May 06, 2024

Support Activity	Notes	Lifecycle Stage		
		Active	In Maintenance	End of Maintenance
Reactive Support (Escalation Support)	Requires the user to open cases resulting in tactical fixes for critical bugs, where backporting is feasible.	Yes	Yes[1]	No
Assisted Support (Usage / Debug Support)	Does not include patch fixes.	Yes	Yes	No

[1] Includes fixes for critical bugs and CVEs reported to Support. Does not include documentation updates.

HPE Ezmeral Runtime Enterprise Versions

See Software Versions on page 97 for details.

In HPE Ezmeral Runtime Enterprise interfaces and documentation, versions are expressed as a dot-separated string of numbers having two or three places. Updates and bug fixes result in changes to the major, minor, and maintenance versions of a release:

Notification of Changes in Support for Released Versions

To notify users about changes in HPE Ezmeral Runtime Enterprise support, Hewlett Packard Enterprise issues periodic support advisories. When releases are deprecated or discontinued, users of those releases are encouraged to upgrade to newer versions.

For service advisories, see Support and Other Resources on page 75

HPE Ezmeral Runtime Enterprise Version Support and Lifecycle Status

This page shows the support and lifecycle status for all versions of HPE Ezmeral Runtime Enterprise software.

Whenever possible, upgrade to the latest version of HPE Ezmeral Runtime Enterprise so that you can take advantage of new features, usability enhancements, and defect repair. If your installed version is "in maintenance," you have a limited amount of time to plan and execute a HPE Ezmeral Runtime Enterprise version upgrade.

- For lifecycle information , see Understand the Software Lifecycle on page 12
- For compatibility and interoperability information, see Support Matrixes on page 54
- For supported operating systems, see OS Support on page 85.

Lifecycle and Maintenance Dates

IMPORTANT: Consider the following points for pre-5.5 ERE versions:

- HPE cannot guarantee the support for any pre-5.5 version of HPE Ezmeral Runtime Enterprise beyond the End-of-maintenance dates–listed in the following table. Kubernetes container images used in pre-5.5 ERE versions are no longer available from public repositories, and security vulnerabilities in those images cannot be addressed.
- HPE recommends upgrading to the latest General availability (GA) version of HPE Ezmeral Runtime Enterprise from any pre-5.5.0 HPE Ezmeral Runtime Enterprise versions. Contact the HPE Support team for any questions related to HPE Ezmeral Runtime Enterprise and Kubernetes support.

IMPORTANT: Before upgrading to HPE Ezmeral Runtime Enterprise 5.6.x, HPE Ezmeral Product and Engineering team recommends upgrading all pre-5.5.1 deployments to HPE Ezmeral Runtime Enterprise 5.5.1, and to perform EzKube migration for the pre-5.5.1 Kubernetes clusters.

Table

Release	Release Date	Lifecycle Status	In Maintenance	End of Maintenance
5.6.4	September 1, 2023	Active	September 1, 2024	March 1, 2025
5.6.2	June 5, 2023	Active	June 4, 2024	Dec 4, 2024
5.6.1	Mar 29, 2023	Active	Mar 28, 2024	Sept 28, 2024
5.6.0	Jan 26, 2023	In Maintenance	Jan 25, 2024	July 25, 2024
5.5.1	Dec 19, 2022	In Maintenance	Dec 18, 2023	June 18. 2024
5.5.0	NA	In Maintenance	NA	Dec 31, 2023
5.4.2	NA	In Maintenance	NA	• Dec 31, 2024 (For EPIC installations) • Dec 31, 2023 (For Kubernetes installations)
5.4.1 and 5.4.0	NA	In Maintenance	NA	Dec 31, 2023
5.3.x	NA	In Maintenance	NA	Dec 31, 2023
5.2.x	NA	In Maintenance	NA	Dec 31, 2023
5.1.1	NA	In Maintenance	NA	• June 30, 2024 (For EPIC installations) • Dec 31, 2023 (For Kubernetes installations)
5.1.1 or earlier (including *EPIC 4.x or 3.x*)	NA	In Maintenance	NA	Dec 31, 2023

Related reference

Understand the Software Lifecycle on page 12

last-updated: May 06, 2024

This page describes the HPE Ezmeral Runtime Enterprise lifecycle and defines the lifecycle stages, which are Active, In Maintenance, and End of Maintenance.

Software Versions on page 97

Enhancements

This topic refers to where you can find information about new and changed features and functions for this release.

For a list of the new features and enhancements in HPE Ezmeral Runtime Enterprise see What's New in Version 5.6.x on page 99

Issues and Workarounds

This topic describes issues and workarounds in version 5.6.x of HPE Ezmeral Runtime Enterprise.

This topic describes issues and workarounds in HPE Ezmeral Runtime Enterprise version 5.6.x.

Installation Issues (prior releases)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.1. Unless otherwise noted, these issues also apply to later releases.

EZCP-2639: Add-Ons missing after installing HPE Ezmeral Runtime Enterprise on a reused Controller or Shadow host

Symptom: After you uninstall HPE Ezmeral Runtime Enterprise from a Controller or Shadow host, and then reuse that host as a Controller or Shadow host in a new deployment, expected system add-ons are not displayed on the **Application Configurations** screen when creating or editing a Kubernetes cluster.

Cause: The uninstall process did not delete the `hpe-cp-manifest` RPM on the host. Consequently, during the installation of HPE Ezmeral Runtime Enterprise on the reused host, the correct manifest RPM is not installed.

Workaround: Manually reinstall the manifest by entering the following command:

```
yum reinstall hpe-cp-manifest
```

To reuse a host:

After you uninstall HPE Ezmeral Runtime Enterprise from a host that will be used in another deployment, if the host was a Primary Controller or Shadow Controller host, erase the `hpe-cp-manifest` RPM:

- If this host is running RHEL/CentOS, enter the following command:

  ```
  yum erase hpe-cp-manifest
  ```

- If this host is running SLES, enter the following command:

  ```
  zypper rm hpe-cp-manifest
  ```

Upgrade Issues (5.6.x)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.6.x. Unless otherwise noted, these issues also apply to later releases.

last-updated: May 06, 2024

EZCP-3854: After upgrading the HPE Ezmeral Runtime Enterprise platform version, then upgrading the Kubernetes add-on versions, then upgrading the Kubernetes version, some pods fail.

Symptom: Some pods fail after performing the following upgrades in order:

1. Upgrading the HPE Ezmeral Runtime Enterprise platform with **Settings** > **Updates** > **Update**.
2. Upgrading the required add-ons with the Kubernetes add-ons upgrade script.
3. Upgrading the Kubernetes version with **Clusters** > **Upgrade Kubernetes** > **Confirm Upgrade**.

Instead of entering **Running** state, some pods such as `kubeflow` and `airflow` fail. For example:

```
kubeflow
katib-mysql-5bf95ddfcc-gdvc4
                    0/1
ContainerCreating   0
76m
kubeflow
minio-6bdd6c645f-p7j4x
                    0/2
Init:0/2            0
76m
kubeflow
minio-console-747896b76-6ld4m
                    0/1
Init:0/1            0
76m
kubeflow
ml-pipeline-5766c8b8bf-db5cr
                    1/2
CrashLoopBackOff    19 (54s ago)
76m
```

Cause: This issue is caused by an incorrect list of namepsaces excluded from OPA Gatekeeper policy constraints.

Workaround: To correct this issue, add all namespaces in the global config to the list of excluded namespaces for OPA Gatekeeper.

Proceed as follows:

1. Use SSH to access the Kubernetes master node.

2. Run the following command to fix the `hpecp-global-config`:

```
kubectl -n hpecp patch hpecpconfig
hpecp-global-config --type=json -p
"[{'op':'replace','path':'/spec/
reservedNamespaceNames','value':
[default,ezmysql,hpecp-falco,isti
o-system,kubeflow-jobs,ezctl,gateke
eper-system,hpe-sparkoperator,hpe-s
torage,hpe-system,knative-eventing,
kubeflow,hpe-csi,hpe-secure,kube-no
de-lease,mapr-external-info,prism-n
s,hpe-externalclusterinfo,hpe-templ
ates-compute,hpecp-cert-manager,kub
eflow-operator,ezml-model-mgmt,airf
lowop-system,kd-spark,knative-servi
ng,kubernetes-dashboard,velero,auth
,kd-mlops,airflow-base,hpe-nodesvc,
hpecp-observability,kube-system,cer
t-manager,hpe-ldap,hpecp-bootstrap,
kiali-operator,kube-public,argocd,h
pe-nfscsi,hpecp,kd-apps,kubeflow-us
er-example-com]}]"
```

3. On the Kubernetes master node, create the following Python script:

```
// Python script to add
reservedNamespaceNames to excluded
list of gatekeeper config
import os,json

system_namespaces =
os.popen("kubectl get
hpecpconfig -n hpecp -o
jsonpath=\"{.items[0].spec.reserved
NamespaceNames}\"").read()

sna = json.loads(system_namespaces)

system_namespaces_array =
map(lambda x: str(x), sna)

patch_string = "kubectl patch
config config -n
gatekeeper-system --type=json -p
\"[{'op':'replace','path': '/spec/
match/0/excludedNamespaces',
'value': %s}]\""%
(list(set(system_namespaces_array))
)

os.popen(patch_string)
```

This script fetches `reservedNamespaceNames` from `hpecp.global.config` and appends it to the list of exclued namespaces.

4. Run the Python script:

```
# python <python-script-name>
```

For example:

```
# python
gatekeeper_update_excluded_namespac
es.py
```

Upgrade Issues (5.6.x)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.6.x. Unless otherwise noted, these issues also apply to later releases.

EZCP-3808: Kiali dashboard is not accessible, in Kubernetes 1.24.X or later versions, and HPE Ezmeral Runtime Enterprise 5.6.0 and earlier versions

Symptom: In earlier Kubernetes versions, when a service account was created, a token would be automatically created,.This token in the tenant namespace is required to access the kiali dashboard . In Kubernetes versions 1.24.x or later, this token does not get created automatically, and must be created manually with hpecp-agent.

Workaround: Create the service account token, by executing the following command:

```
kubectl apply -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: <token-name>
  namespace: <namespace>
  annotations:
    kubernetes.io/service-account.name:
<serviceaccount-name>
type: kubernetes.io/
service-account-token
EOF
```

Enter `<token-name>` and `<serviceaccount-name>` for the name of the kiali service account in the tenant namespace. The kiali service account name will be displayed as `<tenant-name>-kiali-service-account`

NOTE: The kiali pod will come up before you create this service account token. You must delete the kiali pod in the namespace after the service account token is created. Then, a new kaili pod gets created automatically, and that pod uses the service token.

EZESC-1521: `bds-worker` on controller fails to start during upgrade from 5.3.6 to 5.5.1, or later version

Symptom: When you are upgrading from HPE Ezmeral Runtime Enterprise5.3.6 to 5.5.1, or later versions, upgrade may fail and rollback, as `dtap.ko.signed` gets deleted by new RPM.

In HPE Ezmeral Runtime Enterprise 5.5.1 or later versions, the RPM does not include `dtap.ko.signed` binary, and only includes `dtap.ko` binary. So if `dtap.ko.signed` is used before the

upgrade, perform the following workaround to solve the issue.

Workaround:

IMPORTANT: This workaround must be performed on each of the three controllers, and also on each of the workers.

On the primary controller, check if the dtap driver is loaded, by running `lsmod | grep` command, for example:

```
[root@mip-bd-vm134 ~]# lsmod | grep dtap
dtap 196679163 0
```

- If the output is empty, then this workaround is not needed.
- If the output shows that the dtap driver is loaded, check the log file `/var/log/bluedata/bds-worker.log` for **dtap.ko** binary, as follows:
  1. If `/sbin/insmod /opt/bluedata/common-install/data_server/drivers/`**`dtap.ko`** is available, this workaround is not needed.
  2. If `/sbin/insmod /opt/bluedata/common-install/data_server/drivers/`**`dtap.ko.signed`** is available, run the following command:

     ```
     bdconfig --set
     bdshared_install_nodtapdriver=tr
     ue
     ```

  3. Make sure that `/sbin/insmod /opt/bluedata/common-install/data_server/drivers/`**`dtap.ko`** is available in the `bds-worker.log` file.

EZCP-3742: After upgrading HPE Ezmeral Runtime Enterprise to a newer version, the Istio add-on appears in the UI but is not deployed, and edit actions on the cluster fail.

Symptom: After upgrading HPE Ezmeral Runtime Enterprise to a newer version, the Istio add-on appears as enabled in the HPE Ezmeral Runtime Enterprise UI, but is not deployed on the backend. Edit actions on the Kubernetes cluster fail until the cluster is submitted with the Istio add-on deployed.

Workaround: You must execute the Kubernetes add-ons upgrade script after upgrading HPE Ezmeral Runtime Enterprise to a newer version. See Kubernetes Add-On Upgrade Script on page 908.

Upgrade Issues (5.5.0)

EZML-2059: Upgrading a Kubernetes cluster with a Kubeflow add-on in HPE Ezmeral Runtime Enterprise might fail.

Symptom: If your Kubernetes cluster has an existing Kubeflow add-on, the Kubernetes cluster upgrade might fail with the following message in the platform

controlller logs within namespaces `kubeflow`, `knative-serving`, `knative-eventing`:

```
Cannot evict pod as it would violate
the pod's disruption budget
```

Cause: This issue is caused by attempting an upgrade on a Kubernetes cluster with a version of Kubeflow lower than 1.6 enabled.

Workaround:

1. Execute the following commands:

```
kubectl delete pdb -n
knative-serving --all
```

```
kubectl delete pdb -n
knative-eventing --all
```

```
kubectl delete pdb -n
kubeflow --all
```

2. Re-run the Kubernetes cluster upgrade, as described in Upgrading Kubernetes on page 487.

Upgrade Issues (5.4.x)

EZCP-2582: Upgrading Kubeflow on HPE Ezmeral Runtime Enterprise requires assistance.

If your environment includes Kubeflow and you are upgrading HPE Ezmeral Runtime Enterprise, contact Hewlett Packard Enterprise support for assistance before you begin the upgrade. Several manual steps must be performed to replace the existing version of Kubeflow with the new version of Kubeflow.

HPE Ezmeral Data Fabric on Kubernetes Issues (5.4.1)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.1. Unless otherwise noted, these issues also apply to later releases.

EZSPA-661: HPE Ezmeral Data Fabric on Kubernetes pods and Livy pods not able to resolve AD users

Symptom: HPE Ezmeral Data Fabric on Kubernetes pods and Livy pods could not submit any queries successfully. These queries will fail, if customer's AD/LDAP servers do not support TLS version 1.3. You might encounter error `key too small`.

Workaround: Contact Hewlett Packard Enterprise Technical support for assistance.

HPE Ezmeral Data Fabric on Kubernetes Issues (Prior Releases)

The following issues were identified in a version of HPE Ezmeral Runtime Enterprise prior to version 5.5.0. Unless otherwise noted, these issues also apply to later releases.

EZKDF-627: With HPE Ezmeral Data Fabric on Kubernetes version 1.5.0, If a cldb or mfs pod is deleted, `mrconfig` info instances may show an incorrect number of instances.

Symptom: With HPE Ezmeral Data Fabric on Kubernetes version 1.5.0, if you delete a `cldb` or `mfs`

pod, `mrconfig` info may show an incorrect number of instances.

Workaround: After the pod is restarted, and is in healthy state, restart MFS repeatedly up to three times, until it shows the correct number. For example, use the following commands to restart MFS upto three times:

```
sudo touch /opt/mapr/kubernetes/
maintenance
/opt/mapr/initscripts/mapr-mfs restart
/opt/mapr/initscripts/mapr-mfs restart
/opt/mapr/initscripts/mapr-mfs restart
sudo rm /opt/mapr/kubernetes/maintenance
To verify mfs instances count,
Run mrconfig info instances
```

EZKDF-710: With HPE Ezmeral Data Fabric on Kubernetes version 1.5.0, if a cldb or mfs pod is upgraded, `mrconfig` info instances may show an incorrect number of instances.

Symptom: With HPE Ezmeral Data Fabric on Kubernetes version 1.5.0, if you upgrade a `cldb` or `mfs` pod applying a new CR, and the change the cpu, memory and/or disk parameters, `mrconfig` info may show an incorrect number of instances.

Workaround: After the pod is restarted and is in healthy state, restart MFS repeatedly up to three times, till it shows the correct number. For example, use the following commands to restart MFS upto three times:

```
kubectl exec -it <mfs_pod> -n <cluster
namespace> bash
Within the mfs_pod or CLDB pod, execute
the following commands.
sudo touch /opt/mapr/kubernetes/
maintenance
/opt/mapr/initscripts/mapr-mfs restart
/opt/mapr/initscripts/mapr-mfs restart
/opt/mapr/initscripts/mapr-mfs restart
sudo rm /opt/mapr/kubernetes/maintenance
```

EZESC-563: ZooKeeper issue when running the `saveAsNewAPIHadoopFile` method on HPE Ezmeral Data Fabric on Kubernetes cluster.

Symptom: Running the `saveAsNewAPIHadoopFile` method on HPE Ezmeral Data Fabric on Kubernetes cluster generates the following error:

```
ERROR MapRZKRMFinderUtils: Unable to
determine ResourceManager service
address from Zookeeper at
xxx.xxx.xxx.xxx
```

Workaround: Set the `yarn.resourcemanager.ha.custom-ha-enabled` and `yarn.resourcemanager.recovery.enabled` property on `/opt/mapr/hadoop/hadoop-2.7.4/etc/hadoop/yarn-site.xml` configuration file to `false`.

EZKDF-109: After CLDB upgrade, MFS pods remain in a bad state.

Workaround: Use the following command to restart the MAST gateway:

```
kubectl exec -it -n
<namespace> <mfs-pod> -- /opt/mapr/
initscripts/mapr-mastgateway restart
```

EZKDF-404: Clusters that Implement HPE Ezmeral Data Fabric on Kubernetes fail to start after Kubernetes version or HPE Ezmeral Runtime Enterprise version upgrade.

The following advice applies to deployments that have separate Data Fabric clusters, and deployments that combine compute and Data Fabric nodes in the same cluster. This advice does not apply to deployments that implement Embedded Data Fabric only.

Attempts to upgrade or patch Kubernetes or upgrade HPE Ezmeral Runtime Enterprise in deployments that include HPE Ezmeral Data Fabric on Kubernetes can fail in ways that require a significant number of recovery steps.

Contact your Hewlett Packard Enterprise support representative for upgrade assistance for any of the following:

- Upgrading or patching the Kubernetes version on any cluster that implements **HPE Ezmeral Data Fabric on Kubernetes**.
- Upgrading **HPE Ezmeral Data Fabric on Kubernetes** independently of an upgrade to HPE Ezmeral Runtime Enterprise.
- Upgrading HPE Ezmeral Runtime Enterprise on deployments that implement HPE Ezmeral Data Fabric on Kubernetes.

If your environment deploys a version of HPE Ezmeral Runtime Enterprise prior to version 5.3.5, Hewlett Packard Enterprise recommends that you upgrade to HPE Ezmeral Runtime Enterprise 5.3.5 or later before you add **HPE Ezmeral Data Fabric on Kubernetes**.

EZESC-563: ZooKeeper issue when running the `saveAsNewAPIHadoopFile` method on HPE Ezmeral Data Fabric on Kubernetes cluster.

Symptom: Running the `saveAsNewAPIHadoopFile` method on HPE Ezmeral Data Fabric on Kubernetes cluster generates the following error:

```
ERROR MapRZKRMFinderUtils: Unable to
determine ResourceManager service
address from Zookeeper at
xxx.xxx.xxx.xxx
```

Workaround: Set the `yarn.resourcemanager.ha.custom-ha-enabled` and `yarn.resourcemanager.recovery.enabled` property on `/opt/mapr/hadoop/hadoop-2.7.4/etc/hadoop/yarn-site.xml` configuration file to `false`.

EZKDF-109: After CLDB upgrade, MFS pods remain in a bad state.

Workaround: Use the following command to restart the MAST gateway:

```
kubectl exec -it -n
<namespace> <mfs-pod> -- /opt/mapr/
initscripts/mapr-mastgateway restart
```

Open Policy Agent Issues (5.4.1)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.1. Unless otherwise noted, these issues also apply to later releases.

EZCP-2688: CSI drivers fail to install due to enforcement of the `psp-privileged-container` OPA policy.

Symptom: An attempt to install CSI drivers fails with the error `ReplicaFailure`, and gives the following message:

```
Error creating: admission webhook
"validation.gatekeeper.sh" denied the
request: [psp-privileged-container]
Privileged container is not allowed:
csi-provisioner, securityContext:
{"privileged": true}

[psp-privileged-container] Privileged
container is not allowed: direct-csi,
securityContext: {"privileged": true}
```

Workaround:

1. On the master node of the Kubernetes cluster, save the following Python script as `priv_constraint_update_excluded_namespaces.py`:

```
import json,os

csi_driver_system_namespace =
sys.argv[1]

system_namespaces =
os.popen("kubectl get
k8spspprivilegedcontainer.constrain
ts.gatekeeper.sh/
psp-privileged-container -o=jsonpat
h=\"{.spec.match.excludedNamespaces
}\"").read()

sna = json.loads(system_namespaces)

system_namespaces_array =
map(lambda x: str(x), sna)

system_namespaces_array.append(csi_
driver_system_namespace)

patch_string = "kubectl patch
k8spspprivilegedcontainer.constrain
ts.gatekeeper.sh/
psp-privileged-container --type=jso
n -p \"[{'op':'replace','path': '/
spec/match/excludedNamespaces',
'value': %s}]\""%
(list(set(system_namespaces_array))
)

os.popen(patch_string)
```

2. On the master node of the Kubernetes cluster, execute the script with the following command:

```
python
priv_constraint_update_excluded_nam
espaces.py <csi-driver-namespace>
```

`<csi-driver-namespace>` refers to the namespace in which you are creating the privileged container.

For example, if you want to create the `direct-csi` pod in the namespace `csi`, then execute:

```
python
priv_constraint_update_excluded_nam
espaces.py csi
```

Kubernetes Issues (5.6.1)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.6.1. Unless otherwise noted, these issues also apply to later releases.

EZCP-3936: A Kubernetes cluster's `kubelet` service fails, and `kubectl` commands stop working.

Symptom: A Kubernetes cluster on HPE Ezmeral Runtime Enterprise stops working correctly because its SSL certificates have expired. The `kubelet` service fails, and `kubectl` commands stop working.

Workaround: Follow the steps in this guide to renew the SSL certificates: Procedure for updating Kubernetes cluster certificates.

Kubernetes Issues (5.6.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.6.0. Unless otherwise noted, these issues also apply to later releases.

EZCP-3741: The log for a deployed Kubernetes cluster shows the errors `etcdserver: timed out` and `slow disk`.

Symptom: The log for a deployed Kubernetes cluster shows the errors `etcdserver: timed out` and `leader failed to send out heartbeat on time; took too long, leader is overloaded likely from slow disk`.

For example:

```
Feb 07 14:29:36 example.hpecorp.net
etcd[6249]:
{"level":"warn","ts":"2023-02-07T14:29
:36.553-0800","caller":"etcdserver/
server.go:1159","msg":"failed to
revoke
lease","lease-id":"7602862df1792313","
error":"etcdserver: request timed
out"}

Feb 07 14:29:36 example.hpecorp.net
etcd[6249]:
{"level":"warn","ts":"2023-02-07T14:29
:36.651-0800","caller":"v3rpc/
interceptor.go:197","msg":"request
stats","start
time":"2023-02-07T14:29:29.650-0800","
time
spent":"7.000923805s","remote":"127.0.
0.1:50504","response type":"/
etcdserverpb.KV/Txn","request
count":0,"request size":0,"response
count":0,"response size":0,"request
content":""}

Feb 07 14:29:39 example.hpecorp.net
etcd[6249]:
{"level":"warn","ts":"2023-02-07T14:29
:39.128-0800","caller":"etcdserver/
server.go:1159","msg":"failed to
revoke
lease","lease-id":"4c87862de93218b3","
error":"etcdserver: request timed
out"}
```

```
Feb 07 14:29:39 example.hpecorp.net
etcd[6249]:
{"level":"warn","ts":"2023-02-07T14:29
:39.295-0800","caller":"etcdserver/
raft.go:415","msg":"leader failed to
send out heartbeat on time; took too
long, leader is overloaded likely
from slow
disk","to":"973a665ee093f602","heartbe
at-interval":"100ms","expected-duratio
n":"200ms","exceeded-duration":"161.31
6838ms"}
```

In some cases, other errors may occur. For example, the Kubernetes cluster might fail to enter a **Ready** state, with the bootstrap log for `hpecp-bootstrap-prometheus` displaying the error `UPGRADE FAILED`:

```
[jenkins@mip-bd-ap07-n3-vm01 install]
$ kubectl logs
hpecp-bootstrap-prometheus-868c8b97d-h
gx65 -n hpecp-bootstrap
Wed Jan  4 04:40:02 UTC 2023:
Starting prometheus reconfigure
process
Error: UPGRADE FAILED: pre-upgrade
hooks failed: warning: Hook
pre-upgrade kube-prometheus-stack/
templates/prometheus-operator/
admission-webhooks/job-patch/
serviceaccount.yaml failed:
etcdserver: request timed out
failed to reconfigure helm chart
configmap/hpecp-bootstrap-prometheus
patched
```

Cause: This issue is caused by insufficent disk I/O when performing etcd operations. This issue can impact any add-on or pod running on a Kubernetes cluster that is also running high volume api-server operations simultaneously.

Workaround: To check whether your environment meets minimum disk speed requirements for etcd, you can run one of the etcd benchmark tools described in the official etcd documentation (link opens an external site in a new browser tab or window).

To ensure your environment has the required disk speed for etcd operations, Hewlett Packard Enterprise recommends using a solid state drive.

Kubernetes Issues (5.5.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.5.0. Unless otherwise noted, these issues also apply to later releases.

EZCP-3543: In a deployment that includes Embedded Data Fabric, deleting a Kubernetes cluster does not automatically delete the CSI state volume.

Symptom: In a deployment that uses Embedded Data Fabric, when you delete a Kubernetes cluster, the CSI state volume is not deleted automatically. The volume and a small (a few megabytes) file remain.

Workaround: After you delete the Kubernetes cluster, delete the CSI state volume manually. On the Controller, do the following:

1. Look for the following error message in `/var/log/bluedata/bds-mgmt.log`:

   ```
   got an error trying to
   delete snapshot state volume
   ("<cluster-id>")
   ```

   - If the log does not contain the error message, the volume was deleted successfully. No other actions are required.
   - If the log contains the error message, proceed to the next step.

2. From the log message, note the cluster ID of the deleted Kubernetes cluster.

   For example, in the following error message, the cluster ID is `10`:

   ```
   got an error trying to delete
   snapshot state volume("10")
   ```

3. Delete the CSI state volume by entering the following commands:

   ```
   /opt/bluedata/ezpylib/bluedata/
   mapr/bds-mapr-config.py
   deleteVolume --vol-name
   apps-k8s-<deleted-cluster-id>-k8s-c
   si-state
   /opt/bluedata/ezpylib/bluedata/
   mapr/bds-mapr-config.py
   deleteHadoopDir --dir-name /apps/
   k8s-<deleted-cluster-id>
   ```

   For example, if the cluster ID is `10` the commands you enter are the following:

   ```
   /opt/bluedata/ezpylib/bluedata/
   mapr/bds-mapr-config.py
   deleteVolume --vol-name
   apps-k8s-10-k8s-csi-state
   /opt/bluedata/ezpylib/bluedata/
   mapr/bds-mapr-config.py
   deleteHadoopDir --dir-name /apps/
   k8s-10
   ```

Kubernetes Issues (5.4.3)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.3. Unless otherwise noted, these issues also apply to later releases.

EZCP-3070: Falco pods are in a `CrashLoopBackOff` state due to an incompatible runtime schema version.

Symptom: Falco pods installed on HPE Ezmeral Runtime Enterprise are in a `CrashLoopBackOff`

state due to an incompatible runtime schema version. The pod logs show a `Runtime error`. For example:

```
Runtime error: Driver supports schema
version 2.0.0, but running version
needs 1.0.0.
```

Workaround:

1. Update the Falco kernel driver to the latest version.
2. Ensure the latest Falco pods are in the `hpecp-falco` namespace.
3. **If you are upgrading Falco pods**, you must use the latest `falco-no-driver` images. Download the latest images here (link opens an external site in a new browser tab or window).

Kubernetes Issues (5.4.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.0. Unless otherwise noted, these issues also apply to later releases.

EZCP-1925: When you delete a Kubernetes cluster, `iptable` settings do not get cleaned up on each associated Kubernetes host

Symptom: When you delete a Kubernetes cluster, `iptable` settings do not get cleaned up, on each associated Kubernetes host. Later, if you add these hosts to any other kubenetes cluster, `kubeproxy` uses the existing `iptable` rules to get routed to the appropriate pods. These `iptable` rules must be removed from the Kubernetes host, when you remove the host from the Kubernetes cluster.

Cause: As these existing `iptable` entries were not removed from the host, various networking routing problems may occur, when the same host is added to any other cluster.

Workaround:

You must manually delete `iptable` settings in the file. Contact Hewlett Packard Enterprise technical support to know how to delete the `iptable` settings.

EZCP-2036: Graphs on the Kubernetes Dashboard hanotherg on very large Kubernetes deployments.

Symptom: Graphs on the **Kubernetes Dashboard** fail to load when displaying information for large scale Kubernetes deployments (such as 1,000 nodes).

Workaround: None at this time.

EZCP-2097, EZESC-1103: Creating a Kubernetes cluster with optional add-ons enabled causes a delay in port service link readiness.

Symptom: Clicking the link for a service endpoint shows a `503 error`, or the links for service endpoints do not appear in the UI.

Cause: When creating Kubernetes clusters with the optional add-ons Istio, Kubeflow, Airflow, and Spark Operator, the HPE Ezmeral Runtime Enterprise gateway port mappings for Argo CD, Istio, Kubeflow, and Kiali NodePort services may take up to twenty minutes to become available after the cluster is ready.

Workaround:

- If the UI shows the links for the service endpoints, but clicking the link shows a `503 error`, then check that all pods are running and ready. Once the pods are in the ready state, they will become available.
- If the UI shows the cluster as ready, but the UI does not show the links for the service endpoints, then delete the `hpecp-agent` pod:

  ```
  kubectl -n hpecp delete pod $
  (kubectl -n hpecp get pod -l
  name=hpecp-agent -o
  jsonpath='{.items[0].metadata.name}
  ')
  ```

  The pod will be re-created once the cluster enters the ready state. The services gateway port mappings will immediately be created once the `hpecp-agent` pod is running.

Kubernetes Issues (Prior Releases)

The following issues were identified in a version of HPE Ezmeral Runtime Enterprise prior to version 5.4.0. Unless otherwise noted, these issues also apply to later releases.

EZESC-542: On the Kubernetes Application screen, clicking an ingress service endpoint link, such as for Istio, returns an HTTP or HTTPS error.

Symptom: On the **Service Endpoints** tab of the **Kubernetes Application** screen, endpoint links are displayed for Kubernetes ingress controllers, such as the Istio ingress gateway, but clicking the links return HTTP or HTTPS 503 errors that indicate the service is unavailable or a secure connection could not be made.

Cause: HPE Ezmeral Runtime Enterprise automatically configures ingress gateway service endpoints when an ingress gateway such as `istio-ingress` is configured on a Kubernetes cluster. However, for most Kubernetes applications, there is no corresponding service that is automatically configured, so there is no service available through the endpoint.

Workaround: None. Ignore the service endpoint links.

See also EZKDF-404 in HPE Ezmeral Data Fabric on Kubernetes Issues (Prior Releases) on page 20

EZKDF-404, "Clusters that Implement HPE Ezmeral Data Fabric on Kubernetes fail to start after Kubernetes version or HPE Ezmeral Runtime Enterprise version upgrade," also applies to upgrading Kubernetes versions in HPE Ezmeral Runtime Enterprise 5.3.5 deployments that implement HPE Ezmeral Data Fabric on Kubernetes.

EZCP-1608, EZCP-2306, and EZCP-2358: When an application (e.g. Istio or Airflow) is deployed in the Kubernetes cluster, one or more worker nodes fail to upgrade the Kubernetes version

Symptom: When an application (e.g. Istio or Airflow) is deployed in the Kubernetes cluster, one or more worker nodes fail to upgrade the Kubernetes version, with the following errors:

- `Warning: one or more workers failed to upgrade` on the **Kubernetes Cluster** screen.
- `Upgrade error: Failed to drain node` error at the individual **Kubernetes Host Status** screen

This issue also occurs when the application user deploys **PodDisruptionBudget (PDB)** objects to the application workloads. For more information about PDB, see https://kubernetes.io/docs/concepts/workloads/pods/disruptions/

Cause: There are PDB objects for Istio (or any other application) resources with minimum replica as 1. This prevents the "kubectl drain" from succeeding during the Kubernetes upgrade.

Workaround: Execute the following commands on the Kubernetes Master before initiating the Kubernetes Upgrade from the **Kubernetes Cluster** screen. The following example is for Istio:

```
kubectl -n istio-system delete
poddisruptionbudget/istiod
kubectl -n istio-system
delete poddisruptionbudget/
istio-ingressgateway
kubectl -n istio-system
delete poddisruptionbudget/
istio-egressgateway
```

NOTE: This workaround can also be applied if the Kubernetes upgrade fails with `Failed to drain node` error on the Kubernetes hosts/workers. In such case, execute the preceeding `kubectl` commands on the Kubernetes Master, and continue with the Kubernetes upgrade on the remaining workers using the **Retry Kubernetes Upgrade on Failed Workers** action on the cluster from the **Kubernetes Cluster** screen.

Before doing Kubernetes Upgrade, make sure you have drained all the pods on the node. If an application has Pod disruption budget (PDB) violation, that pod will not get drained and Kubernetes upgrade will fail. This typically happens when you have smaller cluster with limited resources.

PDB voilation will show a similar message like:

```
kubectl drain
mip-bd-vm694.mip.storage.hpecorp.net --d
elete-local-data --ignore-daemonsets --t
imeout=5m
evicting pod airflow-base/af-base-nfs-0
evicting pod airflow-base/
af-base-postgres-0
error when evicting pod "af-base-nfs-0"
(will retry after 5s): Cannot evict pod
as it would violate the pod's
disruption budget.
error when evicting pod
"af-base-postgres-0" (will retry after
5s): Cannot evict pod as it would
violate the pod's disruption budget.
```

EZCP-561: When Istio mTLS is enabled in `STRICT` mode, the Kiali Dashboard and KubeDirector service endpoints are not accessible through NodePort

Symptom: When Istio is configured to use Mutual Transport Layer Security (mTLS) in `STRICT` mode, the following issues occur:

- None of the KubeDirector service endpoints are accessible through the NodePort service.
- If mTLS in `STRICT` mode is enabled in a tenant, the Kiali Dashboard is not accessible through NodePort. Clicking on the endpoint results in an error.

Workaround: If possible, configure Istio to use `PERMISSIVE` mode (the default mode).

EZESC-232: "Failed to pull image" ImagePullBackoff Errors received on Kubernetes clusters

When working with Kubernetes clusters in HPE Ezmeral Runtime Enterprise, you receive errors similar to the following:

```
Failed to pull image "bluedata/
hpe-agent:1.1.5": rpc error: code =
Unknown desc = Error response from
daemon: toomanyrequests: You have
reached your pull rate limit. You may
increase the limit by authenticating
and upgrading: https://www.docker.com/
increase-rate-limit
```

Cause: Kubernetes clusters running on any version of HPE Ezmeral Runtime Enterprise can occasionally encounter problems caused by the pull rate limit that Docker Hub applies to all free and anonymous accounts. These limits can cause cluster creation and application deployment to fail. If Kubernetes pods in a non-Air-gap environment are failing to come into Ready state and are showing ImagePullBackoff or related errors, this is the most likely cause.

Workaround: Do one of the following:

- Wait until the current rate limiting timeout has expired, then re-try.
- Create a local image registry, then configure the air-gap settings to use that registry. For more information about air gap, see Kubernetes Air-Gap Requirements on page 834.

  NOTE:

  Hewlett Packard Enterprise strongly recommends performing air-gap configuration steps before adding Kubernetes hosts to the HPE Ezmeral Runtime Enterprise environment. Kubernetes hosts do not implement air-gap changes until the hosts are rebooted or the Kubernetes version is upgraded.

- Upgrade your Docker Hub account as described in https://www.docker.com/increase-rate-limits (link opens an external website in a new browser tab/window), then on all hosts, do the following:

1. Execute a `docker login` operation with the credentials of the ugpgraded account.

   Docker will create or update its `config.json` file after a successful login (or you might want to use an existing configuration file).

2. Ensure that kubelet uses the new `config.json` file by placing it in one of the known search locations kubelet uses for credential files:

   a. Create a `.docker` directory directly under the root of the filesystem and place the config.json file in that directory. For example: `/.docker/config.json`

   b. Restart kubelet:

   ```
   systemctl restart kubelet
   ```

   c. Verify that kublet has restarted:

   ```
   systemctl status kubelet
   ```

   Kubelet will then choose that `config.json` file and use the paid account that generated that config, ensuring that no image pull rate limit will be exceeded.

The following article (link opens an external website in a new browser tab/window) shows all the locations that kubelet searches for Docker credentials files:

https://kubernetes.io/docs/concepts/containers/images/#configuring-nodes-to-authenticate-to-a-private-registry

- Create a Docker proxy cache as described in the following article (link opens an external website in a new browser tab/window):

  https://docs.docker.com/registry/recipes/mirror/

EZCP-811: Webterms do not work for imported clusters. You will encounter an error if you try to start a webterm on an imported cluster.

Workaround: Execute the following command using either the Kubeconfig file used to import the cluster or a Kubeconfig file for the imported cluster downloaded from the HPE Ezmeral Runtime Enterprise as described in Downloading Admin Kubeconfig:

```
kubectl patch hpecpconfigs
hpecp-global-config -n hpecp --type
merge --patch '{"spec":{"fsMount":
{"enabled":false} } }'
```

After the command is issued, starting a webterm should not generate an error.

EZCP-823: Kubernetes Upgrade dialog empty or not showing latest Kubernetes version after upgrade to HPE Ezmeral Runtime Enterprise 5.3.x.

Workaround: Refresh the browser screen.

BDP-574: Unable to add a Kubernetes host when Platform HA (High Availability) is being enabled.

Workaround: Wait until Platform HA finishes before adding the Kubernetes host.

HAATHI-15093 : A GPU is visible in a non-GPU-requesting pod.

Symptom:When an app spawns on a device having a GPU, it is able to access the GPU even when there are no requests for one. This is a known issue with the NVIDIA k8s-device-plugin.

Workaround: You must manually create an environment variable in the `Kubedirectorcluster` YAML that 'hides' the GPU from the App. The variable is named `NVIDIA_VISIBLE_DEVICES` with value `VOID`.

For example:

```
apiVersion: "kubedirector.bluedata.io/
apiVersion" kind:
"KubeDirectorCluster" metadata: name:
"sample-name" spec: app: sample-app
roles: - id: samplerole resources:
requests: memory: "4Gi" cpu: "2"
limits: memory: "4Gi" cpu: "2" env: -
name : "NVIDIA_VISIBLE_DEVICES"
value: "VOID"
```

Spark on Kubernetes Issues (5.6.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.6.0. Unless otherwise noted, these issues also apply to later releases.

Livy Session: PySpark code in Livy session results in an error

Symptom: Running PySpark code in Livy session returns the following error:

```
'JavaPackage' object is not callable
```

Cause: PythonSQLUtils is not imported in java_gateway.jvm

Workaround: Perform explicit imports by running the following commands in Livy session:

```
from py4j.java_gateway import
java_import
jvm = SparkContext._jvm
java_import(jvm,
"org.apache.spark.sql.api.python.*")
```

EZSPA-1037: Data Fabric DB OJAI jobs fails with ANTLR incompatibility exception

Symptom: Data Fabric DB OJAI jobs will fail with ANTLR incompatibility exception.

Workaround: Contact Hewlett Packard Enterprise Technical Support.

EZSPA-1010: Some pyspark APIs do not work, due to python version compatibility

Symptom: Some pyspark APIs do not work as expected.

Cause: Some pyspark APIs do not work, due to python version compatibility issues.

Workaround: Contact Hewlett Packard Enterprise Technical Support.

EZSPA-1008: Livy session fails when group names for users in Active Directory are not POSIX compliant.

Symptom: When you start a Livy session on HPE Ezmeral Runtime Enterprise as `user1` and group names for users in Active Directory are not POSIX compliant, the following error occurs:

```
groupadd: 'Domain Users' is not a
valid group name
```

Cause: The main group name of the `user1` user in Active Directory database is `Domain Users`. `Domain Users` group name contains a space symbol which makes it an invalid group name in Linux.

Workaround: The group names for users in Active Directory need to be POSIX compliant. The set of valid user names in POSIX is defined as lower and upper case ASCII letters, digits, period, underscore, and hyphen. Note that hyphen is not permitted as first character of the user name.

Spark on Kubernetes Issues (5.5.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.5.0. Unless otherwise noted, these issues also apply to later releases.

EZCP-3572: Add-ons upgrade for Spark Operator add-on fails after upgrading from 5.4.x to 5.5.x or later version of HPE Ezmeral Runtime Enterprise.

Symptom: When you perform the following steps:

1. Create a Kubernetes cluster in 5.4.x and 5.5.x.
2. Enable the Spark Operator add-on.
3. Upgrade the platform to 5.5.x from 5.4.x.
4. Run Kubernetes add-ons upgrade script.

The Spark Operator add-on upgrade fails and you'll see the following warning message:

```
2022-10-25 04:40:37,032 INFO add-ons
upgrade failed: cluster
        state: warning
```

Cause: Spark Operator is running with an old Spark Operator image in a cluster.

Workaround: Contact Hewlett Packard Enterprise support team.

Spark on Kubernetes Issues (5.4.1)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.1.

EZCP-2624: Launching KubeDirector application tiles for Spark results in Config Error.

Symptom: After upgrading to HPE Ezmeral Runtime Enterprise 5.4.1 from HPE Ezmeral Runtime Enterprise 5.3.x; when you launch the Livy, Spark History Server, Spark Thrift Server, and Hive Metastore in standard tenants after running the `sparkapps.sh` command, you will get the following error:

```
Config Error Detail: execution of app
config failed: configure failed with
```

```
exit status {120}
Last Config Data Generation: 1
Last Configured Container: docker://
d7f5c968a029f494889da2d06d26ff066b52f3
42538e4ad822e5d88638e57181
Last Connection Version: 0
Last Known Container State:
unresponsive
Last Setup Generation: 1
Start Script Stderr Message: Error
from server (Forbidden): configmaps
"cluster-cm" is forbidden: User
"system:serviceaccount:nonml:ecp-tenan
t-member-sa" cannot get resource
"configmaps" in API group "" in the
namespace "<namespace>"
Start Script Stdout Message: Error:
expected at most two arguments,
unexpected arguments:
image.tag=<spark-tenant-services-imag
e-tag>
Failed to exec: helm install
<spark-tenant-services-name> /
<path-to-spark-tenant-services-chart>
 --namespace --set image.tag=
<spark-tenant-services-image-tag>
```

Cause: The `ecp-tenant-member-sa` service account was added in HPE Ezmeral Runtime Enterprise 5.4.0. The `member` rolebinding do not have the `ecp-tenant-member-sa` service account binding on the tenants that were created prior to 5.4.0 releases.

Workaround: Delete the existing `member` rolebinding. To delete the rolebinding, run:

```
kubectl delete
rolebinding <name_of_rolebinding> -n
<tenant-namespace>
```

Deleting an existing `member` rolebinding will automatically create a new `member` rolebinding with `ecp-tenant-member-sa` service account binding providing an access to the current tenant services.

Spark on Kubernetes Issues (5.4.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.0. Unless otherwise noted, these issues also apply to later releases.

EZESC-1211: Unable to update Helm charts for Hive Metastore, Livy, Spark History Server, and Spark Thrift Server using `kubectl apply` command.

Symptom: When you run the `kubectl apply -f <spark-services-yaml-file>` command to update the Helm charts for Hive Metastore, Livy, Spark History Server, and Spark Thrift Server in the same cluster, update fails with the following error message:

```
Error: release: already exists
Failed to exec: helm install
<spark-services-release-name> /
<path-to-helm-chart> --namespace
```

```
<tenant-namepace> --set
image.tag=202202161825P150 --set
eventlogstorage.kind=pvc --set
eventlogstorage.storageSize=10Gi --set
 eventlogstorage.pvcStoragePath=/
<path-to-storage>
```

Cause: Only one instance of Hive Metastore, Livy, Spark History Server, and Spark Thrift Server can be installed in the single tenant within a cluster. When you try to install the multiple instances of the Spark services, Helm with throw an error since the same cluster name is used as the release name for all the Helm installation of the Spark services.

Workaround: To update the Helm charts for Hive Metastore, Livy, Spark History Server, and Spark Thrift Server, see Updating Helm Charts for Spark Services on page 253.

EZSPA-576: Authentication fails on Spark tenant services when the permissions for External User Groups on tenants are set at a higher level than the External Groups on Data Fabric clusters.

Symptom: The authentication on tenant services, for example, Livy, Spark History Server, Spark Thrift Server, Hive Metastore fails with the following error:

```
INFO login.PasswordAuthentication:
Failed authentication for user qa1:
javax.security.auth.login.FailedLoginE
xception: Permission denied.
ERROR server.BasicAuthHandler: User
Principal is null while trying to
authenticate with Basic Auth
```

Cause: You have set the permissions for **External User Groups** on tenants at a higher level than the **External Groups** on Data Fabric clusters.

Workaround: Ensure the permissions for **External Groups** on **Creating Kubernetes Cluster** step is set at a broader level than the permissions for **External User Groups** on **Creating New K8s Tenant** step. See Kubernetes Tenant/Project External Authentication on page 456 and Creating a New Kubernetes Cluster on page 463.

EZSPA-566: Spark Thrift Server restarts continuously when Hive Metastore ConfigMap is not set.

Symptom: When you do not enter the ConfigMap with `hive-site.xml` configuration of the Hive Metastore during the Spark Thrift Server installation, Spark Thrift Server restarts continuously and gives the following error:

```
Error: Unable to instantiate
org.apache.hadoop.hive.ql.metadata.Ses
sionHiveMetaStoreClient
```

Cause: The ConfigMap with `hive-site.xml` configuration of the Hive Metastore was not identified and is therefore missing during the Spark Thrift Server installation.

Workaround: You can enter ConfigMap values using YAML or HPE Ezmeral Runtime Enterprise GUI and there are three separate workarounds for three situations. See Integrating Spark Thrift Server with Hive Metastore on page 307.

EZSPA-508: Spark submit fails when using the third-party dependency jars on MinIO.

Symptom: When you submit the Spark applications configured using the third-party dependency jars on MinIO, the `spark-submit` fails with the following exception:

```
Exception in thread "main"
com.amazonaws.SdkClientException:
Unable to execute HTTP request
```

Cause: Unable to add CLI options to the spark-submit command.

Workaround: None at this time.

EZSPA-504: Livy and Hive Metastore integration fails in the non Data Fabric type tenants.

Symptom: Livy and Hive Metastore integration fails in non Data Fabric (none) type tenants with the following message:

```
java.lang.RuntimeException:
java.io.IOException: Could not create
FileClient err: 104
```

Workaround: None at this time.

EZSPA-446: Spark application fails when `jars` option is set with non-file URI scheme for SparkR.

Symptom:

When you set the jars option for DataTap with non-file schema, for example, `- local:///opt/bdfs/bluedata-dtap.jar`, Spark applications fail with the following exception:

```
Exception in thread "main"
java.lang.IllegalArgumentException:
URI scheme is not "file"
```

Cause: The `jars` option is set with non-file URI scheme for SparkR.

```
deps:
    jars:
      - local:///opt/bdfs/
bluedata-dtap.jar
```

Workaround: To integrate SparkR with DataTap, configure SparkR with the file URI scheme.

For example: Set the `files` option to add DataTap jar to classpath for SparkR.

```
deps:
   files:
     - local:///opt/bdfs/
bluedata-dtap.jar
```

EZSPA-442: Authentication fails on SAML environment.

Symptom: When you authenticate Livy on SAML, authentication fails with the following message:

```
INFO login.PasswordAuthentication:
Failed authentication for user
<user1>:
javax.security.auth.login.FailedLoginE
xception: Permission denied.
```

Workaround: None at this time.

EZSPA-232: Livy and Hive Metastore integration fails when using DataTap to access the data from same Hive Metastore.

Symptom: When you create Livy sessions in the DataTap integration enabled environment, you are unable to use Hive Metastore. For example: You are unable to view the tables created in one Livy session from the another Livy session.

Workaround: To use the Hive Metastore in Livy, remove `"spark.driver.extraClassPath"` option from Livy session configurations. However, in this case, you are unable to pass the application dependencies using `dtap` in Livy.

EZCP-1808: After upgrading to HPE Ezmeral Runtime Enterprise 5.4.0, launching KubeDirector Spark applications as Kubernetes Tenant Admin or Kubernetes Tenant member, fails with an error.

Symptom: After you upgrade to HPE Ezmeral Runtime Enterprise 5.4.0, if you launch KubeDirector Spark applications as Kubernetes Tenant Admin or Kubernetes Tenant member, applications fail with an error.

Workaround:

1. Access Kubernetes cluster as Cluster Administrator and download the `kubeconfig` file. To download the `kubeconfig` file, you can either follow the steps in Downloading Admin Kubeconfig on page 486 or SSH to Kubernetes Master using following command:

   ```
   kubectl get hpecptenant -n hpecp
   ```

   Example of output:

   ```
   NAME       AGE
   hpecp-tenant-4   9h
   hpecp-tenant-5   8h
   hpecp-tenant-6   8h
   ```

2. Replace the **<tenant-name>** with the desired value using following command:

   ```
   kubectl edit hpecptenant
   <tenant-name> -n hpecp
   ```

3. Add **Patch** verb at `kubedirectorclusters` resources for following Role Ids:

   Patch verb is added in the following examples:

   Add in **Default Admin RBACS**:

```
- apiGroups:
      - kubedirector.hpe.com
      resources:
      - kubedirectorclusters
      - kubedirectorapps
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
      - patch
```

   In **Default Member RBACS**:

```
- apiGroups:
      - kubedirector.hpe.com
      resources:
      - kubedirectorclusters
      verbs:
      - create
      - update
      - delete
      - get
      - list
      - watch
      - patch
```

   Also, Add at **secrets** resources of **Default Member RBACS**

```
- apiGroups:
      - ""
      resources:
      - secrets
      verbs:
      - get
      - create
      - update
      - patch
```

4. Save and exit the file.

Spark on Kubernetes Issues (Prior Releases)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.3.x. Unless otherwise noted, these issues also apply to later releases.

- You may encounter a `certificate generation failed` error when executing `spark-submit` or `spark- shell` commands in the `tenantcli` or `spark-client` pods. You can avoid this issue by executing the command using the `--conf spark.ssl.enabled=false` option. Doing so disables encryption for the Spark driver UI. The UI is not exposed outside of the Kubernetes cluster, so it is safe to use this option.

- The pod restarts continuously instead of transitioning to an **Error** state if `hivesitesource` points to an existing ConfigMap that does not have a `hive-site.xml` key.
- The Autoticket-generation feature does not work for scheduled Spark applications. Manually create your user secrets using the `ticketcreator.sh` script in the `tenantcli` pod for this purpose.

KubeDirector Issues (5.4.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.0. Unless otherwise noted, these issues also apply to later releases.

EZML-810: The Jupyter notebook does not appear in the UI.

Symptom: When creating or launching a Jupyter notebook with `kubectl apply`, the Jupyter notebook does not appear in the UI. However, the Jupyter endpoint is visible.

Workaround: Add the user ID in the label in `nb.yaml` as:

```
---
metadata:
    labels:
        kubedirector.hpe.com/
createdBy:
```

This will prevent user ID mismatch, allowing the logged-in user to view the Jupyter notebook.

EZML-994: When opening an R-kernel in Jupyter notebook, a `TypeError` occurs.

Symptom: When opening an R-kernel in Jupyter notebook, a pop-up appears with the message `TypeError`.

Cause: This is a known issue with JupyterLab 2.3.

Workaround: Click **Dismiss** and proceed with your R session.

EZML-1037: When submitting a KFP job in a KD notebook using Kale, an `RPC Error` occurs.

Symptom: When uploading a pipeline in KD notebook, the message `An RPC Error has occurred` is displayed.

Workaround: Before creating the KFP client, execute:

```
%kubeRefresh
```

After successful execution, recreate the KFP client. For detailed instructions about the prerequisites of Kale, see: `examples/kubeflow/kale/README.ipynb`.

If the error persists after you have executed `%kubeRefresh` and the kubeconfig file is still fresh, then dismiss the `RPC Error` message and restart the Kale extension.

KubeDirector Issues (Prior Releases)

The following issues were identified in a version of HPE Ezmeral Runtime Enterprise prior to version 5.4.0. Unless otherwise noted, these issues also apply to later releases.

EZESC-1066: "503 Service Unavailable" error for MinIO or MySQL after MLflow cluster pod automatically restarts.

Symptom: When a pod managed by an MLflow cluster is deleted and then automatically recreated,

attempts to access the MinIO service endpoint or MySQL result in the error:

```
503 Service Unavailable
```

This issue occurs on pods that are managed by an MLflow cluster that is configured with persistent storage (PVC) only.

Cause: When an MLflow cluster is configured with persistent volumes, KubeDirector does not automatically execute startup scripts when the controller restarts. However, state information for MySQL is not retained by the persistent volume, and MinIO is not restarted because it is not a `systemd` process. The startup script (`startscript`) is responsible for configuring and starting services such as MySQL and MinIO.

Workaround: From the Kubernetes master node, manually execute the startup script for the pod by executing the following commands, where `<kdcluster_pod>` is the name of the pod and `<tenant_ns>` is the tenant namespace:

```
kubectl exec -it <kdcluster_pod>  -n
<tenant_ns> bash
opt/guestconfig/appconfig/
startscript --configure
exit
```

EZESC-217: 503 Service Unavailable error when connecting to training engine instance

Attempts to connect to a training engine instance from a JupyterLab Notebook fail. When you attempt to connect to the service endpoint of the training engine instance in a browser, the error "503 Service Unavailable" is returned.

Cause: One of the possible cause is when the High Availability Proxy (HAProxy) service is not running on the gateway host. If you are not sure whether HAProxty is running or not, contact HPE support for assistance.

Workaround: If this is HAProxy issue, then start (or restart) the HAproxy service. From the master node, enter the following command::

```
kubectl exec -c app -n
<tenant-namespace>
<trainingengineinstance-loadbalancer-p
od> - systemctl restart haproxy
```

Airflow Issues (5.5.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.5.0. Unless otherwise noted, these issues also apply to later releases:

EZML-2026: Airflow does not work on Kubernetes clusters set with a custom pod domain.

Symptom: Airflow does not work on Kubernetes clusters set with a pod domain other than `cluster.local`.

Workaround: For each of the four listed resources, perform the steps described below:

- `sts af-cluster-airflowui`
- `sts af-cluster-scheduler`
- `cm af-cluster-airflowui`
- `cm af-cluster-scheduler`

1. Edit the resource:

   ```
   kubectl edit <resource-name> -n
   <airflow-tenant-ns>
   ```

   For example:

   ```
   kubectl edit
   sts af-cluster-airflowui -n
   <airflow-tenant-ns>
   ```

2. Delete all labels in the `metadata` section for the resource.

   For example:

   ```
   <...>
   metadata:
     <...>
     labels:
       custom-resource:
   v1alpha1.AirflowCluster
       custom-resource-name:
   af-cluster
       custom-resource-namespace:
   <...>
       using: <...>
   <...>
   ```

3. Replace all occurrences of `af-base-sql.airflow-base.svc.cluster.local` with `af-base-sql.airflow-base`.

Kubeflow Issues (5.5.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.5.0. Unless otherwise noted, these issues also apply to later releases:

- Error "Could not find CSRF cookie XSRF-TOKEN in the request" is returned when creating a Kserve model in Kubeflow UI exposed via HTTP. For more information about this issue, see: https://github.com/kubeflow/manifests/pull/2262

Kubeflow Issues (5.4.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.0. Unless otherwise noted, these issues also apply to later releases.

- Kubeflow does not support groups. See:

  https://github.com/kubeflow/kubeflow/issues/4188

- The Dex authentication component does not support the use of LADP/AD external groups. See:

  https://github.com/dexidp/dex/issues/1562

EZML-616: Only a single AD server configuration is supported, even when multiple LDAP server addresses are provided.

Symptom: Only a single AD server configuration is supported, even when multiple LDAP server addresses are provided on the **Cluster Configuration** tile. A single LDAP server adderss is set when DEX is created, and does not change when the chosen LDAP server address becomes unavailable or inaccessable.

Cause: Kubeflow DEX does not support the use of multiple AD/LDAP servers for authentication.

Workaround: If the selected LDAP server that was set during installation becomes inaccessable, specify a different server as follows:

1. Get the current configuration for DEX from the secret:

   ```
   kubectl get secrets -n
   auth dex-config-secret -o
   "jsonpath={.data['config\.yaml']}"
   | base64 -d
   ```

   Copy the returned value and save it.

2. In the copied value, locate the string which starts with the substring `host:` . After this substring, replace the existing domain string with the domain for the correct server. There should be only one domain.

   For example:

   ```
   host: example.com:636
   ```

3. Open any base64 encoder and encode the whole modified configuration.

   The following are links to base64 encoders:

   - https://www.base64encode.org/
   - base64

4. Update the secret:

   ```
   kubectl edit secrets -n auth
   dex-config-secret
   ```

   Replace the value after `config.yaml` with your modified and encoded value.

5. Save the changes.

6. Restart the DEX deployment:

   ```
   kubectl rollout restart deploy -n
   auth dex
   ```

EZML-1475: When you deploy a model using InferenceService, the KNative Serving controller fails to fetch the image used by the model from the airgap docker image registry.

Symptom: When you deploy a model using InferenceService, the InferenceService fails to become `READY`. The KNative Serving controller fails to fetch the image used by the model from the airgap docker image registry and gives the message: `x509: certificate signed by unknown authority.`

Workaround:

1. After deploying Kubeflow, run the following as Cluster Administrator:

```
kubectl edit cm -n knative-serving
config-deployment
```

2. Add the following under the `data` field:

```
registriesSkippingTagResolving:
"<host-name-of-your-airgap-image-re
gistry>"
```

Kubeflow Issues (Prior Releases)

The following issues were identified in a version of HPE Ezmeral Runtime Enterprise prior to version 5.4.0. Unless otherwise noted, these issues also apply to later releases.

- If you specify an external user group, the group is not taken into account when a user logs in to Kubeflow. The user will be allowed to log in to Kubeflow regardless of to which groups that the user belongs. See the following for more information:

  https://stackoverflow.com/questions/58276195/mandate-group-search-condition-in-dex-ldap-coonector

- Occasionally, the `v1beta1.webhook.cert-manager.ioapiservice` is unavailable for a period of time after deploying Kubeflow services (applying a manifest). To make the service available, restart the service as follows:

```
kubectl delete apiservices v1beta1.webhook.cert-manager.io
```

- There is an issue with Istio authorization for HTTP traffic in which the KFServing predict request returns `503 Service Unavailable`. See the following for more information:

  https://github.com/kserve/kserve/issues/820

Katib Issues (Prior Releases)

The following issues were identified in a version of HPE Ezmeral Runtime Enterprise prior to version 5.4.0. Unless otherwise noted, these issues also apply to later 5.4.x releases.

The following issues occur in Katib, which is a Kubernetes-native project for automated machine learning.

- Suggestion pods running after experiment completes:

  https://github.com/kubeflow/katib/issues/1043

- Katib with Kubernetes 1.19 and higher:

  https://github.com/kserve/kserve/issues/1197

  https://github.com/kubeflow/katib/issues/1395

General Platform Issues (5.6.1)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.6.1. Unless otherwise noted, these issues also apply to later 5.6.x releases.

EZCP-3949: On Python 3 hosts (RHEL 8 and SLES 15.4), clicking Support/Troubleshooting in the HPE Ezmeral Runtime Enterprise UI might result in an error. Attempting to generate SOS logs on the Support/Troubleshooting page results in an error.

Symptom: On RHEL 8 OS, clicking **Support/Troubleshooting** in the HPE Ezmeral Runtime Enterprise UI might return a `404 Page Not Found` error. Attempting to generate SOS logs on the **Support/Troubleshooting** page results in the following error message:

```
Got an error while performing the
operation
Additional Details: Error
```

Workaround: None at this time.

General Platform Issues (5.6.0)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.6.0. Unless otherwise noted, these issues also apply to later 5.6.x releases.

EZCP-3844: When you upgrade to HPE Ezmeral Runtime Enterprise 5.6.0, and perform the Cluster Upgrade from 1.21.x to 1.22.x, some pods are in CrashLoopBackOff state.

Symptom:When you upgrade to HPE Ezmeral Runtime Enterprise 5.6.0, and perform the Cluster Upgrade from 1.21.x to 1.22.x, some pods are in `CrashLoopBackOff` state.

Cause: Pods that are running on the worker are unable to access the pods that are running on the master node, due to missing routes on the master node.

Workaround: On the master node that is missing the routes to some or all the workers, do the following:

1. Find the canal pod running in that master node, using the command:

```
CANAL_POD_NAME=$(kubectl get
pods -n kube-system -o wide | grep
<hostipaddr> | grep canal | awk
'{print $1}')
```

2. Delete the canal pod using the following command:

```
kubectl delete -n kube-system pod
${CANAL_POD_NAME}
```

3. Restart the pod. The missing routes will be restored.

General Platform Issues (5.4.1)

The following issues were identified in HPE Ezmeral Runtime Enterprise 5.4.1. Unless otherwise noted, these issues also apply to later 5.4.x releases.

EZCP-2669: When an attempt to enable High Availability (HA) on HPE Ezmeral Runtime Enterprise fails, the log repeats error messages multiple times.

Symptom: When an attempt to enable HA on HPE Ezmeral Runtime Enterprise fails, the `bds_mgmt.log`

repeats error messages multiple times, making it difficult to read and debug the issue.

Workaround: To view log files that capture the whole configuration, look at `/var/log/bluedata/install/enableha*`.

General Platform Issues (Prior Releases)

The following issues were identified in a version of HPE Ezmeral Runtime Enterprise prior to version 5.4.0. Unless otherwise noted, these issues also apply to later 5.4.x releases.

EZESC-253: After upgrade, UI becomes inaccessible and browser displays internal error 500.

Symptom: Following an upgrade to HPE Ezmeral Runtime Enterprise 5.3, the UI for the controller is inaccessible, failing with internal error 500. The system fails with the error:

```
No space left on device
```

The `/var/lib/monitoring/logs` director contains large `hpecp-monitoring_access` and `hcep-monitoring_audit` logs.

Cause:

Dangling Search Guard indexes exist after the upgrade. You might see log entries similar to the following:

```
[WARN ][o.e.g.DanglingIndicesState]
[xxxxx] [[searchguard/
xxx-xxxxxxxxx-xxxxxxx]] can not be
imported as a dangling index, as index
with same name already exists in cluster
metadata
```

Workaround: Search Guard indexes are not used by HPE Ezmeral Runtime Enterprise 5.3. You can remove the Search Guard indexes, delete the large log files, and resume monitoring on the HA nodes.

1. Remove the Search Guard indexes using one of the following methods:
   - If Elasticsearch is running, you can delete the Search Guard index through the Elasticsearch REST API.

     For example:

     ```
     curl --insecure  -u $
     (bdconfig --getvalue
     bdshared_elasticsearch_admin):$
     (bdconfig --getvalue
     bdshared_elasticsearch_adminpass
     ) --silent -X DELETE https://
     localhost:9210/searchguard
     ```

   - If Elasticsearch is not able to run, you must identify and delete SearchGuard indexes manually:

a. Identify the indexes.

   Change the directory to `/var/lib/monitoring/elasticsearch/nodes/0`, then enter the following command:

   ```
   find . -name
   "state-*.st" -print | xargs
   grep searchguard
   ```

   All the indices that are from Search Guard are displayed. You can use matching entries to determine which indexes to remove.

   For example, this line identifies a state file related to that contains the word Search Guard. The index name is part of the full file path of that file. In this example, the index name: `xtSTTUb7RgOeUlCXWH8dAg`

   ```
   ./indices/
   xtSTTUb7RgOeUlCXWH8dAg/
   _state/state-45.st matches
   ```

b. Use the `rm` command to remove the index.

   For example:

   ```
   rm -rf ./indices/
   xtSTTUb7RgOeUlCXWH8dAg
   ```

2. Delete the large log files.

3. On the HA cluster nodes only, restart monitoring. For example, from the controller, enter the following command:

   ```
   HPECP_ONLY_RESTART_ES=1 /opt/
   bluedata/bundles/hpe-cp-*/
   startscript.sh --action
   enable_monitoring
   ```

BDP-2879: The Python ML and DL Toolkit lists a deleted Training cluster in the %attachments list.

Workaround: Ignore the deleted cluster. No jobs will be submitted to deleted clusters.

BDP-841: When enabling multi-domain authentication, the password field must be filled out for all domains before submitting changes to any domain, otherwise the web interface will fail to react.

Workaround: None at this time.

HAATHI-15068: Unable to create a tenant or FS mount if any host is down.

Workaround: Consider removing the Kubernetes host from the Kubernetes cluster or wait until the host is back up and running.

HAATHI-12781: When HPE Ezmeral Runtime Enterprise is installed on RedHat 7.x systems, system reboots are observed under heavy load.

Workaround: Update the RedHat kernel to the newest kernel version.

HAATHI-14220: Adding a license when one or more Worker hosts is in an error state may cause an error.

Workaround: Remove the affected hosts before uploading the license.

HAATHI-12810: After restarting the container that handles monitoring, the service may fail to restart and will show red in the Services tab of the Platform Administrator Dashboard screen.

Workaround: Restart the service manually from the Controller host by executing the command `systemctl restart bds-monitoring`.

HAATHI-12829: For RHEL/CentOS 7.x OS installs, if a server is physically rebooted, some services that depend on network services may be down as shown in the Services tab of the Platform Administrator Dashboard screen.

Workaround: Execute the following commands on the Controller host:

```
$ systemctl stop NetworkManager
$ systemctl disable NetworkManager
$ systemctl restart network
$ systemctl restart bds-controller
$ systemctl restart bds-worker
```

HAATHI 13253: HPE Ezmeral Runtime Enterprise does not compress or archive Nagios log files.

Workaround: Manually archive files as needed in the `/srv/bluedata/nagios` directory on the Controller.

EZCP-463: Platform HA must be enabled before creating Kubernetes clusters.

Workaround: If you enable Platform HA after Kubernetes cluster creation, then reconfigure host monitoring as follows:

1. On a Kubernetes master node bring up the monitoring bootstrap deployment:

   ```
   kubectl -n hpecp-bootstrap scale
   deployment/
   hpecp-bootstrap-hpecp-monitoring
    --replicas=1
   ```

2. Exec into the bootstrap pod

   ```
   kubectl -n hpecp-bootstrap
   exec -it $(kubectl -n
   hpecp-bootstrap get -o
   jsonpath='{.items[0].metadata.name}
   ' pods -l
   name=hpecp-bootstrap-hpecp-monitori
   ng) -c hpecp-monitoring - bash
   ```

3. Delete running deployment (if exist):

   ```
   kubectl -n kube-system -delete -f /
   workspace/monitoring.yaml
   ```

4. Export / change any needed bds_xxx env variables (e.g. redeploy after HA enable)

   ```
   export bds_ha_enabled='Yes'
           export
   bds_ha_nodes='<controller IP list>'
   ```

   (e.g. `export bds_ha_nodes='16.143.21.35,16.143.21.237,16.143.21.38'`)

5. Run startscript install:

```
/usr/local/bin/
startscript --install
```

This places `metricbeat.yaml` in the workspace folder.

6. Deploy metricbeat deployment:

```
kubectl -n kube-system create -f /
workspace/monitoring.yaml
```

7. Exit the bootstrap pod and scale down bootstrap deployment:

```
kubectl -n hpecp-bootstrap scale
deployment/
hpecp-bootstrap-hpecp-monitoring
 --replicas=0
```

BDP-685: Kubernetes cluster creation fails with an "internal error."	***Workaround:*** Remove the Kubernetes hosts, verify that all system clocks are synchronized, and then re-add the hosts and recreate the Kubernetes cluster.
BDP-852: All uploaded files and new folders created by AD/LDAP users via the HPE Ezmeral Runtime Enterprise FS mounts interface will have root ownership and full permission for all tenant members.	***Workaround:*** None at this time.
BDP-1868: An admin kubeconfig file downloaded from an imported external Kubernetes cluster will not contain expected edits from the HPE Ezmeral Runtime Enterprise web interface.	***Workaround:*** Manually edit the kubeconfig file after download.

Application Issues (Prior Releases)

The following issues were identified in a version of HPE Ezmeral Runtime Enterprise prior to version 5.4.0. Unless otherwise noted, these issues also apply to later 5.4.x releases.

HAATHI-14109: When using CEPH for persistent storage, a discrepancy between the client and server versions will cause HPE Ezmeral Runtime Enterprise to fail to load App Store images with the error "Failed to map the volume."	***Workaround:*** Remove the persistent storage until the client and server versions are the same.
HAATHI-14192: Running the Impala shell on a container where the Impala daemon is not running.	***Workaround:*** Use the `-i` option to refer to the worker node. For example, `impala-shell -i <worker hostname>`.
HAATHI-14461: Notebooks with a name that includes one or more spaces cannot be committed to GitHub.	***Symptom:*** When working in an AI/ML project that includes a GitHub repository, creating a Jupyterhub notebook with a name that includes one or more spaces will cause an error when trying to commit that notebook to GitHub. ***Workaround:*** Do not include any spaces when naming a Jupyterhub notebook.
HAATHI-10733: Hive jobs that use DataTap paths may fail with a SemanticException error.	***Cause:*** When Hive creates a table, the location where the table metadata is stored comes from the Hive

configuration parameter `fs.defaultFS` by default (which will point to the cluster file system). If a Hive job references DataTap paths outside of the file system where the table metadata is stored, then the job will fail with a `SemanticException` error because Hive enforces that all data sources must come from the same file system.

Workaround: Explicitly set the table metadata location to a path on the same DataTap that you will use for the job inputs and/or outputs, using the `LOCATION` clause when creating the table. For example, if you intend to use the **TenantStorage** DataTap, you would set the table metadata location to some path on that DataTap such as:

```
CREATE TABLE docs (c1 INT, c2 STRING)
LOCATION
'dtap://TenantStorage/hive-table-docs'
```

HAATHI-12546: Some http links in applications running on HPE Ezmeral Runtime Enterprise show the hostname of the instance. These links will not work when HPE Ezmeral Runtime Enterprise is installed with the non-routable network option.

Workaround: See "Configure Client to use Hostname instead of IP Address, below."

HAATHI-13254: If a user updates an app inside a container instead of via the App Store screen, then cluster expansion will fail.

Workaround: Expand the cluster before performing the upgrade. Once the update is complete, edit `classpath` to point to the correct .jar files, such as `hadoop-common-*.jar`.

DOC-9: Cloudera Manager reports incorrect values for a node's resources.

Cause: Cloudera Manager accesses the Linux `/proc` file system to determine the characteristics of the nodes it is managing. Because container technology is used to implement virtual nodes, this file system reports information about the host rather than about the individual node, causing Cloudera Manager to report inflated values for a node's CPU count, memory, and disk.

Workaround: Use the web interface to see a node's virtual hardware configuration (flavor).

DOC-19: Spark applications may wait indefinitely if no free vCPUs are available.

Cause: This is a general Spark behavior, but it is worth some emphasis in an environment where various virtual hardware resources (possibly in small amounts) can be quickly provisioned for use with Spark.

Workaround:

A Spark application will be stuck in the **Waiting** state if all vCPUs in the cluster are already considered to be in-use (by the Spark framework and other running Spark applications). In Spark version 1.5, the thrift server is configured to use 2 vCPUs on the Spark master node by default. You can reduce this to 1 vCPU by editing the total-executor- cores argument value in the `/etc/init.d/hive-thriftserver` script, and then restarting the thrift server (`$ sudo service hive-thriftserver restart`).

K8S-1887: A MapR software version alarm is generated, indicating that “One or more services on the node are running an unexpected version.” The alarm includes a “recommended action” to stop and restart the node.

Workaround: You can ignore the alarm and recommended action for container-based HPE Ezmeral Data Fabric.

CUDA and GPU Issues (Prior Releases)

The following issue applies to HPE Ezmeral Runtime Enterprise relese 5.3.5 and later.

EZESC-964: CUDA applications fail to run on A100 GPU HGX hosts with NVIDIA NVLink switches

Symptom: CUDA applications fail to run on A100 GPU HGX hosts that have NVIDIA NVLink switches.

Workaround: On A100 GPU HGX systems with NVIDIA NVLink switches, you must install the and configure the NVIDIA Fabric Manager on the system before adding it as a host to HPE Ezmeral Runtime Enterprise.

1. Install the NVIDIA Fabric Manager on the host.

   For instructions, see the Fabric Manager for NVIDIA NVSwitch Systems User Guide (link opens an external website in a new browser tab or window)

2. Change the Fabric Manager service start-up options to ensure that the Manager service is started before the kubelet service:

   In the `[Unit]` section of the `nvidia-fabricmanager.service` file, add the following line:

   ```
   Before=kubelet.service
   ```

   For example:

   ```
   [Unit]
   Description=FabricManager service
   Before=kubelet.service
   After=network-online.target
   Requires=network-online.target
   ```

3. Verify the NVLink switches topology to ensure that “NV12” appears between peer GPUs. This result indicates that all 12 NVLinks are trained and available for full bi-directional bandwidth.

   For example, execute the command: `nvidia-smi topo -m`

   The following is an example of a portion of the output:

   ```
           GPU0    GPU1    GPU2
   GPU0     X      NV12    NV12
   GPU1    NV12     X      NV12
   GPU2    NV12    NV12     X
   ```

4. After you add the host to HPE Ezmeral Runtime Enterprise and to the Kubernetes cluster, verify the CUDA Kubernetes application by doing the following:

last-updated: May 06, 2024

a. Create a test pod: `kubectl create -f cuda-test.yaml`

   For example, the following pod executes the `nvidia/samples:vectoradd-cuda10.2` test:

```
apiVersion: v1
kind: Pod
metadata:
  name: nvidia-cuda-test
spec:
  restartPolicy: OnFailure
  containers:
  - name: cuda-vector-add
    image: "nvidia/
samples:vectoradd-cuda10.2"
    resources:
      limits:
         nvidia.com/gpu: 1
```

b. Verify that the test passed by executing the following command:

```
kubectl logs nvidia-cuda-test
```

   Example result:

```
[Vector addition of 50000
elements]
Copy input data from the host
memory to the CUDA device
CUDA kernel launch with 196
blocks of 256 threads
Copy output data from the CUDA
device to the host memory
Test PASSED
Done
```

For more information about the installing and configuring the Fabric Manager, the following NVIDIA documentation (link opens an external website in a new browser tab or window):

- NVIDIA HGX A100 Software User Guide
- Fabric Manager for NVIDIA NVSwitch Systems User Guide

Installation Instructions

Describes where to find installation instructions for this release.

For information about installing HPE Ezmeral Runtime Enterprise, see Planning the Deployment on page 803 and Installation Overview on page 837.

Upgrade Information

This topic describes where to find instructions for upgrading from previous releases of HPE Ezmeral Runtime Enterprise.

This topic describes information about upgrading from previous releases of HPE Ezmeral Runtime Enterprise.

Upgrade Instructions

For information about upgrading to this version of HPE Ezmeral Runtime Enterprise, including the supported upgrade paths from previous releases, see Upgrading to HPE Ezmeral Runtime Enterprise 5.6.x on page 885.

Related Information

The latest documentation for HPE Ezmeral Runtime Enterprise is available at:

https://docs.containerplatform.hpe.com

For HPE Ezmeral Runtime Enterprise Air Gap Utility release notes, see HPE Ezmeral Runtime Enterprise Air Gap Utility Release Notes on page 53.

HPE Ezmeral Runtime Enterprise Air Gap Utility Release Notes

Change history and version compatibility information for the HPE Ezmeral Runtime Enterprise Air Gap Utility, `hpe-airgap-util`, on HPE Ezmeral Runtime Enterprise.

Description

The HPE Ezmeral Runtime Enterprise Air Gap Utility, `hpe-airgap-util`, is utility you can use to query, filter, and download all air gap container images necessary for your HPE Ezmeral Runtime Enterprise environment to a local filesystem or remote registry.

Supersede Information

Utility Version	Supersedes Version
1.3	1.1, 1.2, and 1.0
1.0	0.4 and 0.3

Operating Systems

This utility is supported on the operating systems listed in Using the Air Gap Utility on page 869.

Languages

Languages supported for this release: English

Compatibility and Interoperability

HPE Ezmeral Runtime Enterprise Air Gap Utility Version	HPE Ezmeral Runtime Enterprise Releases
1.3	5.5.0 and later releases, until superseded by a newer version of HPE Ezmeral Runtime Enterprise Air Gap Utility
1.0	5.4.1 and later releases, until superseded by a newer version of HPE Ezmeral Runtime Enterprise Air Gap Utility
0.4	5.4.0

Change Log

HPE Ezmeral Runtime Enterprise Air Gap Utility 1.3:

- Improved query speed for listing releases and images for each release.
- `--list_releases` output now shows whether the release is RC or GA.
- Added the ability to change the log directory by setting the `AIRGAP_UTIL_LOGDIR` environment variable.

HPE Ezmeral Runtime Enterprise Air Gap Utility 1.0:

- Added support for Python 2.7.
- Added the `hpe-airgap-util --version` command.
- Changed the name of the `--version` filter to: `--release`
- Changed the name of the `--list_versions` filter to: `--list_releases`
- Added the `--list_components` filter.
- Added the ability to download a single image file with the `--image` filter.
- Added support for logging and accessing log files.

HPE Ezmeral Runtime Enterprise Air Gap Utility 0.4:

- Set the correct container name and tag when saving to a file.

HPE Ezmeral Runtime Enterprise Air Gap Utility 0.3:

- Creates the destination directory if the specified directory does not exist.
- Reduced processing time for filtering images.
- Fixed image file compression when using the `--dest_compress` parameter is used.

Installation Instructions

See Using the Air Gap Utility on page 869.

Related Information

The latest documentation for HPE Ezmeral Runtime Enterprise is available at:

https://docs.containerplatform.hpe.com

For HPE Ezmeral Runtime Enterprise release notes, see Release Notes on page 11.

Support Matrixes

This section provides information about support and interoperability for HPE Ezmeral Runtime Enterprise and its components.

This section provides information about support and interoperability for HPE Ezmeral Runtime Enterprise and its components.

This information supplements information about system requirements. See System Requirements on page 808

OS Versions

See OS Support on page 85.

GPU Support

For hardware support, see the following table. For software, driver, and MIG support, see GPU and MIG Support on page 721.

HPE Ezmeral Runtime Enterprise Release Version	GPU Hardware
5.4.0-5.6.4	NVIDIA Tesla K80 (AWS environment[1]) NVIDIA Tesla P4 NVIDIA Tesla T4 NVIDIA Tesla P100 NVIDIA Tesla V100[1] NVIDIA Quadro P4000[1] NVIDIA A30 with MIG mode NVIDIA A100 with MIG mode[1]
5.3.5-5.3.6	NVIDIA Tesla P4 NVIDIA Tesla P100 NVIDIA Tesla V100 NVIDIA A100 with MIG mode
5.3.1	NVIDIA Tesla P4 NVIDIA Tesla P100 NVIDIA Tesla V100 NVIDIA Tesla A100 (non-MIG mode only)

Fully tested

HPE Ezmeral Runtime Enterprise Components

Table

HPE Ezmeral Runtime Enterprise Release Version	HPECP Agent	HPE Kubectl Plugin	Kubectl Client	Container Runtime (All OS)
5.6.4	1.3.1-7e1d6f6-04f50a2	3.7-18	1.23.10-hpe1	containerd 1.6.9.hpe-1, Docker CE 19.03 (On Controller, Shadow Controller, Arbiter, and Gateway hosts, and legacy Kubernetes hosts from upgraded deployments only.)

[1] Fully tested

last-updated: May 06, 2024

Table (Continued)

HPE Ezmeral Runtime Enterprise Release Version	HPECP Agent	HPE Kubectl Plugin	Kubectl Client	Container Runtime (All OS)
5.6.2	1.3.1-7e1d6f6-04f50a2	3.7-18	1.23.10-hpe1	containerd 1.6.9.hpe-1, Docker CE 19.03 (On Controller, Shadow Controller, Arbiter, and Gateway hosts, and legacy Kubernetes hosts from upgraded deployments only.)
5.6.1	1.3.1-7e1d6f6-04f50a2	3.7-18	1.23.10-hpe1	containerd 1.6.9.hpe-1, Docker CE 19.03 (On Controller, Shadow Controller, Arbiter, and Gateway hosts, and legacy Kubernetes hosts from upgraded deployments only.)
5.6.0	1.3.0-02f5d92-2e67bb4	3.6	1.20.2	containerd 1.6.9.hpe-1, Docker CE 19.03 (On Controller, Shadow Controller, Arbiter, and Gateway hosts, and legacy Kubernetes hosts from upgraded deployments only.)
5.5.1	1.2.7-234eba4-6155d23	3.6	1.20.2	containerd 1.5.1.hpe-1, Docker CE 19.03 (On Controller, Shadow Controller, Arbiter, and Gateway hosts, and legacy Kubernetes hosts from upgraded deployments only.)
5.5.0	1.2.5-5df266a-6155d23	3.6	1.20.2	containerd 1.5.1.hpe-1, Docker CE 19.03 (On Controller, Shadow Controller, Arbiter, and Gateway hosts, and legacy Kubernetes hosts from upgraded deployments only.)
5.4.1	1.2.2	3.5.13	1.20.0	Docker CE 19.03
5.4.0	1.2.1	3.5.13	1.20.0	Docker CE 19.03
5.3.5-5.3.6	1.1.13	3.4-14	1.20.0	Docker CE 19.03
5.3.1	1.1.5	3.4-14	1.20.0	Docker CE 19.03

HPE Ezmeral Data Fabric

HPE Ezmeral Runtime Enterprise supports the use of different implementations of HPE Ezmeral Data Fabric. Depending on the release version, a given implementation of HPE Ezmeral Data Fabric can be connected as external storage, registered as Tenant/Persistent storage, or both.

last-updated: May 06, 2024

In summary, the implementations of HPE Ezmeral Data Fabric are the following:

HPE Ezmeral Data Fabric on Bare Metal

HPE Ezmeral Data Fabric on Bare Metal is an implementation of HPE Ezmeral Data Fabric that is on physical or virtual machines that are not part of the HPE Ezmeral Runtime Enterprise deployment.

HPE Ezmeral Data Fabric on Bare Metal is the only supported implementation of HPE Ezmeral Data Fabric for production deployments of HPE Ezmeral Runtime Enterprise. HPE Ezmeral Data Fabric on Bare Metal is also supported for non-production deployments.

HPE Ezmeral Data Fabric on Kubernetes

HPE Ezmeral Data Fabric on Kubernetes is an implementation of HPE Ezmeral Data Fabric in a Kubernetes cluster.

HPE Ezmeral Data Fabric on Kubernetes is available for use in non-production deployments of HPE Ezmeral Runtime Enterprise, but it is not supported for production environments.

Embedded Data Fabric

Embedded Data Fabric is a legacy implementation of HPE Ezmeral Data Fabric that is locally **Embedded** and runs on HPE Ezmeral Runtime Enterprise hosts.

Embedded Data Fabric is not supported on 5.5.0 and later releases of HPE Ezmeral Runtime Enterprise. Hewlett Packard Enterprise recommends using HPE Ezmeral Data Fabric on Bare Metal for production deployments. For non-production deployments, you can use either HPE Ezmeral Data Fabric on Bare Metal or HPE Ezmeral Data Fabric on Kubernetes.

In an HPE Ezmeral Runtime Enterprise deployment, only **one** HPE Ezmeral Data Fabric instance can be registered as Tenant/Persistent storage.

Table

HPE Ezmeral Runtime Enterprise Release Version	HPE Ezmeral Data Fabric on Bare Metal	HPE Ezmeral Data Fabric on Kubernetes[1]	Embedded Data Fabric[2]
5.6.4	7.2	1.5.2	Not supported
5.6.2	7.2	1.5.2	Not supported
5.6.1	7.2	1.5.2	Not supported
5.6.0	7.0 and 6.2	1.5.2	Not supported
5.5.0-5.5.1	7.0 and 6.2	1.5.2	Not supported
5.4.1	6.2	1.5.1	Not supported for new deployments
5.4.0	6.2	1.5.0	Not supported for new deployments
5.3.5-5.3.6	Not supported	1.4.1	Discouraged [3]
5.3.1	Not supported	1.4.1	Discouraged [3]

[1] Beginning with HPE Ezmeral Runtime Enterprise 5.5.0, HPE Ezmeral Data Fabric on Kubernetes is supported in non-production environments only.

[2] Embedded Data Fabric aligns with HPE Ezmeral Data Fabric on bare metal Core 6.1.

last-updated: May 06, 2024

[3] "Discouraged" means that Hewlett Packard Enterprise strongly recommends that you upgrade to the latest release of HPE Ezmeral Runtime Enterprise (5.4.x, 5.5.0 or later) and migrate existing Embedded Data Fabric deployments toHPE Ezmeral Data Fabric on Bare Metal, or (for non-production deployments only) HPE Ezmeral Data Fabric on Kubernetes. For more information, contactHewlett Packard Enterprise Technical Support.

Table

HPE Ezmeral Runtime Enterprise	HPE Ezmeral Data Fabric on Bare Metal
5.6.4	7.2 and 6.2.0 (As tenant/persistent storage and as external storage) 6.1.0 (As external storage only)
5.6.2	7.2 and 6.2.0 (As tenant/persistent storage and as external storage) 6.1.0 (As external storage only)
5.6.1	7.2 and 6.2.0 (As tenant/persistent storage and as external storage) 6.1.0 (As external storage only)
5.6.0	7.0 and 6.2.0 (As tenant/persistent storage and as external storage) 6.1.0 (As external storage only)
5.5.0-5.5.1	7.0 and 6.2.0 (As tenant/persistent storage and as external storage) 6.1.0 (As external storage only)
5.4.0-5.4.1	6.2.0 (As tenant/persistent storage and as external storage) 6.1.0 (As external storage only)
5.3.1, 5.3.5, or 5.3.6 (with HPE Ezmeral Data Fabric on Kubernetes version 1.4.1)	6.1.0 or 6.2.0

Table

HPE Ezmeral Runtime Enterprise Release Version	HPE Ezmeral Data Fabric on Kubernetes Version	Core HPE Ezmeral Data Fabric Components	CSI	OS Version (Base Containers)	Alpine Version (Operators)	Open LDAP
5.6.4	1.5.2	6.2.0.23	FUSE POSIX CSI v1.2.8 (Default) NFS-loopback CSI v1.0.7	Rocky Linux release 8.4	Alpine Linux v3.13	1.5.0
5.6.2	1.5.2	6.2.0.23	FUSE POSIX CSI v1.2.8 (Default) NFS-loopback CSI v1.0.7	Rocky Linux release 8.4	Alpine Linux v3.13	1.5.0

Table (Continued)

HPE Ezmeral Runtime Enterprise Release Version	HPE Ezmeral Data Fabric on Kubernetes Version	Core HPE Ezmeral Data Fabric Components	CSI	OS Version (Base Containers)	Alpine Version (Operators)	Open LDAP
5.6.1	1.5.2	6.2.0.23	FUSE POSIX CSI v1.2.7 (Default) NFS-loopback CSI v1.0.7	Rocky Linux release 8.4	Alpine Linux v3.13	1.5.0
5.6.0	1.5.2	6.2.0.23	FUSE POSIX CSI v1.2.7 (Default) NFS-loopback CSI v1.0.7	Rocky Linux release 8.4	Alpine Linux v3.13	1.5.0
5.5.0-5.5.1	1.5.2	6.2.0.23	FUSE POSIX CSI v1.2.7 (Default) NFS-loopback CSI v1.0.5	Rocky Linux release 8.4	Alpine Linux v3.13	1.5.0
5.4.1	1.5.1	6.2.0.18	FUSE POSIX CSI v1.2.5 (Default) NFS-loopback CSI v1.0.5	Rocky Linux release 8.4 (Green Obsidian)	3.13.5	1.5.0
5.4.0	1.5.0	6.2.0.11	FUSE POSIX CSI v1.2.5 (Default) NFS-loopback CSI v1.0.5	Rocky Linux release 8.4 (Green Obsidian)	3.13.5	1.5.0
5.3.5-5.3.6	1.4.1	6.206 EBF	FUSE POSIX CSI v1.2.5 (Default) NFS-loopback CSI v1.0.5	CentOS 8.3.2011	3.13	1.5
5.3.1	1.4.1	6.206 EBF	FUSE POSIX CSI v1.2.1 (Default) NFS-loopback CSI v1.0.1	CentOS 8.3.2011	3.13	1.5

See also:

- Configuring Cross-Cluster Trust on page 652
- Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595
- Using the CSI on page 634
- NFS Support on page 714

Kubernetes Versions

The supported versions of Kubernetes vary by host OS and whether the Kubernetes cluster is imported into HPE Ezmeral Runtime Enterprise. See HPE Ezmeral Runtime Enterprise Components on page 55 and Version Requirements for Imported Kubernetes Clusters on page 61.

In some cases, you can update Kubernetes to later versions than listed for yourHPE Ezmeral Runtime Enterprise release version without upgrading HPE Ezmeral Runtime Enterprise. See Kubernetes Bundles on page 97.

For additional Kubernetes requirements, see also:

- Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595
- The Kubernetes System Requirements on page 808
- Kubernetes Cluster Types and Compatibility on page 322

Table

HPE Ezmeral Runtime Enterprise Release Version	Kubernetes versions for RHEL/ CentOS hosts	Kubernetes versions for SLES hosts
5.6.4	1.24.8-hpe2, 1.25.12-hpe1 and 1.26.7-hpe1	1.24.8-hpe2, 1.25.12-hpe1 and 1.26.7-hpe1
5.6.2		1.22.15-hpe4, 1.23.16-hpe1 and 1.24.10-hpe1
5.6.1	1.22.15-hpe4, 1.23.16-hpe1 and 1.24.10-hpe1	1.22.15-hpe4, 1.23.16-hpe1 and 1.24.10-hpe1
5.6.0	1.22.15-hpe3, 1.23.14-hpe2 and 1.24.8-hpe2	1.22.15-hpe3, 1.23.14-hpe2 and 1.24.8-hpe2
5.5.1	1.22.15-hpe1, 1.23.13-hpe1 1.21.14-hpe2 (Supported during migration only[3]) 1.21.14, 1.22.11, 1.23.8 (Supported during migration only[2])	1.22.15-hpe1, 1.23.13-hpe1 1.21.14-hpe2 (Supported during migration only[3]) 1.21.14, 1.22.11, 1.23.8 (Supported during migration only[2])
5.5.0	1.22.12-hpe1, 1.23.9-hpe1 1.21.14-hpe1 (Supported during migration only[1]) 1.21.14, 1.22.11, 1.23.8 (Supported during migration only[2])	1.22.12-hpe1, 1.23.9-hpe1 1.21.14-hpe1 (Supported during migration only[1]) 1.21.14, 1.22.11, 1.23.8 (Supported during migration only[2])
5.4.1	1.19.15, 1.20.11, 1.21.10	Not Supported
5.4.0	1.19.15, 1.20.11, 1.21.3	Not Supported
5.3.6	1.18.6, 1.19.15, 1.20.11	1.18.6 (CaaS 4.5)
5.3.5	1.18.6, 1.19.5, 1.20.2	1.18.6 (CaaS 4.5)
5.3.1	1.18.6, 1.19.5, 1.20.2	1.18.6 (CaaS 4.5)

[1] Supported during migration of existing Kubernetes clusters to 1.23.9-hpe1 (preferred) or to 1.22.12-hpe1. Not supported for new Kubernetes clusters or for ongoing operations in a production deployment.

[2] Supported for existing Kubernetes clusters until those legacy clusters are migrated to a Hewlett Packard Enterprise distribution of Kubernetes. You cannot upgrade a legacy Kubernetes cluster without also

migrating the cluster to a Hewlett Packard Enterprise distribution of Kubernetes. Not supported for new Kubernetes clusters or for ongoing operations in a production deployment.

[3] Supported during migration of existing Kubernetes clusters to 1.23.13-hpe1 (preferred) or to 1.22.15-hpe1. Not supported for new Kubernetes clusters or for ongoing operations in a production deployment.

Imported Kubernetes Clusters

HPE Ezmeral Runtime Enterprise 5.6.x or higher does not support Imported Kubernetes clusters.

HPE Ezmeral Runtime Enterprise Release Version	Imported Kubernetes clusters
5.6.4	Not Supported
5.6.2	Not Supported
5.5.0-5.5.1	Not Supported
5.4.0-5.4.1	Not Supported
5.3.5-5.3.6	Supported. See Table 7: Imported Cluster Types and Versions on page 61
5.3.1	Supported. See Table 7: Imported Cluster Types and Versions on page 61

NOTICE: End of Life (EOL) for Elastic Private Instant Clusters (EPIC)

HPE Ezmeral Runtime Enterprise 5.4.1 is the last release that includes support for EPIC. Beginning with the next general availability release, deployments that use EPIC to manage virtual nodes/containers are not supported. No future enhancements to EPIC are planned; however, support (such as bug fixes) will continue to be provided until the EPIC functionality reaches End of Life (EOL).

Existing deployments that use EPIC can be transitioned to the newer Kubernetes-based solution on the latest HPE Ezmeral Runtime Enterprise release. Existing deployments that continue to use EPIC will be supported until EPIC reaches End of Life (EOL) on December 30, 2024.

Version Requirements for Imported Kubernetes Clusters

Imported Kubernetes clusters are not supported on HPE Ezmeral Runtime Enterprise 5.4.x, 5.5.x or later releases.

HPE Ezmeral Runtime Enterprise 5.2.x and 5.3.x releases support the following Kubernetes versions for clusters that are imported into the platform.

NOTE:

Import is supported for Kubernetes versions shown in the following table, even if the version is not one of the versions supported for clusters created by HPE Ezmeral Runtime Enterprise.

Imported clusters do not support add-ons.

Table

Kubernetes Service	Supported Versions for Imported Clusters
Amazon Elastic Kubernetes Service (EKS)	1.18, 1.19
Google Kubernetes Engine (GKE)	1.18.16-gke-302, 1.18.16-gke-502
Azure Kubernetes Service (AKS)	1.18.14, 1.19.7, 1.20.2
VMware Tanzu Kubernetes Grid (PKS)	1.7

Add-On Versions

The following table lists the versions of system and application add-ons by HPE Ezmeral Runtime Enterprise release. KubeDirector is a prerequisite for the analytics and HPE Ezmeral ML Ops add-ons.

In some cases, you can update add-ons to later versions without upgrading HPE Ezmeral Runtime Enterprise to a later version. See Kubernetes Bundles on page 64.

For version information about the container runtimes, HPECP agent, and kubectl, see HPE Ezmeral Runtime Enterprise Components on page 55.

Table

HPE Ezmeral Runtime Enterprise Release Version	KubeDirector	Apache Spark	Kubeflow	Airflow	MLflow
5.6.4	0.11.0-9726ef0	1.3.8.0 (Contains Apache Spark 2.4.7 and Apache Spark 3.3.1)	Not prepackaged as add-on. Contact HPE Technical Support team for more details.	Not prepackaged as add-on. Contact HPE Technical Support team for more details.	N/A (Replaced by integrated model management framework.)
5.6.2	0.11.0-9726ef0	1.3.8.0 (Contains Apache Spark 2.4.7 and Apache Spark 3.3.1)	1.7-0fc57da (Contains Kubeflow 1.6-0fc57da)	2.3-2b38953 (Contains Airflow 2.4.3)	N/A (Replaced by integrated model management framework.)
5.6.1	0.11.0-9726ef0	1.3.8.0 (Contains Apache Spark 2.4.7 and Apache Spark 3.3.1)	1.7-0fc57da (Contains Kubeflow 1.6-0fc57da)	2.3-2b38953 (Contains Airflow 2.4.3)	N/A (Replaced by integrated model management framework.)
5.6.0	0.11.0-9726ef0	1.3.8.0 (Contains Apache Spark 2.4.7 and Apache Spark 3.3.1)	1.7-0fc57da (Contains Kubeflow 1.6-0fc57da)	2.3-2b38953 (Contains Airflow 2.4.3)	N/A (Replaced by integrated model management framework.)
5.5.1	0.11.0-9726ef0	1.3.8.0-14ea66c-5455643 (Contains Apache Spark 2.4.7 and Apache Spark 3.2.0)	1.6-d475bce	2.3-2b38953 (Contains Airflow 2.4.3)	N/A (Replaced by integrated model management framework.)
5.5.0	0.11.0-9726ef0	1.3.7.1-1a83a98-5455643 (Contains Apache Spark 2.4.7 and Apache Spark 3.2.0)	1.6-fd89c89	2.2-ce70c0b (Contains Airflow 2.3.4)	N/A (Replaced by integrated model management framework.)
5.4.1	0.9.0	Apache Spark 2.4.7 and Apache Spark 3.1.2	1.3*	Airflow 2.2.5	KDApp MLflow 1.5, includes MLFlow Tracking 1.12.0 (Technical Preview)

Table (Continued)

HPE Ezmeral Runtime Enterprise Release Version	KubeDirector	Apache Spark	Kubeflow	Airflow	MLflow
5.4.0	0.8.1	Apache Spark 2.4.7 and Apache Spark 3.1.2	1.3*	Airflow 2.2.0	KDApp MLflow 1.5, includes MLFlow Tracking 1.12.0 (Technical Preview)
5.3.5-5.3.6	0.6.2	Apache Spark 2.4.7 and Apache Spark 3.1.1 (Preview)	1.2 (With Istio 1.3.1)	Airflow 2.0.1	KDApp MLflow 1.5, includes MLFlow Tracking 1.12.0 (Technical Preview)
5.3.1	0.6.1	Apache Spark 2.4.7 and Apache Spark 3.1.1 (Preview)	1.2 (With Istio 1.3.1)	Airflow 2.0	KDApp MLflow 1.5, includes MLFlow Tracking 1.12.0 (Technical Preview)

* Istio is a prerequisite for Kubeflow 1.3 and later. However, Istio 1.9.8, which is the version of Istio that is shipped with HPE Ezmeral Runtime Enterprise 5.4.0 or 5.4.1, is not supported on Kubernetes 1.21 and later. HPE Ezmeral Runtime Enterprise prevents existing clusters that have Istio 1.9.8 from being upgraded to Kubernetes 1.21.x.

Table

HPE Ezmeral Runtime Enterprise Release Version	Istio Service Mesh	Argo CD	Falco	Open Policy Agent Gatekeeper	Data Fabric Tenant Operator	NVIDIA plug-in
5.6.4	1.14.5	2.5.2-04f50a2	0.33.0-04f50a2	3.10.0-04f50a2	picasso-1.5.2-GA-ERE56-drop1-75-0	0.12.3-04f50a2
5.6.2	1.14.5	2.5.2-04f50a2	0.33.0-04f50a2	3.10.0-04f50a2	picasso-1.5.2-GA-ERE56-drop1-75-0	0.12.3-04f50a2
5.6.1	1.14.5	2.5.2-04f50a2	0.33.0-04f50a2	3.10.0-04f50a2	picasso-1.5.2-GA-ERE56-drop1-75-0	0.12.3-04f50a2
5.6.0	1.14.5	2.5.2-ba31668	0.33.0-b9eff56	3.10.0-d006b58	picasso-1.5.2-GA-ERE56-drop1-75-0	0.12.3-d006b58
5.5.1	1.13.5-e3f62ee	2.4.17-3c54efb	0.32.2-a3473ed	3.10.0-58c0a78	picasso-1.5.2-drop1-74-0	0.12.3-4ae1978
5.5.0	1.13.5-547eb22	2.2.5-ad3d17b	0.32.2-a3473ed	3.9.0-ad3d17b	picasso-1.5.2-drop1-74-0	0.12.3-4ae1978
5.4.1	1.9.8*	ArgoCD 2.2.5	Falco 2.23.1	3.7.0-7	picasso-1.5.1-P151RC2-71-0	0.9.0
5.4.0	1.9.8*	ArgoCD 2.2.5	Falco 2.23.1	3.7.0-4	picasso-1.5.0-P150RC10-69-0	0.9.0
5.3.5-5.3.6	1.9.0	ArgoCD 1.8.4	Falco 2.21.0	3.3.0-2	picasso-1.4.1-drop7-43-2	0.9.0

last-updated: May 06, 2024

Table (Continued)

HPE Ezmeral Runtime Enterprise Release Version	Istio Service Mesh	Argo CD	Falco	Open Policy Agent Gatekeeper	Data Fabric Tenant Operator	NVIDIA plug-in
5.3.1	1.7.1	ArgoCD 1.8.4	Falco 2.21.0	3.3.0-2	picasso-1.4.1-drop7-43-1	1.0.0-beta-6

* Istio is a prerequisite for Kubeflow 1.3 and later. However, Istio 1.9.8, which is the version of Istio that is shipped with HPE Ezmeral Runtime Enterprise 5.4.0 or 5.4.1, is not supported on Kubernetes 1.21 and later. HPE Ezmeral Runtime Enterprise prevents existing clusters that have Istio 1.9.8 from being upgraded to Kubernetes 1.21.x.

Table

HPE Ezmeral Runtime Enterprise Release Version	Monitoring	NVIDIA GPU Metrics	ERE Service Accounts	Kube State Metrics	Kubernetes Dashboard	Metrics Server
5.6.4	6.6.7-04f50a2	6.6.3-04f50a2	0.2-04f50a2	6.6.3-a4fbfa3	v2.7.0-04f50a2	v0.6.2-04f50a2
5.6.2	6.6.7-04f50a2	6.6.3-04f50a2	0.2-04f50a2	6.6.3-a4fbfa3	v2.7.0-04f50a2	v0.6.2-04f50a2
5.6.1	6.6.7-04f50a2	6.6.3-04f50a2	0.2-04f50a2	6.6.3-a4fbfa3	v2.7.0-04f50a2	v0.6.2-04f50a2
5.6.0	6.6.7-b9eff56	6.6.3-a4fbfa3	0.2-da188d2	6.6.3-a4fbfa3	v2.7.0-0349604	v0.6.2-15bd0b2
5.5.1	6.6.7-6155d23	6.5.5-6155d23	0.2-6155d23	2.3.0-db64c34	v2.7.0-a650c50	v0.6.1-6155d23
5.5.0	6.6.7-6155d23	6.5.5-6155d23	0.2-6155d23	2.3.0-db64c34	v2.7.0-a650c50	v0.6.1-6155d23
5.4.1	6.6.5-8.0	6.5.5	0.2-4	1.9.6-2	2.0.0-rc2-4	0.3.6-4
5.4.0	6.6.5-8.0	6.5.5	0.2-4	1.9.6-2	2.0.0-rc2-3	0.3.6-4
5.3.5-5.3.6	6.6.5-7.0	6.5.5	0.2-2	1.9.6-2	2.0.0-rc2-2	0.3.6-4
5.3.1	6.6.5-7.0	6.5.5	0.2-2	1.9.6-2	2.0.0-rc2-2	0.3.6-4

Kubernetes Bundles

The following table lists the versions and content of Kubernetes bundles and their compatible versions of HPE Ezmeral Runtime Enterprise. For more information about Kubernetes bundles, see Kubernetes Bundles on page 97.

Table

Kubernetes Bundle Version	Compatible HPE Ezmeral Runtime Enterprise Versions	Kubernetes Versions	Add-On Versions	EOL Date
2.1.2[2]	5.6.4	1.24.8-hpe2, 1.25.12-hpe1 and 1.26.7-hpe1	Airflow (2.3-04f50a2) ArgoCD (2.5.2-04f50a2) ERE Service Accounts (0.2-04f50a2) Falco (0.33.0-04f50a2) HPECP Agent (1.3.1-7e1d6f6-04f50a2) Istio (1.14.5-04f50a2) KubeDirector (0.11.0-04f50a2) Kubeflow (1.6-5.6.1-67dcdba) Kubernetes Dashboard (v2.7.0-04f50a2) Metrics Server (v0.6.2-04f50a2) Monitoring (6.6.7-04f50a2) NVIDIA GPU Metrics (6.6.3-04f50a2) NVIDIA plugin (0.12.3-04f50a2) Open Policy Agent Gatekeeper (3.10.0-04f50a2) Spark Operator (1.3.8.1-82f8759-04f50a2) Tenant Operator (picasso-1.5.2-GA-ERE56-drop1-75-0) hpecp-bootstrap-prometheus (41.7.4-04f50a2)	25 JAN 2024

Table (Continued)

Kubernetes Bundle Version	Compatible HPE Ezmeral Runtime Enterprise Versions	Kubernetes Versions	Add-On Versions	EOL Date
2.1.2[3]	5.6.2	1.22.15-hpe4 1.23.16-hpe1 1.24.10-hpe1	Airflow (2.3-04f50a2) ArgoCD (2.5.2-04f50a2) ERE Service Accounts (0.2-04f50a2) Falco (0.33.0-04f50a2) HPECP Agent (1.3.1-7e1d6f6-04f50a2) Istio (1.14.5-04f50a2) KubeDirector (0.11.0-04f50a2) Kubeflow (1.6-5.6.1-67dcdba) Kubernetes Dashboard (v2.7.0-04f50a2) Metrics Server (v0.6.2-04f50a2) Monitoring (6.6.7-04f50a2) NVIDIA GPU Metrics (6.6.3-04f50a2) NVIDIA plugin (0.12.3-04f50a2) Open Policy Agent Gatekeeper (3.10.0-04f50a2) Spark Operator (1.3.8.1-82f8759-04f50a2) Tenant Operator (picasso-1.5.2-GA-ERE56-drop1-75-0) hpecp-bootstrap-prometheus (41.7.4-04f50a2)	

[2] Included in the HPE Ezmeral Runtime Enterprise 5.6.4 package.

Table (Continued)

Kubernetes Bundle Version	Compatible HPE Ezmeral Runtime Enterprise Versions	Kubernetes Versions	Add-On Versions	EOL Date
2.1.1[4]	5.6.1	1.22.15-hpe4 1.23.16-hpe1 1.24.10-hpe1	Airflow (2.3-04f50a2) ArgoCD (2.5.2-04f50a2) ERE Service Accounts (0.2-04f50a2) Falco (0.33.0-04f50a2) HPECP Agent (1.3.1-7e1d6f6-04f50a2) Istio (1.14.5-04f50a2) KubeDirector (0.11.0-04f50a2) Kubeflow (1.6-5.6.1-67dcdba) Kubernetes Dashboard (v2.7.0-04f50a2) Metrics Server (v0.6.2-04f50a2) Monitoring (6.6.7-04f50a2) NVIDIA GPU Metrics (6.6.3-04f50a2) NVIDIA plugin (0.12.3-04f50a2) Open Policy Agent Gatekeeper (3.10.0-04f50a2) Spark Operator (1.3.8.1-82f8759-04f50a2) Tenant Operator (picasso-1.5.2-GA-ERE56-drop1-75-0) hpecp-bootstrap-prometheus (41.7.4-04f50a2)	

[3] Included in the HPE Ezmeral Runtime Enterprise 5.6.2 package.

Table (Continued)

Kubernetes Bundle Version	Compatible HPE Ezmeral Runtime Enterprise Versions	Kubernetes Versions	Add-On Versions	EOL Date
2.0.0[5]	5.6.0	1.22.15-hpe3 1.23.14-hpe2 1.24.8-hpe2	Airflow (2.3-67a049e) ArgoCD (2.5.2-ba31668) ERE Service Accounts (0.2-da188d2) Falco (0.33.0-b9eff56) HPECP Agent (1.3.0-8bf3031-2e67bb4) Istio (1.14.5-d006b58) KubeDirector (0.11.0-7c59acf) Kubeflow (1.6-5.6.0-0c86715) Kubernetes Dashboard (v2.7.0-0349604) Metrics Server (v0.6.2-15bd0b2) Monitoring (6.6.7-b9eff56) NVIDIA GPU Metrics (6.6.3-a4fbfa3) NVIDIA plugin (0.12.3-d006b58) Open Policy Agent Gatekeeper (3.10.0-d006b58) Spark Operator (1.3.8.0-e69e44a-d88da3c) Tenant Operator (picasso-1.5.2-GA-ERE56-drop1-75-0) hpecp-bootstrap-prometheus (41.7.4-d49d66c)	

[4] Included in the HPE Ezmeral Runtime Enterprise 5.6.1 package.

Table (Continued)

Kubernetes Bundle Version	Compatible HPE Ezmeral Runtime Enterprise Versions	Kubernetes Versions	Add-On Versions	EOL Date
2.0.0[6]	5.5.1	1.22.15-hpe3 1.23.14-hpe2 1.24.8-hpe2	Airflow (2.3-67a049e) ArgoCD (2.5.2-ba31668) ERE Service Accounts (0.2-da188d2) Falco (0.33.0-b9eff56) HPECP Agent (1.3.0-8bf3031-2e67bb4) Istio (1.14.5-d006b58) KubeDirector (0.11.0-7c59acf) Kubeflow (1.6-5.6.0-0c86715) Kubernetes Dashboard (v2.7.0-0349604) Metrics Server (v0.6.2-15bd0b2) Monitoring (6.6.7-b9eff56) NVIDIA GPU Metrics (6.6.3-a4fbfa3) NVIDIA plugin (0.12.3-d006b58) Open Policy Agent Gatekeeper (3.10.0-d006b58) Spark Operator (1.3.8.0-e69e44a-d88da3c) Tenant Operator (picasso-1.5.2-GA-ERE56-drop1-75-0) hpecp-bootstrap-prometheus (41.7.4-d49d66c)	

[5] Included in the HPE Ezmeral Runtime Enterprise 5.6.0 package.

Table (Continued)

Kubernetes Bundle Version	Compatible HPE Ezmeral Runtime Enterprise Versions	Kubernetes Versions	Add-On Versions	EOL Date

1.0.4[7]	5.5.1, 5.5.0	`1.21.14-hpe2`
		`1.22.15-hpe1`
		`1.23.13-hpe1`
			`Airflow (2.3-2b38953)`
			`ArgoCD (2.4.17-3c54efb)`
			`ERE Service Accounts (0.2-6155d23)`
			`Falco (0.32.2-a3473ed)`
			`HPECP Agent (1.2.7-234eba4-6155d23)`
			`Istio (1.13.5-e3f62ee)`
			`Kube State Metrics (2.3.0-db64c34)`
			`KubeDirector (0.11.0-9726ef0)`
			`Kubeflow (1.6-d475bce)`
			`Kubernetes Dashboard (v2.7.0-a650c50)`
			`Metrics Server (v0.6.1-6155d23)`
			`Monitoring (6.6.7-6155d23)`
			`NVIDIA GPU Metrics (6.5.5-6155d23)`
			`NVIDIA plugin (0.12.3-4ae1978)`
			`Open Policy Agent Gatekeeper (3.10.0-58c0a78)`
			`Spark Operator (1.3.8.0-14ea66c-5455643)`
			`Tenant Operator (picasso-1.5.2-drop1-74-0)`
				22 OCT 2023

[6] Included in the HPE Ezmeral Runtime Enterprise 5.6.0 package.

Table (Continued)

Kubernetes Bundle Version	Compatible HPE Ezmeral Runtime Enterprise Versions	Kubernetes Versions	Add-On Versions	EOL Date
1.0.3[8]	5.5.0	1.21.14-hpe1 1.22.12-hpe1 1.23.9-hpe1	airflow 2.2-ce70c0b argocd 2.2.5-ad3d17b falco 0.32.2-a3473ed hpecp-agent 1.2.5-5df266a-6155d23 hpecp-monitoring 6.6.7-6155d23 hpecp-nvidiagpu beat 6.5.5-6155d23 hpecp-serviceaccounts 0.2-6155d23 istio 1.13.5-547eb22 kube-state-metrics 2.3.0-db64c34 kubedirector 0.11.0-9726ef0 kubeflow 1.6-fd89c89 kubernetes-dashboard v2.7.0-a650c50 metrics-server v0.6.1-6155d23 nvidia-plugin 0.12.3-4ae1978 opa-gatekeeper 3.9.0-ad3d17b picasso picasso-1.5.2-drop1-74-0 spark-operator 1.3.7.1-1a83a98-5455643	22 OCT 2023

Included in the HPE Ezmeral Runtime Enterprise 5.6.4 package.Included in the HPE Ezmeral Runtime Enterprise 5.6.2 package.Included in the HPE Ezmeral Runtime Enterprise 5.6.1 package.Included in the HPE Ezmeral Runtime Enterprise 5.6.0 package.Included in the HPE Ezmeral Runtime Enterprise 5.6.0 package.Included in the HPE Ezmeral Runtime Enterprise 5.5.1 package.Included in the HPE Ezmeral Runtime Enterprise 5.5.0 package.

[7] Included in the HPE Ezmeral Runtime Enterprise 5.5.1 package.
[8] Included in the HPE Ezmeral Runtime Enterprise 5.5.0 package.

Kubernetes Applications

See the following:

- Spark Support on page 247 and Interoperability Matrix for Spark on page 246
- Livy Overview on page 275
- Kubeflow
- Airflow

Kubeflow Components (5.6.x and higher)

Kubeflow Operator version is 1.6.

Beginning with HPE Ezmeral Runtime Enterprise 5.5.1, Kubeflow notebooks are available. However, Hewlett Packard Enterprise recommends that you use full-featured KubeDirector notebooks instead.

The following table lists the components that Kubeflow deploys.

Table

Component	Version in HPE Ezmeral Runtime Enterprise 5.6.x
Cert Manager	1.5.0
Dex	2.31.2
Katib	0.14.0
Kserve	0.8.0
Knative	1.2.5
Kubeflow Dashboard	ecp-5.6.0-release
ML Metadata	2.0.0-alpha.3
ML Pipelines (KFP)	2.0.0-alpha.3
Prism	ecp-5.6.0-release
Profile Controller (KFAM)	1.6.0
Seldon	1.12.0
Tensorboard	ecp-5.6.0-release
Training Operator	1-e1434f6
Volumes Web App	1.6.0
Workflow Controller	3.2.3

Kubeflow Components (5.5.x)

Kubeflow Operator version is 1.6.

Beginning with HPE Ezmeral Runtime Enterprise 5.5.1, Kubeflow notebooks are available. However, Hewlett Packard Enterprise recommends that you use full-featured KubeDirector notebooks instead.

The following table lists the components that Kubeflow deploys.

Table

Component	Version in HPE Ezmeral Runtime Enterprise 5.5.x
Cert Manager	1.5.0

Table (Continued)

Component	Version in HPE Ezmeral Runtime Enterprise 5.5.x
Dex	2.31.2
Katib	0.14.0
Kserve	0.8.0
Knative	1.2.4
Kubeflow Dashboard	ecp-5.5.0-release
ML Metadata	2.0.0-alpha.3
ML Pipelines (KFP)	2.0.0-alpha.3
Prism	ecp-5.5.0-release
Profile Controller (KFAM)	1.6.0
Seldon	1.12.0
Tensorboard	v1.6.0
Training Operator	1-e1434f6
Volumes Web App	1.6.0
Workflow Controller	3.2.3

Kubeflow Components (5.4.0-5.4.1)

The Kubeflow Operator version is: 1.3

The following table lists the components that Kubeflow deploys.

Table

Component	Version in HPE Ezmeral Runtime Enterprise 5.4.0 and 5.4.1
Dex	2.24.0
Katib	0.11.0
Kfserving	0.6.0
Knative	0.22.1
Kubeflow Dashboard	1.3.0-rc.1
ML Metadata	1.5.0
MPI Operator	0.2.3
MXNet Operator	1.1.0
Notebook Controller	1.3.0-rc.1
Kubeflow Pipelines (KFP)	1.5.0
Profile Controller	1.3.0-rc.1
Pytorch	0.7.0
Seldon	1.9.0
Tensorboard Controller	1.3.0-rc.1
TFJob Operator	1.1.0

last-updated: May 06, 2024

Table (Continued)

Component	Version in HPE Ezmeral Runtime Enterprise 5.4.0 and 5.4.1
Volumes Web App	1.3.0-rc.1
XGBoost Operator	0.2.0

Container Image Vulnerabilities and CVE Reports

Describes how HPE Ezmeral Engineering provides software updates to address container image vulnerabilities.

HPE Ezmeral Engineering takes security very seriously and makes every effort to ensure that the container images for HPE Ezmeral software products are free of known vulnerabilities at the time of release. However, because new vulnerabilities are always being discovered and reported, it is likely that scanning product images with tools such as Trivy will show lists of CVEs that affect packages inside the images.

The HPE Ezmeral Engineering team also regularly scans product images to identify new vulnerabilities and creates action plans to modify the git product images. Please note that most vulnerabilities are present in open-source software leveraged by HPE Ezmeral Engineering. Therefore, HPE Ezmeral Engineering determines when it is best to update products with updated open-source content.

HPE Ezmeral Engineering typically updates vulnerable packages from one minor software product version to the next (for example, from 1.3 to 1.4). For critical vulnerabilities, HPE may provide security-patched container images outside of the established software release cycle, in accordance with the following table.

To keep your platform as secure as possible, please ensure that you upgrade or patch your HPE Ezmeral Software to the latest available software.

Severity (CVSS Base Score Range)	SLA of Response
Critical (9.0 – 10.0)	HPE Ezmeral Engineering will prioritize and begin working on a fix. The team will make the fix available as soon as possible. This might take the form of a special maintenance release of an HPE Ezmeral software product for the sole purpose of making the fix available. If it is possible to deploy the fix as a patch more quickly or conveniently, the patch will also be made available. In the meantime, the support team will work with the community to mitigate the issue.
High (7.0 – 8.9)	HPE Ezmeral Engineering will include a fix in the next planned release (major or minor) of the HPE Ezmeral software product. HPE Ezmeral software releases typically happen on a quarterly basis. The fix will be made available in patch form for customers who want to deploy it sooner, and the support team will assist with applying the patch.
Medium (4.0 – 6.9)	HPE Ezmeral Engineering will include a fix in the next planned release (major or minor) of the HPE Ezmeral product.
Low (0.1 – 3.9)	HPE Ezmeral Engineering will include a fix in the next major release of the HPE Ezmeral product, or the team will provide detailed steps that can be taken to mitigate the issue.

Legal Notices

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty.

Acknowledgments

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Google and the Google Logo are registered trademarks of Google LLC. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. NVIDIA® and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries. UNIX® is a registered trademark of The Open Group. VMware® is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All third-party marks are property of their respective owners.

Regulatory Information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center.

- Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found here.
- For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, click here.
- For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, click here.

Support and Other Resources

HPE Ezmeral Websites

- HPE Ezmeral main site
- HPE Ezmeral Marketplace

Developer and User Communities

- HPE Developer Community
- HPE Ezmeral user forum
- Blog: HPE Ezmeral Uncut

last-updated: May 06, 2024

General Websites

- Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix
- Storage white papers and analyst reports
- Security bulletins and vulnerability reports

Accessing Hewlett Packard Enterprise Support

For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website.

To access documentation and support services, go to the Hewlett Packard Enterprise Support Center.

Information to collect:

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing Updates

Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.

- To download product updates, visit the Hewlett Packard Enterprise Support Center.
- Software downloads
- Software Depot
- To subscribe to eNewsletters and alerts, click here.
- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center More Information on Access to Support Materials page.

Documentation Feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, use the **Feedback** button and icons to send any errors, suggestions, or comments. All document information is captured by the process.

Definitions

This article contains two sets of definitions:

- **General:** General terms used with HPE Ezmeral Runtime Enterprise. See General on page 77.

last-updated: May 06, 2024

- **HPE Ezmeral Data Fabric on Kubernetes:** Terms used exclusively when discussing HPE Ezmeral Data Fabric in a Kubernetes environment. See HPE Ezmeral Data Fabric on Kubernetes on page 79.

General

These articles use the following terms (provided in alphabetical order):

- **Active Directory (or AD):** This is a Microsoft directory service for Windows domain networks.
- **Arbiter:** An *Arbiter* is a designated host that triggers the Shadow Controller host to assume the Controller role if the primary Controller host fails.
- **Cluster:** For Kubernetes, a *cluster* is a group of *nodes* (hosts) that each contain one or more *pods*.
- **Big Data/AI application:** A *Big Data application* generally refers to a distributed, multi-node, inter-related service that can process large amounts of data computing on several nodes. Some examples of Big Data and AI applications include Hadoop, Spark, Kafka, TensorFlow, H2O, and others. Big Data/AI applications should not be confused with microservices.
- **cnode:***cnode* is the HPE Ezmeral Runtime Enterprise caching node service, which reduces latency when transferring storage I/O requests to and from the HPE Ezmeral Runtime Enterprise implementation of the HDFS Java client.
- **Compute host (or Compute Worker)** In Kubernetes deployments, a compute host or compute worker is a Kubernetes host that is managed by the Kubernetes control plane and is not used for HPE Ezmeral Data Fabric on Kubernetes storage.
- **Container:** A *container* is a lightweight, standalone, executable software package that runs specific services. An Open Container Initiative (OCI)-compliant container includes code, runtime, system libraries, configurations, and forth, that run as an isolated process in user space. An OCI-compliant container container is typically used to deploy scalable and repeatable *microservices*.
- **Controller host:** A *Controller* is a host that manages the HPE Ezmeral Runtime Enterprise deployment.
- **DataTap:** A *DataTap* is a shortcut that points to a storage resource on the network. A Tenant Administrator creates a DataTap within a tenant and defines the storage namespace that the DataTap represents (such as a directory tree in a file system). A Tenant Member may then access paths within that resource for data input and/or output. Creating and editing DataTaps allows Tenant Administrators to control which storage areas are available to the members of each tenant, including any specific sharing or isolation of data between tenants.
- **Deployment:** Another term for *platform*.
- **Ephemeral storage:** *ephemeral storage* is storage space available for backing the root file systems of hosts in the HPE Ezmeral Runtime Enterprise. Ephemeral storage is not persistent. Contrast with *Tenant storage*.
- **Filesystem Mount (or FS Mount):** A *filesystem mount* enables HPE Ezmeral Runtime Enterprise to automatically add NFS volumes or mounts to Kubernetes clusters. This enables Kubernetes clusters to directly access NFS shares as if they were local directories.
- **Gateway host (or Gateway Worker):** A *Gateway host* or *Gateway Worker* is a host that is managed by a Controller. Each Gateway host in HPE Ezmeral Runtime Enterprise maps services running on containers to ports in order to allow users to access those services
- **HCP Agent:** A custom Kubernetes controller that is installed on every Kubernetes cluster instantiated by HPE Ezmeral Runtime Enterprise. The agent performs key tasks, such as creating or associating namespaces to tenants, creating annotations for mapping NodePort services to Gateways, and creating FS mounts.

last-updated: May 06, 2024

- **Host:** A *host* is either a physical server or a virtual server, located on your premises or in a public cloud, that is available to HPE Ezmeral Runtime Enterprise.
- **HPE Ezmeral Runtime Enterprise:***HPE Ezmeral Runtime Enterprise* consists of the hosts that comprise the overall infrastructure available to create, run, and manage Kubernetes clusters.
- **Kubeconfig:** A file that configures access to Kubernetes when used in conjunction with either the kubectl command line tool or other clients.
- **Kubectl:** A command line tool for controlling a Kubernetes cluster.
- **KubeDirector:** An open source-project designed to simplify running complex stateful scale-out application clusters on Kubernetes. KubeDirector is built using the Kubernetes custom resource definition (CRD) framework and leverages the native Kubernetes API extensions and design philosophy. This enables transparent integration with Kubernetes user/resource management as well as existing clients and tools.
- **Lightweight Directory Access Protocol (LDAP):** This is a client-server directory service protocol that runs on a layer above the TCP/IP stack and provides a mechanism for connecting to, searching, and modifying networked directories.
- **Master node:** An outdated term for the Kubernetes control plane.
- **Microservice:** A *microservice* is a method of developing software applications as a suite of small, modular, and independently deployable services in which each service runs a unique process and communicates through a well-defined, lightweight mechanism to serve a business goal.
- **Node:** For Kubernetes, a *node* is a *host* that is a member of a Kubernetes cluster.
- **Node storage:** See Ephemeral Storage.
- **Platform:** A *platform* includes all of the tenants, projects, nodes, and users that exist on a given HPE Ezmeral Runtime Enterprise deployment. These articles may also use the term *deployment* to refer to "HPE Ezmeral Runtime Enterprise."
- **Platform Administrator:** The *Platform Administrator* (or *Platform Admin*) is an HPE Ezmeral Runtime Enterprise user that has been granted the role of `Site Admin`. A user with this role has the ability to create/delete tenants. This user will typically also be responsible for managing the hosts in the deployment.
- **Pod:** For Kubernetes, a *pod* is a group of containers deployed on a single host.
- **Project (or AI/ML Project):** A *project* or *AI/ML project* is a unit of resource partitioning and data/user access control in a given deployment that is used for running AI/ML workloads in HPE Ezmeral ML Ops. The resources of an HPE Ezmeral Runtime Enterprise deployment are shared among the tenants AI/ML projects on that platform. All users who are a member of an AI/ML project can access the resources and data objects available to that project. This is analogous to a *tenant*, except that a tenant is not pre-configured for AI/ML workloads.
- **Security Assertion Markup Language (SAML):** This is an open standard for exchanging authentication and authorization data between parties, such as between an identity provider (IdP) and a service provider.
- **Shadow Controller host:** A *Shadow Controller host* is a host that assumes the Controller host role if the primary Controller host fails.

last-updated: May 06, 2024

- **Tenant:** A *tenant* is a unit of resource partitioning and data/user access control in a given deployment. The resources of an HPE Ezmeral Runtime Enterprise deployment are shared among the tenants on that platform. All users who are a member of a tenant can access the resources and data objects available to that tenant. If a tenant is used to run HPE Ezmeral ML Ops, then is it called either a *project* or an *AI/ML project*.
- **Tenant Administrator:** A *Tenant Administrator* (or *Tenant Admin*) is a role granted to an HPE Ezmeral Runtime Enterprise user. A user with this role has the ability to manage the specific tenants for which they have been granted this role, including creating DataTaps for that tenant.
- **Tenant Member:** A *Tenant Member* (or *Member*) is a role granted to an HPE Ezmeral Runtime Enterprise user. A user with this role has non-administrative access to the specific tenants for which they have been granted this role. Members may use existing DataTaps for reading and writing data.
- **Tenant storage:***Tenant storage* is a shared storage space that may be provided by either a local HPE Ezmeral Data Fabric installation within HPE Ezmeral Runtime Enterprise or a remote storage service. Every tenant is assigned a sandbox area within this space that is accessible by a special, non-editable **TenantStorage** DataTap. All virtual nodes within the tenant can access this DataTap and use it for persisting data that is not tied to the life cycle of a given cluster. Tenant storage differs from other DataTap-accessible storage as follows:
  - A tenant may not access tenant storage outside of its sandbox.
  - The Platform Administrator can choose to impose a space quota on the sandbox.
- **User:** A *user* is the set of information associated with each person accessing the HPE Ezmeral Runtime Enterprise, including the authentication and site roles.
- **Worker node:** A *Worker node* is a container that is managed by a Master node in a cluster. For example, the Spark Worker is the worker node in a Spark virtual cluster. For Kubernetes, this is another term for Worker host. See *node*, above.

HPE Ezmeral Data Fabric on Kubernetes

The following terms are used when discussing HPE Ezmeral Data Fabric in a Kubernetes environment on HPE Ezmeral Runtime Enterprise. This list is intended to basic information to a user who is unfamiliar with HPE Ezmeral Data Fabric storage.

- **HPE Ezmeral Data Fabric:** A general purpose data store and file system that scales to support data-driven analytics, ML, and AI applications. HPE Ezmeral Data Fabric provides file store and NoSQL database (HBase API for binary and JSON) to move data in and out of the cloud, and provides event streams for streaming applications.
- **HPE Ezmeral Data Fabric on Bare Metal:** The name of the implementation of HPE Ezmeral Data Fabric on physical or virtual machines.
- **HPE Ezmeral Data Fabric on Kubernetes:** The name of the implementation of HPE Ezmeral Data Fabric in a Kubernetes cluster running in HPE Ezmeral Runtime Enterprise.
- **Data Fabric:** This is the short form of the term HPE Ezmeral Data Fabric. The term is often used when the type of implementation is not relevant to the concept or task.
- **Embedded Data Fabric:** This is a legacy option, and not supported on HPE Ezmeral Runtime Enterprise 5.5.0 or later releases.
- **Data Fabric cluster:** This is a Kubernetes cluster that is used for HPE Ezmeral Data Fabric storage. A Data Fabric cluster is a Custom Resource in Kubernetes that is supported by operators in HPE Ezmeral Runtime Enterprise.

last-updated: May 06, 2024

- **Node:** A *node* is a Kubernetes host that has been added to an HPE Ezmeral Runtime Enterprise cluster.
- **Data Fabric CR:** This typically refers to the Custom Resource specification for a Data Fabric cluster that is supported by an HPE Ezmeral Runtime Enterprise `dataplatform` operator. It specifies each type of pod that the cluster would comprise. The per-pod specification may include CPU, memory, disk, and port requirements. Together with node labels and annotations, the Data Fabric CR influences the placement and scheduling of cluster pods by Kubernetes. HPE Ezmeral Runtime Enterprise creates and applies the Data Fabric CR when creating the first Data Fabric cluster. The Data Fabric CR may be subsequently patched/modified when expanding the cluster, or by a user with suitable privileges.
- **Core Pods:** These are the pods that are specified in the `/spec/core` path of a Data Fabric CR. Some examples of core pods in a Data Fabric cluster incude CLDB, Zookeeper, MFS, and admincli pods.
- **Services:** These are generally the pods specified in the /spec/coreservices and /spec/monitoring paths of a Data Fabric CR. Some examples of service pods in a Data Fabric cluster include MCS (HPE Ezmeral Data Fabric Control System), Kibana, and Grafana. Any non-CLDB, non-ZK, and non-MFS pod may also be referred to as a service pod.

Key Features and Benefits

The key features and benefits of HPE Ezmeral Runtime Enterprise include:

- **Integrated platform for Big Data analytics and machine/deep learning:**HPE Ezmeral Runtime Enterprise is an infrastructure platform purpose-built for Big Data and/or AI applications—including data science, analytics, machine learning (ML), and deep learning (DL)—using enterprise-grade security, networking, and support for a variety of local and remote storage options.
- **Runs on-premises and/or on public cloud virtual machines (VMs):** HPE Ezmeral Runtime Enterprise can be deployed on-premises, in the public cloud, or in a hybrid environment that includes both public cloud and on-premises resources.
- **Create virtual clusters:** HPE Ezmeral Runtime Enterprise uses containers to replicate the functionality of physical clusters while adding flexibility and scalability at reduced cost. You may create, modify, re-prioritize, and remove containerized clusters (referred to as *virtual clusters* throughout these articles) on demand in response to ever-changing needs within individual business units/departments. HPE Ezmeral Runtime Enterprise reduces time-to-value from months to hours.
- **Multi-tenancy and enterprise-grade security model:** HPE Ezmeral Runtime Enterprise integrates with enterprise LDAP and Active Directory authentication systems. Administrators can create groupings of users and resources that restrict access to jobs, data, or clusters based on department membership and/or roles. The result is an integrated, secure, multi-tenant infrastructure.
- **Self-service portal:** HPE Ezmeral Runtime Enterprise includes a self-service web portal that allows users to create and manage clusters, create and manage nodes, run jobs, and view monitoring statistics. User visibility into resources and ability to take action on the platform vary based on each user's role and tenant membership, in accordance with existing enterprise security policies. For example, department administrators can use the portal to provision their own nodes/clusters without impacting nodes/clusters that are assigned to different departments and without having to manage the physical infrastructure.
- **RESTful API:** HPE Ezmeral Runtime Enterprise supports a RESTful API that surfaces programmable access to the same capabilities available via the self-service portal.

last-updated: May 06, 2024

- **Superior performance:** HPE Ezmeral Runtime Enterprise provides storage I/O optimizations to deliver data to applications without the penalties commonly associated with virtualization or containerization. The CPU cores and RAM in each host are pooled and then partitioned into virtual resource groups based on tenant requirements.
- **Works with existing infrastructure:** HPE Ezmeral Runtime Enterprise allows your enterprise to repurpose its existing infrastructure investments. HPE Ezmeral Runtime Enterprise can run on your physical and virtualized infrastructure, including CPUs and GPUs, as well as on all three major public clouds (Amazon Web Services, Google Cloud Platform, and Microsoft Azure). Existing storage protocols are also supported (HDFS, HDFS with Kerberos, and NFS).
- **Reduced IT overhead:** HPE Ezmeral Runtime Enterprise streamlines operations and reduces IT costs by automating provisioning, unifying management, and supporting push-button upgrades.
- **Increases utilization while lowering costs:** HPE Ezmeral Runtime Enterprise delivers hardware and operational cost savings while simultaneously eliminating the complexity of managing multiple physical clusters. HPE Ezmeral Runtime Enterprise allows clusters to share a common pool of hardware resources (e.g. CPU and storage).
- **High Availability:** HPE Ezmeral Runtime Enterprise supports three levels of High Availability (at the platform, virtual cluster, and/or Gateway node level) to provide redundancy and protection.
- **Compute and storage separation:** HPE Ezmeral Runtime Enterprise supports decoupling of analytical processing from data storage, giving you the ability to independently scale compute and storage capacity instantly on an as-needed basis. This permits more effective utilization of infrastructure resource and reduces overall costs.
- **In-place access to both on-premises enterprise storage and cloud storage:** HPE Ezmeral Runtime Enterprise enables you to access and run jobs directly against both existing enterprise-class storage systems and cloud storage systems. The separation of compute and storage provided by HPE Ezmeral Runtime Enterprise means that you don't need to move or duplicate data before running analytics.

Application Support

HPE Ezmeral Runtime Enterprise provides powerful support for Artificial Intelligence (AI) and Machine Learning (ML) applications (see Artificial Intelligence and ML/DL Workloads on page 82 for additional information). It also includes pre-configured, ready-to-run versions of major Hadoop distributions, such as Cloudera (CDH), Hortonworks (HDP), and MapR (CDP). It also includes recent versions of Spark standalone as well as Kafka and Cassandra. I. Other distributions, services, commercial applications, and custom applications can be easily added to an HPE Ezmeral Runtime Enterprise deployment, as described in App Store on page 85. Some of the Big Data, AI, and ML application services that are supported out-of-the-box include:

- **CAFFE2:** CAFFE (Convolutional Architecture for Fast Feature Embedding) is a deep learning framewors that is merged into PyTorch.
- **Cloudera Manager (for CDH):** Cloudera Manager provides a real-time view of CDH clusters, including a real-time view of the nodes and services running, in a single console. It also includes a full range of reporting and diagnostic tools to help optimize performance and utilization.
  - **Flume:** Flume-NG is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of server log data. It is robust and fault tolerant with many failover and recovery mechanisms. It uses a simple extensible data model that allows one to build online analytic applications.

  - **HBase:** HBase is a distributed, column-oriented data store that provides random, real-time read/write access to very large data tables (billions of rows and millions of columns) on a Hadoop cluster. It is modeled after Google's BigTable system.
  - **MapReduce:** MapReduce assigns segments of an overall job to each Worker, and then reduces the results from each back into a single unified set.
  - **Sqoop:** Sqoop is a tool designed for efficiently transferring bulk data between Hadoop and structured datastores, such as relational databases. It facilitates importing data from a relational database, such as MySQL or Oracle DB, into a distributed filesystem like HDFS, transforming the data with Hadoop MapReduce, and then exporting the result back into an RDBMS.
- **GraphX (for Spark):** GraphX works seamlessly with graphs and collections by combining Extract/Transform/Load (ETL), exploratory analysis, and iterative graph computation within a single system. The Pregel API allows you to write custom iterative graph algorithms.
- **Hive:** Hive facilitates querying and managing large amounts of data stored on distributed storage. This application provides a means for applying structure to this data and then running queries using the HiveQL language. HiveQL is similar to SQL.
- **JupyterHub:** JupyterHub is a multi-user server that provides a dedicated single-user Jupyter Notebook server for each user in a group.
- **Kafka:** Kafka allows a single cluster to act as a centralized data repository that can be expanded with zero down time. It partitions and spreads data streams across a cluster of machines to deliver data streams beyond the capability of any single machine.
- **Kubeflow:** A platform for developing and deploying an ML system. This is the ML toolkit for Kubernetes.
- **MLlib:** MLlib is Spark's scalable machine learning library that contains common learning algorithms, utilities, and underlying optimization primitives.
- **Oozie:** Oozie is a workflow scheduler system for managing Hadoop jobs that specializes in running workflow jobs with actions that run Hadoop MapReduce and Pig jobs.
- **Pig:** Pig is a language developed by Yahoo that allows for data flow and transformation operations on a Hadoop cluster.
- **PyTorch:** Open-source ML library based on the Torch library and used for applications such as computer vision and natural language processing.
- **Spark SQL:** Spark SQL is a Spark module designed for processing structured data. It includes the DataFrames programming abstraction and can also act as a distributed SQL query engine. This module can also read data from an existing Hive installation.
- **SparkR:** SparkR is an R package which provides a lightweight front end for using Spark from R.
- **Spark Streaming:** Spark Streaming is an extension of the core Spark API that enables fast, scalable, and fault-tolerant processing of live data streams.
- **Tensorflow:** Open-source framework to run ML, deep learning, and other statistical and predictive analytics workloads.

Artificial Intelligence and ML/DL Workloads

Enterprises are increasingly turning to AI to solve complex problems, conduct research, and maintain or boost their competitive advantages in the marketplace. AI and machine learning (ML)/deep learning (DL)

technologies have moved into the mainstream with a broad range of data-driven enterprise applications: credit card fraud detection, stock market prediction for financial trading, credit risk modeling for insurance, genomics and precision medicine, disease detection and diagnosis, natural language processing (NLP) for customer service, autonomous driving and connected car IoT use cases, and more.

A typical distributed ML/DL workflow may look something like this:

1. The model is conceptualized.
2. The model is built in one or more sandbox/custom environment(s) that require access to data and model storage.
3. Subsequent versions are created that may add libraries and/or features and that require rerunning the model.
4. The model is saved and deployed, and any API endpoints are published.
5. Measurements to determine model efficacy occur both in real time and in batch feedback loops. This feedback is used to continue conceptualizing the model.

Needs

Enterprises wanting to deploy distributed ML/DL infrastructures typically have some or all of the following needs:

- Role-based access control to some or all of the following:
  - ML/DL tools, such as TensorFlow, H2O, MXNet, BigDL for Spark, Caffe, and SparkMLib..
  - Common “big” and “small” data frameworks, such as Kafka, HDFS, HBase, Spark, model storage, and workflow management.
  - Data science notebooks, such as Jupyter, RStudio, and Zeppelin.
  - Various related analytics, business intelligence (BI), and ETL tools.
- Choice of modeling techniques.
- Ability to build, share, and iterate.
- Reproducibility.
- Easy scaling for testing on actual data sets.
- Support for varying roles and actions.

Challenges

Some of the key challenges enterprises face when looking to build, deploy, and operationalize their ML/DL pipelines to meet the needs described above include:

- Traditional analytics tools were built to process structured data in databases. AI use cases that require ML/DL tools require a large and continuous flow of data that is typically unstructured.
- Data scientists and developers may have built and designed their initial ML/DL algorithms to operate in a single-node environment (e.g. on a laptop, virtual machine, or cloud instance) but need to parallelize the execution in a multi-node distributed environment.
- Enterprises cannot meet their AI use case requirements using the data processing capabilities and algorithms of a single ML/DL tool. They need to use data preparation techniques and models from multiple open source and/or commercial tools.

last-updated: May 06, 2024

- Data science teams are increasingly working in more collaborative environments where the workflow for building distributed ML/DL pipelines spans multiple different domain experts.
- Many ML / DL deployments use hardware acceleration such as GPUs to improve processing capabilities. These are expensive resources, and this technology can add to the complexity of the overall stack.
- ML/DL technologies and frameworks are different from existing enterprise systems and traditional data processing frameworks.
- ML/DL stacks are complex because they require both multiple software and infrastructure components and version compatibility and integration across those components.
- Assembling all of the required systems and software is time consuming, and most organizations lack the skills to deploy and wire together all of these components.

The HPE Ezmeral Runtime Enterprise Solution for ML/DL

HPE Ezmeral Runtime Enterprise goes beyond Application Support on page 81 by leveraging the inherent infrastructure portability and flexibility of containers to support distributed AI for both ML and DL use cases. The separation of compute and storage for Big Data and ML/DL workloads is one of the key concepts behind this flexibility, because organizations can deploy multiple containerized compute clusters for different workflows (e.g. Spark, Kafka, or TensorFlow) while sharing access to a common data lake. This also enables hybrid and multi-cloud HPE Ezmeral Runtime Enterprise deployments, with the ability to mix and match on- and/or off-premises compute and storage resources to suit each workload. Further, compute resources can be quickly and easily scaled and optimized independent of data storage, thereby increasing flexibility and improving resource utilization while eliminating data duplication and reducing cost.

Some of the key ML/DL features and benefits that HPE Ezmeral Runtime Enterprise provides include:

- **Container-based automation:** HPE Ezmeral Runtime Enterprise creates virtual clusters that each contain one or more container(s). Containers are now widely recognized as a fundamental building block to simplify and automate deployments of complex application environments, with portability across on-premises infrastructure and public cloud services.
- **Deployment of ML/DL workloads:** HPE Ezmeral Runtime Enterprise can be used to deploy distributed ML/DL environments such as TensorFlow, Caffe2, H2O, BigDL, and SparkMLlib. This allows organizations embarking on AI initiatives to quickly spin up multi-node ML/DL sandbox environments for their data science teams. If available, they can also easily and securely tap into an existing data lake to build and deploy their ML/DL pipelines.
- **Rapid and reproducible provisioning:** Users can spin up new, fully-provisioned distributed ML/DL applications in multi-node containerized environments on any infrastructure, whether on-premises or in the cloud, using either CPUs and/or GPUs. These fully-configured environments can be created in minutes via either RESTful APIs or a few mouse clicks in the HPE Ezmeral Runtime Enterprise web interface. IT teams can ensure enterprise-grade security, data protection, and performance with elasticity, flexibility, and scalability in a multi-tenant architecture. The template feature allows organizations to preserve specific cluster configurations for reuse at any time with just a few mouse clicks. EPIC publishes service endpoint lists for each virtual node/cluster.
- **Decoupling of compute and storage resources:** As described above, the separation of compute from storage allows organizations to reduce costs by scaling these infrastructure resources independently while leveraging their existing storage investments in file, block, and object storage to extend beyond their petabyte-scale HDFS clusters. HPE Ezmeral Runtime Enterprise allows secure integrations with distributed file systems including HDFS, NFS, and S3 for storing data and ML / DL models, including pass-through security from the compute clusters.

App Store

Describes the App Store and its relationship to HPE Ezmeral Runtime Enterprise. (Not available in HPE Ezmeral Runtime Enterprise Essentials.)

HPE Ezmeral Runtime Enterprise includes an App Store with one-click deployment for common Big Data and AI tools.

The App Store contains Docker container images of each available application, allowing fully automated self-service deployment. Each image in the App Store provides a particular version, is pre-configured, and ready-to-run on HPE Ezmeral Runtime Enterprise. HPE Ezmeral Runtime Enterprise also supports a “bring your own app” model that allows users to quickly add images to the App Store.

The **App Store** contains three classes of images:

Hadoop, Spark, Kafka, and other Big Data tools provided out-of-the-box by HPE	These images contain open-source software that is unmodified and supported by these vendors.
ML/DL, analytics, and data science tools supported out-of-the-box by HPE	The App Store includes several pre-configured open-source tools as examples. Other tools have also been tested for compatibility with HPE Ezmeral Runtime Enterprise and can be made available to customers, including both open-source and commercial applications.
Custom tools and applications added specifically by individual customers	HPE Ezmeral Runtime Enterprise provides an *Application Workbench* that allows customers to create and add their own images to the App Store. Users can then deploy these images and use them in a similar way as any of the out-of-the-box images described above.

NOTE: App Store images are independent from the HPE Ezmeral Runtime Enterprise itself. Any tool or application can be added or removed from the deployment to suit your specific needs.

The Platform Administrator may install or uninstall images. Installed images are available for use by Tenant Members when creating jobs and clusters.

Hewlett Packard Enterprise and/or application vendors may provide new images or new versions of existing images.

For legacy EPIC applications, if the HPE Ezmeral Runtime Enterprise Controller host can access the internet and a new version becomes available for an image that is currently installed, the image will be marked in the **App Store** screen with an **Upgrade Available** banner, and its tile will provide a button for upgrading to the new version. Other new images or new versions of currently uninstalled images will display a **New** banner.

If you are deploying applications in Kubernetes, then you can use the KubeDirector feature that comes pre-installed in HPE Ezmeral Runtime Enterprise. This feature lists the applications that you can launch into your cluster by accessing a Kubernetes tenant and then clicking the **Applications** tab.

OS Support

HPE Ezmeral Runtime Enterprise supports the following operating systems:

HPE Ezmeral Runtime Enterprise Version	CentOS Support	RHEL Support	SUSE Support
5.6.4 and higher	7.8, 7.9	7.8, 7.9, 8.x* Minimum kernel version: 3.10.0-1062**	15 SP2, 15 SP3, 15 SP4
5.6.2	7.8, 7.9	7.8, 7.9, 8.x* Minimum kernel version: 3.10.0-1062**	15 SP2, 15 SP3, 15 SP4
5.6.1	7.8, 7.9	7.8, 7.9, 8.x* Minimum kernel version: 3.10.0-1062**	15 SP2, 15 SP3, 15 SP4
5.6.0	7.8, 7.9	7.8, 7.9, 8.1-8.5* Minimum kernel version: 3.10.0-1062**	15 SP2, 15 SP3
5.5.0-5.5.1	7.8, 7.9	7.8, 7.9, 8.1-8.5* Minimum kernel version: 3.10.0-1062**	15 SP2, 15 SP3
5.4.1	7.8, 7.9	7.8, 7.9, 8.1-8.5* Minimum kernel version: 3.10.0-1062	Not Supported
5.4.0	7.8, 7.9	7.8, 7.9, 8.1-8.4* Minimum kernel version: 3.10.0-1062	Not Supported
5.3.5-5.3.6	7.8, 7.9 Minimum kernel version: 3.10.0-1062	7.8, 7.9, 8.1-8.2* Minimum kernel version: 3.10.0-1062	Contact HPE Support for information and assistance.
5.3.1- 5.3.4	7.8, 7.9 Minimum kernel version: 3.10.0-1062	7.8, 7.9, 8.1-8.2* Minimum kernel version: 3.10.0-1062	15 SP2

* RHEL 8.x is supported as follows:

- On HPE Ezmeral Runtime Enterprise 5.6.0 and later releases:
  - RHEL 8.x is supported on Kubernetes cluster hosts, with all hosts in the Kubernetes cluster on RHEL 8.x.
  - For fresh installations of HPE Ezmeral Runtime Enterprise, RHEL 8.x is also supported on HPE Ezmeral Runtime Enterprise control plane (Controller, Shadow, Arbiter, and Gateway) hosts.
  - For existing deployments of HPE Ezmeral Runtime Enterprise, upgrading control plane (Controller, Shadow, Arbiter, and Gateway) hosts to RHEL 8.x is not supported.

  See also Operating System Requirements on page 820.

- On HPE Ezmeral Runtime Enterprise 5.5.x and earlier releases, RHEL 8.x is supported on Kubernetes hosts only, with all hosts in the Kubernetes cluster on RHEL 8.x. See Operating System Requirements on page 820.

** The minimum kernel version requirement is relevant to deployments that have been upgraded from versions of HPE Ezmeral Runtime Enterprise prior to 5.5.0 only.

For information about operating system configuration requirements for all OS versions, see Operating System Requirements on page 820 and Configuration Requirements on page 826.

Host operating system upgrades to minor operating system releases are supported. For example, if HPE Ezmeral Runtime Enterprise is installed on RHEL 7.8, you can upgrade the host operating system to RHEL 7.9. For more information about upgrading host operating systems, see System Maintenance on page 802.

To minimize the need for troubleshooting, Hewlett Packard Enterprise recommends newer kernel versions.

SELinux is supported on HPE Ezmeral Runtime Enterprise 5.2 and later in Enforcing, Permissive, and Disabled mode as follows:

- To enable Enforcing mode on nodes that are part of HPE Ezmeral Data Fabric on Kubernetes, contact Hewlett Packard Enterprise Support.
- The mode cannot be changed after installing HPE Ezmeral Runtime Enterprise.
- For SLES 15 SP2 and SLES 15 SP3, supported with HPE-installed SELinux policies only.

Product Licensing

Provides information related to product licensing.

What's Included

Provides links to product QuickSpecs.

For information about the products and features included in HPE Ezmeral Runtime Enterprise and related products, refer to the product QuickSpecs:

HPE Ezmeral Runtime Enterprise	HPE Ezmeral Runtime Enterprise QuickSpecs
HPE Ezmeral Data Fabric	HPE Ezmeral Data Fabric QuickSpecs
HPE Ezmeral Machine Learning (ML) Ops	HPE Ezmeral Machine Learning (ML) Ops QuickSpecs

Related concepts

Licensing on page 734

HEWLETT PACKARD ENTERPRISE SOFTWARE END USER SUBSCRIPTION AGREEMENT

BY CLICKING A BOX INDICATING LICENSEE'S ACCEPTANCE, BY EXECUTING AN ORDER THAT REFERENCES THIS AGREEMENT, OR BY USING THE SOFTWARE, LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. THE PERSON ENTERING INTO THIS AGREEMENT ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY REPRESENTS THAT THEY HAVE THE AUTHORITY TO BIND SUCH ENTITY AND ITS AFFILIATES TO THESE TERMS AND CONDITIONS, IN WHICH CASE THE TERM “LICENSEE” SHALL REFER TO SUCH ENTITY AND ITS AFFILIATES.

Scope. This End User Subscription Agreement (“Agreement”) sets forth the exclusive terms and conditions under which Hewlett Packard Enterprise (“Licensor”) grants a limited use license to the end user (“Licensee”) of the Licensor's software (the “Software”) for a specific duration (“Subscription Period”). This Agreement includes supporting material accompanying the software or referenced by Licensor, which may be software license information, additional license authorizations, software specifications, published warranties, supplier terms, open source software licenses, product lists, hardware or software specifications, standard or negotiated service descriptions, data sheets and their supplements, statements of work (SOWs), published warranties, data protection and security addendum, service level agreements, and similar content ("Supporting Material"). Additional license authorizations are at: http://www.hpe.com/

software/SWLicensing. “Order” means the accepted order including any Supporting Material which the parties identify as incorporated either by attachment or reference. Any conflicting term or condition, including those that may be embedded in any purchase order, order acknowledgment, invoice or other forms used by the parties, shall be of no force or effect unless agreed to in a writing by the party against which any such term or condition is asserted.

1. Grant of License. Licensor hereby grants Licensee a nonexclusive, nontransferable license without the right to sublicense, to install and internally use the Software solely in conjunction with Licensee's information technology network and for no other purpose. Licensee must limit access to and use of the Software by only its employees with the need to use the Software for the foregoing purpose (“Authorized Users”).

2. Subscription Types and Term. Unless otherwise specified in the quotation, the Subscription Period will begin on the date of Order fulfillment to Licensor and will end 1-year, 2-years, 3-years, 4-years, or 5-years from the Order fulfillment date depending on the subscription part number, type, description, and associated subscription term. ALL SUBSCRIPTIONS ARE NON-CANCELLABLE.

   a. Universal Subscription. A Universal Subscription allows the Software to be deployed on any server or virtual machine from any vendor on premise and in the public cloud. The Universal Subscription does not include the right to certain modules which may be sold separately.

   b. Select Subscription. A Select Subscription can only be purchased in conjunction with the specific Select for Licensor hardware offerings. Select Subscriptions can only be deployed on those specific Select for Licensor hardware offerings, either natively or in virtual machines, and in a hybrid cloud cluster that includes those specific Select for Licensor hardware offerings. In addition, no more than 30% of the quantity of Select Subscriptions deployed on those specific Select for Licensor hardware offerings can be deployed in a public cloud as part of Licensee's hybrid cloud cluster. If Licensee needs a hybrid cloud cluster with more than 30% of the quantity of Select Subscriptions deployed on those specific Select for Licensor hardware offerings deployed in a public cloud, Licensee will need to purchase Universal Subscriptions for those licenses greater than this 30% cap. The Select Subscription does not include the right to certain modules which may be sold separately.

3. The HPE Ezmeral Container Platform SKUs and the HPE Ezmeral Machine Learning Ops SKUs shall provide entitlement to use the HPE Ezmeral Data Fabric product (formerly known as MapR Data Platform) up to the number of licensed Cores (as defined below) and terabytes of Storage Capacity (as defined below). As such, the number of Cores and terabytes of Storage Capacity used for the HPE Ezmeral Data Fabric deployment and/or an HPE Ezmeral Container Platform deployment cannot exceed the total licensed Cores.

4. Each license allows the customer to deploy the HPE Ezmeral Container Platform on one Core and 2 terabytes of Storage Capacity. The customer must purchase more licenses if they exceed the allowable amount of Cores or Storage Capacity. As used in this Agreement, Core means a part of a CPU that executes a single stream of compiled instruction code. Each physical processor contains smaller processing units called physical CPU cores. Some processors have two cores, some four, some eight, and so on. Core capacity represents the total number of cores available within a given system. The number of cores is counted as the number of logical cores presented to the product guest OS. For licensing purposes, the number of cores on a given Ezmeral Container Platform host is the number of unique cores available to the kernel in the OS on which the Ezmeral Container Platform software is directly installed, regardless of the number of threads in each core. It equals the product of Core(s) per socket and Socket(s), as shown in the output of the lscpu command. This applies whether the OS is running directly on a bare metal host, or a virtual machine. Hyperthreading in the OS is ignored (i.e, if hyperthreading is enabled in the Ezmeral Container Platform's OS, causing 8 vCPUs to exist on a 4-core host, that host will only require 4 cores to be licensed). Conversely, if Ezmeral Container Platform software is installed on a virtual machine on an overprovisioned hypervisor, and the guest OS reports more cores than physically exist on the underlying bare metal host, each of those cores must still be licensed. Storage Capacity means the total storage capacity (HDD & SSD) allocated to and managed by HPE Products, measured in Terabytes (TB) of raw capacity. Includes space for data, data replication, erasure coding, snapshots, metadata, logs and other data that is stored in HPE Data Fabric.

5. Ownership of Software. All right, title, and interest in and to the Software and all modifications and derivatives thereof, including all patent, copyright, trade secret, and other intellectual property rights therein resides and will reside in Licensor and its licensors, as applicable. The Software is licensed to Licensee by Licensor. Licensee acknowledges and agrees that the Software is copyrighted and contains materials that are valuable trade secrets of the Licensor and are protected by copyright, trade secret, and other laws and international treaty provisions relating to proprietary rights. Licensee may not remove, deface, or obscure any of Licensor's or its suppliers' proprietary rights notices on or in the Software or on output generated by the Software. Licensee may not, nor may Licensee permit, any third party to: (a) decompile, reverse engineer, disassemble, decrypt, or otherwise attempt to derive the source code, algorithms, or underlying ideas, design, or architecture of the Software; (b) modify, translate, or create derivative works of the Software; (c) use the Software to provide services to third parties, (d) may not download and use patches, enhancements, bug fixes, or similar updates unless you have a license to the underlying Software. However, such license does not automatically give you a right to receive such updates and Licensor reserves the right to make such updates only available to Licensees with support contracts, (e) may not copy Software or make it available on a public or external distributed network or (f) rent, lease, loan, distribute, transmit, assign, or transfer the Software to any third party, or provide access to or use of the Software by any third party, including any agent of Licensee's other than Authorized Users. Without Licensor's prior written consent, Licensee shall not disclose any information relating to features, performance, or operation of the Software (including any benchmarking or other testing results) to any third party, nor use any of such information other than for purposes permitted under the section titled Section 1 above. All rights not specifically granted in this Agreement are reserved by Licensor. Licensee acknowledges and agrees that any unauthorized use, transfer, sublicensing, or disclosure of the Software, or other violation of this License, would cause irreparable injury to Licensor, and under such circumstances, Licensor shall be entitled to equitable relief, without posting bond or other security, including but not limited to, preliminary and permanent injunctive relief. Licensor may upon written notice terminate this Agreement if the forgoing restrictions are violated or the fees are not paid when due.

last-updated: May 06, 2024

6. Feedback. All questions, comments, or feedback provided by Licensee to Licensor regarding the Software and any other products, services, or materials provided by Licensor (collectively, “Feedback”) will be deemed the property of the Licensor. Licensor will have no obligation to Licensee or any third party with respect to such Feedback, and be free to use such Feedback in any form or manner and for any purpose and without payment of any consideration to Licensee or any third party. All rights, title, and interest in and to the Software, the Feedback, accompanying materials, and all proprietary information contained therein, are owned by Licensor and are protected by copyright, trademark, patent and trade secret law, and international treaties. Licensee will transfer and assign, and hereby does irrevocably transfer and assign, to Licensor all right, title, and interest, including all intellectual property rights that Licensee may have or acquire in the Feedback, and Licensee will provide reasonable assistance to Licensor to effect such assignment.

7. Licensee-provided Data. Licensee is solely responsible for the content created or placed into a Licensor system during Licensee's access or use of Software (“Licensee-provided Data”). As between Licensor and Licensee, Licensee is and will remain the sole and exclusive owner of all right, title, and interest in and to all Licensee-provided Data. Licensee hereby provides to Licensor all necessary rights to Licensee-provided Data to enable Licensor to provide the Software and any related services. Licensor will use Licensee-provided Data only as necessary to provide the Software, technical support, or as otherwise required by law.

8. Personal Data. Where legitimate business purposes require Licensor to collect and process business contact information relating to Licensee's employees or other individuals representing Licensee, Licensor, as a data controller, will process such personal data using appropriate technical and organizational measures and in compliance with its Privacy Statement (https://www.hpe.com/us/en/legal/privacy.html) and applicable laws. Where Licensor discloses personal data relating to its employees or other individuals representing Licensor to Licensee or where such persons provide their personal data directly to Licensee, Licensee will process such personal data using appropriate technical and organizational measures in compliance with Licensee's privacy policies and applicable laws. Where Licensor agrees to process personal data on behalf of Licensee, Licensor, as a data processor, will process such data only as permitted under this Agreement, including Supporting Materials, and in compliance with applicable laws. In the event international data transfers trigger the requirements for an EU Model Contract, Licensee and its applicable affiliates (i) authorize Licensor to execute the EU Model Contract with Licensor's affiliates on Licensee's behalf or (ii) will execute EU Model Contracts directly with the Licensor and its applicable affiliates.

9. Consent to Use of Data. Licensee agrees that Licensor may collect and use technical data and related information, including but not limited to technical information about Licensee's computer system, application software, peripherals that is gathered periodically to facilitate the provision of software updates, product support and other services related to the Software. Licensor may use this information, as long as it is in a form that does not personally identify Licensee, to improve its products or to provide services or technologies.

10. Fees. The license granted hereunder is subject to Licensee's timely payment of fees. Licensee is responsible for payment of sales, use, VAR, import, and all other transaction taxes and fees except for taxes based on Licensor's or its reseller's net income. If Licensee is required by a governmental taxing authority to withhold an amount from any payment due hereunder and pay such amount to the governmental authority, the prices will be grossed-up so that the net payment equals the original price.

11. Intellectual Property Rights Infringement. Licensor will defend and/or settle any claims against you that allege that Licensor-branded software as supplied under this Agreement infringes the intellectual property rights of a third party. Licensor will rely on Licensee's prompt notification of the claim and cooperation with Licensor's defense. Licensor may modify the software so as to be non-infringing and materially equivalent, or Licensor may procure a license. If these options are not available, Licensor will refund to Licensee the remaining prepaid amount and Licensee will discontinue further use of the license. Licensor is not responsible for claims resulting from any unauthorized use of the Software.

last-updated: May 06, 2024

12. Support and Maintenance. Subject to the terms and conditions of this Agreement, Licensor shall make available to Licensee any updates to the Software developed by or on behalf of Licensor during the term of this Agreement that are available for distribution (as determined by Licensor). In the event Licensor provides any additional services to the Licensee which are not expressly included in this Agreement, Licensee shall pay Licensor for such additional services at agreed upon market rates, and enter into appropriate additional Agreements for such services.

13. Disclaimer; Limitation of Liability. LICENSEE'S USE OF THE SOFTWARE IS AT ITS SOLE RISK. LICENSOR WARRANTS THAT ITS BRANDED SOFTWARE PRODUCTS WILL BE FREE OF MALWARE AT THE TIME OF DELIVERY. LICENSOR(S) EXPRESSLY DISCLAIM ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY OR SATISFACTORY QUALITY AND FITNESS FOR A PARTICULAR PURPOSE, NOR DOES THE LICENSOR PROVIDE ANY WARRANTY WITH RESPECT TO VIOLATION OF THE RIGHTS OF THIRD PARTIES. IN ADDITION, LICENSOR DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE. BOTH PARTIES ACKNOWLEDGE THAT THEY HAVE NOT ENTERED INTO THESE TERMS IN RELIANCE UPON ANY WARRANTY OR REPRESENTATION. UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL LICENSOR BE LIABLE FOR ANY DIRECT, INCIDENTAL, SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS), OR FOR THE COST OF PROCURING OTHER SOFTWARE PRODUCTS OR SERVICES, ARISING OUT OF OR RELATING TO THIS AGREEMENT, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. LICENSOR'S LIABILITY TO LICENSEE UNDER THIS AGREEMENT IS LIMITED TO THE AMOUNT ACTUALLY PAID BY LICENSEE TO LICENSOR FOR THE RELEVANT SOFTWARE, EXCEPT FOR AMOUNTS IN SECTION 12 ("INTELLECTUAL PROPERTY RIGHTS INFRINGEMENT"). THIS PROVISION DOES NOT LIMIT EITHER PARTY'S LIABILITY FOR: UNAUTHORIZED USE OF INTELLECTUAL PROPERTY, DEATH OR BODILY INJURY CAUSED BY THEIR NEGLIGENCE; ACTS OF FRAUD; WILLFUL REPUDIATION OF THE AGREEMENT; OR ANY LIABILITY THAT MAY NOT BE EXCLUDED OR LIMITED BY APPLICABLE LAW.

14. Confidentiality. The Software, any related benchmark or performance tests, and information regarding Licensor's business, including technical, marketing, financial, employee, planning, and other confidential or proprietary information is considered Licensor's “Confidential Information”. Licensee shall protect the Confidential Information from unauthorized dissemination and use with the same degree of care that Licensee uses to protect its own like information and, in any event, will use no less than a reasonable degree of care in protecting such Confidential Information. Licensee will use the Confidential Information only for those purposes expressly authorized in this Agreement. Licensee will not disclose to third parties the Confidential Information without the prior written consent of Licensor.

15. Termination. This Agreement shall be effective on the date of receipt of the Licensee's Purchase Order and shall expire and/or terminate (i) if Licensee breaches its obligations under this Agreement or any applicable Supporting Materials, (ii) at the end of the applicable Subscription Period or (iii) to comply with applicable laws or regulations. Upon notice of such event, Licensee's license and rights under this Agreement will terminate. Immediately upon any termination, Licensee must promptly destroy the Software, including all copies and portions thereof, in its possession or under its control and certify such destruction in writing to Licensor. Any terms in this Agreement which by their nature extend beyond termination or expiration of this Agreement will remain in effect until fulfilled and will apply to both parties' respective successors and permitted assigns.

16. General.

last-updated: May 06, 2024

a. Assignment. Licensee may not assign this Agreement without prior written consent of Licensor, payment of transfer fees and compliance with Licensor's software license transfer policies. Authorized assignments will terminate Licensee's license to the Software and you must deliver software and documentation and copies thereof to the assignee. The assignee will agree in writing to this Agreement.

b. U.S. Government. If the Software is licensed to you for use in the performance of a U.S. Government prime contract or subcontract, you agree that, consistent with FAR 12.211 and 12.212, commercial computer software, computer software documentation and technical data for commercial items are licensed under Licensor's standard commercial license.

c. Global Trade Compliance. You agree to comply with the trade-related laws and regulations of the U.S. and other national governments. If you export, import, or otherwise transfer products provided under this Agreement, you will be responsible for obtaining any required export or import authorizations. You confirm that you are not located in a country that is subject to trade control sanctions (currently Cuba, Iran, N. Korea, N. Sudan, and Syria) and further agree that you will not retransfer the products to any such country. Licensor may suspend its performance under this Agreement to the extent required by laws applicable to either party.

d. Audit. Licensor may audit you for compliance with the software license terms. Upon reasonable notice, Licensor may conduct an audit during normal business hours (with the auditor's costs being at Licensor's expense). If an audit reveals underpayments then you will pay to Licensor such underpayments. If underpayments discovered exceed five (5) percent, you will reimburse Licensor for the auditor costs.

e. Open Source Components. To the extent the Supporting Material includes open source licenses, such licenses shall control over this Agreement with respect to the particular open source component. To the extent Supporting Material includes the GNU General Public License or the GNU Lesser General Public License: (a) the software includes a copy of the source code; or (b) if you downloaded the software from a website, a copy of the source code is available on the same website; or (c) if you send Licensor written notice, Licensor will send you a copy of the source code for a reasonable fee.

f. Notices. Written notices under this Agreement may be provided to Licensor via the method provided in the Supporting Material.

g. Governing Law. This Agreement will be governed by the laws of the country where Licensor accepts the order, excluding rules as to choice and conflict of law. You and Licensor agree that the United Nations Convention on Contracts for the International Sale of Goods will not apply.

h. Force Majeure. Neither party will be liable for performance delays nor for non-performance due to causes beyond its reasonable control, except for payment obligations.

i. Entire Agreement. This Agreement represents Licensor entire understanding with respect to its subject matter and supersedes any previous communication or agreements that may exist. Modifications to the Agreement will be made only through a written amendment signed by both parties. If Licensor does not exercise its rights under this Agreement, such delay is not a waiver of its rights.

17. Australian Consumers. If you acquired the software as a consumer within the meaning of the 'Australian Consumer Law' under the Australian Competition and Consumer Act 2010 (Cth) then despite any other provision of this Agreement, the terms at this URL apply: http://www.hpe.com/software/SWLicensing.

18. Russian Consumers. If you are based in the Russian Federation and the rights to use the software are provided to you under a separate license and/or sublicense agreement concluded between you and a duly authorized Licensor partner, then this Agreement shall not be applicable.

Definitions

- **Core Capacity:** HPE Ezmeral Runtime is licensed by the number of unique cores available to the kernel in the OS on which the HPE Ezmeral Runtime software is directly installed, regardless of the number of threads in each core.
- **Storage Capacity:** Total storage capacity (HDD & SSD) allocated to and managed by HPE Products, measured in Terabytes (TB) of raw capacity. Includes space for data, data replication, erasure coding, snapshots, metadata, logs and other data that is stored in HPE Data Fabric.

5.6 Reference

HPE Ezmeral Runtime Enterprise 5.6

HPE Ezmeral Runtime Enterprise

Introduces HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise provides you with an enterprise-grade platform to deploy Kubernetes at scale for a wide range of use cases on bare metal or virtualized infrastructure. It can be run on premises, in hybrid and multi-cloud environments, and at the edge. HPE Ezmeral Runtime Enterprise is also the industry's first container platform designed to run modern applications (both cloud-native and non-cloud-native) with persistent data, making it easier for enterprises to manage their apps with containerized application deployments.

HPE Ezmeral Runtime Enterprise is the full-featured compute, storage, and container-management foundation that supports workload-specific solutions such as HPE Ezmeral ML Ops and HPE Ezmeral Unified Analytics.

For Kubernetes deployments that do not require all of the features that HPE Ezmeral Runtime Enterprise has to offer, see the HPE Ezmeral Runtime Enterprise Essentials product.

Key Features

Edge to cloud	The industry's first and only 100%, open-source, Kubernetes, hybrid analytics platform spanning edge to cloud helps enterprises modernize their apps with containerized application deployments on bare metal or VMs spanning on-premises, multiple clouds, and at the edge; allows you to build once, run anywhere
Public cloud cluster import	A unified control plane makes it easy to import external Kubernetes clusters; includes support for importing clusters from cloud vendors such as Amazon EKS, Google GKE, and Azure AKS.
Multi-cluster, multi-tenant Kubernetes management	Fast, easy deployment, management, and monitoring of Kubernetes clusters both on-prem and off-prem from a single pane of glass.
Enterprise-grade security	Built-in security controls integrate with identity providers such as AD/LDAP; single sign-on; SAML integration; role-based access controls for secure access to the platform; Falco container runtime security for proactive threat detection and alerting.

last-updated: May 06, 2024

GitOps-based centralized policy management and drift management	Seamless and fleet management of clusters; ArgoCD leveraged to ensure clusters are consistent and immutable for continuous compliance.
Turnkey solution	Easily containerize cloud-native and non-cloud-native apps; KubeDirector—an open-source custom Kubernetes controller—allows you to deploy non-cloud-native apps without rearchitecting or refactoring
Accelerated analytics	GPU sharing by using *NVIDIA Multi-instance GPU fractionalization* improves collaboration and GPU utilization.
Frictionless data access	HPE Ezmeral Data Fabric, DataTap and FSMount let you connect to and manage data wherever it is located.
Built-in Service Mesh and observability	For intelligent traffic shaping, load balancing, canary rollouts, and A/B testing of application microservices; visualize tenant-granular workload traffic for rapid troubleshooting and analysis via natively integrated Istio Service Mesh.
One-click provisioning	An App Store of curated, prebuilt, ready-to-run solutions for a wide range of applications including AI/ML, DataOps, analytics, CI/CD, DevOps apps and services, with the ability to BYO application via KubeDirector and App Workbench.
Available via HPE GreenLake	Cloud services for HPE Ezmeral Runtime Enterprise are available through HPE GreenLake to deliver a preconfigured platform designed for multi-cluster, multi-tenant Kubernetes deployment.

License Information

Information about the features included with an HPE Ezmeral Runtime Enterprise license, with a comparison to other HPE Ezmeral Runtime Enterprise product licenses, is provided in the product QuickSpecs. See What's Included on page 87.

More Information

Solutions Briefs, Articles, White Papers, and Videos	• HPE Ezmeral Runtime Enterprise • See also the HPE Ezmeral Software Portfolio interactive demo experience.

HPE Ezmeral Runtime Enterprise Essentials

Describes the HPE Ezmeral Runtime Enterprise Essentials product.

HPE Ezmeral Runtime Enterprise Essentials is a simple form of HPE Ezmeral Runtime Enterprise that is best suited for standalone Kubernetes clusters and edge use cases. There are several enterprise grade features that are NOT enabled including managed gateway, stateful applications with KubeDirector, integrated Kubernetes Data Fabric, centralized policy management, Falco, and MLOps tenants. For access these features, customers must purchase HPE Ezmeral Runtime Enterprise or HPE Ezmeral ML Ops.

last-updated: May 06, 2024

Upgrading from HPE Ezmeral Runtime Enterprise Essentials to HPE Ezmeral Runtime Enterprise

You can upgrade HPE Ezmeral Runtime Enterprise Essentials to the more feature-rich HPE Ezmeral Runtime Enterprise simply by purchasing and applying the HPE Ezmeral Runtime Enterprise license. See Upgrading from HPE Ezmeral Runtime Enterprise Essentials on page 911.

NOTE:

After a deployment is upgraded from HPE Ezmeral Runtime Enterprise Essentials to HPE Ezmeral Runtime Enterprise, the environment cannot be downgraded back to HPE Ezmeral Runtime Enterprise Essentials.

Likewise, you cannot change a deployment of HPE Ezmeral Runtime Enterprise to HPE Ezmeral Runtime Enterprise Essentials as part of an upgrade from one version to another.

License Information

Information about the features included with an HPE Ezmeral Runtime Enterprise Essentials license, with a comparison to other HPE Ezmeral Runtime Enterprise product licenses, is provided in the product QuickSpecs. See What's Included on page 87.

HPE Ezmeral ML Ops

Describes the HPE Ezmeral ML Ops solution and how it relates to HPE Ezmeral Runtime Enterprise.

HPE Ezmeral ML Ops is an end-to-end data science machine learning (ML) solution with the flexibility to run on-premises, in multiple public clouds, or in a hybrid model and respond to dynamic business requirements in a variety of use cases.

The HPE Ezmeral ML Ops solution supports every stage of the machine learning (ML) lifecycle—from data preparation to model build, model training, model deployment, collaboration, and monitoring.

Key Features

Key features of HPE Ezmeral ML Ops include the following:

Model building	**Pre-packaged, self-service sandbox environments:** Sandbox environments with any preferred data science tools—such as TensorFlow, Apache Spark, Keras, PyTorch and more—to enable simultaneous experimentation with multiple ML or deep learning (DL) frameworks.
Model training	**Scalable training environments with secure access to Big Data:** On-demand access to scalable environments—single node or distributed multi-node clusters—for development and test or production workloads. Patented innovations provide highly performant training environments—with compute and storage separation—that can securely access shared enterprise data sources on-premises or in cloud-based storage.
Model deployment	**Flexible, scalable, endpoint deployment:** HPE Ezmeral ML Ops deploys the model's native runtime image, such as Python, R, H2O, into a secure, highly available, load-balanced, and containerized HTTP endpoint. An integrated model registry enables version tracking and seamless updates to models in production. Autoscaling from HPE Ezmeral ML Ops dynamically scales nodes for scoring engines.

Model monitoring	**End-to-end visibility across the ML lifecycle:** Complete visibility into runtime resource usage such as GPU, CPU, and memory utilization. Ability to track, measure, and report model performance along with third-party integrations track accuracy and interpretability.
Collaboration	**Enable CI/CD workflows with code, model, and project repositories:** Project repository and GitHub integration of HPE Ezmeral ML Ops provides source control, eases collaboration, and enables lineage tracking for improved auditability. The model registry stores multiple models—including multiple versions with metadata—for various runtime engines in the model registry.
Security and control	**Secure multitenancy with integration to enterprise authentication mechanisms:** HPE Ezmeral ML Ops software provides multitenancy and data isolation to ensure logical separation between each project, group, or department within the organization. HPE Ezmeral ML Ops integrates with enterprise security and authentication mechanisms such as LDAP, Active Directory, and Kerberos.
Hybrid deployment	**On-premises, public cloud, or hybrid:** HPE Ezmeral ML Ops runs on-premises on any infrastructure, on public clouds, or in a hybrid model, providing effective utilization of resources and lower operating costs.

License Information

HPE Ezmeral ML Ops must be licensed using an HPE Ezmeral ML Ops license, which entitles a maximum number cores that can be assigned to the quota of all ML Ops tenants.

In addition, the HPE Ezmeral ML Ops license includes the features and products that are part of the HPE Ezmeral Runtime Enterprise license, and many of the applications and features that are included in the HPE Ezmeral Runtime Analytics for Apache Spark license.

Information about the features included with an HPE Ezmeral ML Ops license, with a comparison to other HPE Ezmeral Runtime Enterprise product licenses, is provided in the product QuickSpecs. See What's Included on page 87.

More Information

Documentation for users and administrators	HPE Ezmeral ML Ops on page 148
Solutions Briefs, Articles, White Papers, and Videos	• hpe.com/info/mlops • See also the HPE Ezmeral Software Portfolio interactive demo experience.

HPE Ezmeral Runtime Analytics for Apache Spark

Describes HPE Ezmeral Runtime Analytics for Apache Spark, and how it relates to HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Analytics for Apache Spark is a unified analytics environment for big data processing, with built-in modules for streaming, SQL, machine learning, and graph processing.

HPE Ezmeral Runtime Analytics for Apache Spark augments HPE Ezmeral Runtime Enterprise, the full-featured compute, storage, and container-management foundation for the solution.

License Information

The license for HPE Ezmeral Runtime Analytics for Apache Spark is an add-on license to HPE Ezmeral Runtime Enterprise. HPE Ezmeral Runtime Analytics for Apache Spark is not available for use with HPE Ezmeral Runtime Enterprise Essentials.

For information about the features included with an HPE Ezmeral Runtime Analytics for Apache Spark license, with a comparison to other HPE Ezmeral Runtime Enterprise product licenses, is provided in the HPE Ezmeral Runtime Enterprise product QuickSpecs. See What's Included on page 87.

More Information

Documentation for users and administrators	Spark on Kubernetes on page 243
Solutions Briefs, Articles, White Papers, and Videos	• Search for Spark in the HPE Developer Community. • See also the HPE Ezmeral Software Portfolio interactive demo experience.

Software Versions

In most cases, documentation refers to a software version of HPE Ezmeral Runtime Enterprise by its major and minor release, such as release 5.6.

Unless specifically noted, references to the major and minor release also apply to maintenance releases. For example, references to HPE Ezmeral Runtime Enterprise 5.4 includes functions and features in subsequent maintenance releases, such as HPE Ezmeral Runtime Enterprise 5.4.1.

Kubernetes Bundles

Kubernetes Bundles are software packages that can contain software to support newer Kubernetes versions, updated add-ons, and software fixes. Kubernetes Bundles enable you to update your deployment without requiring you to upgrade to a newer version of HPE Ezmeral Runtime Enterprise.

Contents of Kubernetes Bundles

The contents of a Kubernetes bundle can vary. A Kubernetes bundle might include one or more of the following:

- Support for a newer version of Kubernetes
- New versions of application add-ons
- Software or Common Vulnerabilities and Exposures (CVEs) fixes for Kubernetes and add-on versions packaged in previous Kubernetes Bundles

To view the contents of a specific Kubernetes bundle, see Updates Tab on page 801.

Kubernetes Bundle Versions

The Kubernetes bundle version provides information about the scope of the updated software. Kubernetes bundle versions follow the industry-standard format:

```
<major>.<minor>.<maintenance>
```

In the `<major>.<minor>.<maintenance>` format, a number represents each type of version:

- The first number, the **`<major>`** version, changes when the bundle contains a newer Kubernetes version or an add-on version is not compatible with a previous version of HPE Ezmeral Runtime Enterprise. All Kubernetes bundle versions that have the same `<major>` version numbers are compatible with the same set of HPE Ezmeral Runtime Enterprise versions.

  When the **`<major>`** increments, the `<minor>` and `<maintenance>` versions are reset to zero, even though the Kubernetes bundle might also contain minor or maintenance updates.

- The second number, the **`<minor>`** version, changes when the bundle contains support for newer Kubernetes minor versions, newer add-on versions, or both.

  When the **`<minor>`** version increments, the `<maintenance>` version is reset to zero, even though the Kubernetes bundle might also contain maintenance updates.

- The third number, the **`<maintenance>`** version, changes when bundle contains one or more Kubernetes patches, add-on patches, or CVE fixes.

Kubernetes Bundle Compatibility

Kubernetes bundles are compatible with the versions of HPE Ezmeral Runtime Enterprise as listed in the Support Matrixes on page 54.

Support Life Cycle of Kubernetes Bundles

A Kubernetes bundle is supported for 12 months following its first major release. A Kubernetes bundle and its subsequent minor and maintenance updates, if any, share the same End-Of-Life (EOL) date.

For example, if the first release of a Kubernetes bundle is 1.0.3 and the bundle reaches End-Of-Life (EOL) in October of 2023, then Kubernetes bundle 1.1.0 and Kubernetes bundle 1.2.2 also reach EOL in October of 2023.

Quick Links

Welcome! This page links you to overview pages that then take you to the key information you need to get up and running with HPE Ezmeral Runtime Enterprise quickly and easily. If you are a new user, then we recommend that you view this information in the order presented below. As you can see in the left navigation pane, the documentation itself is arranged with the usage/administration information appearing above the planning/deployment instructions. We did this because you will typically install HPE Ezmeral Runtime Enterprise once and then use it continuously, and we want to make it easier to find the information you will need most often.

The links in this article break down into the following categories:

- General Familiarization
- Before Deployment
- Deploying HPE Ezmeral Runtime Enterprise
- Usage and Administration
- Support and Troubleshooting

General Familiarization

This information helps you understand how HPE Ezmeral Runtime Enterprise works.

- **Universal Concepts:** General background information. See Universal Concepts Overview.

Before Deployment

This information helps you plan and prepare to deploy HPE Ezmeral Runtime Enterprise.

- **Planning the Installation:** Information to help you plan your installation. See Planning Overview.
- **System Requirements:** Your infrastructure must meet all applicable system requirements before installation. See System Requirements Overview.

Deploying HPE Ezmeral Runtime Enterprise

When you are ready for installation, these instructions will guide you through the process.

- **Deploying the Platform:** Detailed, step-by-step deployment instructions. See Installation Overview. If your deployment will include GPU resources, then also see GPU Driver Installation.

Usage and Administration

This is where the rubber meets the road as you use and administer Kubernetes, Big Data, and/or AI/ML tenants and projects.

- **Accessing the Platform:** Logging in to HPE Ezmeral Runtime Enterprise once deployment has completed. See Access Overview.
- **Kubernetes:** Creating, using, and administering Kubernetes clusters and tenants. See Kubernetes Overview, Getting Started with General Kubernetes Functionality, and HPE Ezmeral ML Ops on page 148.
- **Platform Administration:** Administering HPE Ezmeral Runtime Enterprise. See Platform Administrator Overview on page 570.
- **Global Settings:** Accessing and using global settings. See Global Settings Overview.

Support and Troubleshooting

Helps you diagnose and resolve issues.

- **Support and Troubleshooting:** This section guides you through diagnosing and resolving problems that may occur. See Troubleshooting Overview.
- **Issues and Workarounds:** For information about the issues and workarounds for the current release, see the Release Notes on page 11.

What's New in Version 5.6.x

This topic summarizes the new features and important changes in HPE Ezmeral Runtime Enterprise 5.6.x compared to HPE Ezmeral Runtime Enterprise 5.5.x.

What's New in Version 5.6.x

The following is a summary of the new features in HPE Ezmeral Runtime Enterprise 5.6.X. The items in this list are relative to the previous general availability 5.6.0 release of HPE Ezmeral Runtime Enterprise.

Support for Kubernetes version 1.25.x/1.26.x

Kubernetes version 1.24.x entered maintenance mode and End of Life in July 2023. To provide access to the latest Kubernetes versions, HPE Ezmeral Runtime Enterprise5.6.x or higher version supports the following versions of HPE provided a CNCF-certified distribution:

- 1.24.8-hpe2

- 1.25.12-hpe1
- 1.26.7-hpe1

See Support Matrixes on page 54 for complete Kubernetes versions.

Prepackaged ML Ops Applications

Starting from HPE Ezmeral Runtime Enterprise 5.6.4:

- Airflow operator, Kubeflow operator, and Spark operator are not included in Prepackaged ML Ops Applications. Also, See Prepackaged Applications on page 101 for details.
- If you try to open Kubedirector notebook is removed. If you try to open notebook, A blank window will be opened without any functions.

CSI version update in HPE Ezmeral Data Fabric on Kubernetes

Starting from HPE Ezmeral Runtime Enterprise 5.6.2, Data Fabric CSI version is FUSE POSIX CSI v1.2.8 by default. See CSI column in Container Versions for HPE Ezmeral Data Fabric on Kubernetes for details.

OS Support

HPE Ezmeral Runtime Enterprise 5.6.1 or higher version adds support for SLES 15 SP4 on HPE Ezmeral Runtime Enterprise control plane (Controller, Shadow Controller, Arbiter, Gateway) nodes and Kubernetes nodes. Only rolling upgrade of nodes (one node at a time) is supported. In-place OS upgrade is not supported. See OS Support on page 85.

HPE Ezmeral Data Fabric on Bare Metal Support

HPE Ezmeral Runtime Enterprise 5.6.1 or higher version adds support for registering HPE Ezmeral Data Fabric on Bare Metal 7.2 as tenant storage. see Support Matrixes on page 54.

OS Support

HPE Ezmeral Runtime Enterprise 5.6.0 or higher version adds support for RHEL 8.x on HPE Ezmeral Runtime Enterprise control plane (Controller, Shadow Controller, Arbiter, Gateway) nodes for fresh installations. See OS Support on page 85.

Support for RHEL 7.x to RHEL 8.x major OS upgrade will be available in the future release.

Updated Versions of Open Source Components

HPE Ezmeral Runtime Enterprise 5.6.0 or higher version supports the latest versions of open-source components, including, but not limited to:

- Airflow
- ArgoCD
- Istio v1.14.5
- Kubeflow
- Open Policy Agent Gatekeeper
- Falco
- NVIDIA GPU Metrics
- NVIDIA plugin

- Kubernetes Dashboard

For more information about supported add-ons, see Support Matrixes on page 54.

Mandatory Certificate Renewal for Kubernetes Components

HPE Ezmeral Runtime Enterprise 5.6.0 or higher version provides a manual workflow for renewing the security certificate of various Kubernetes components. For more details on the Kubernetes cluster certificate management, see Procedure for updating Kubernetes cluster certificates.

GPU status per K8s pod/namespace

Starting HPE Ezmeral Runtime Enterprise 5.6.0, `per pod/namespace` GPU utilization metrics are collected into `Elasticssearch`. This can be later queried or displayed using **ElasticSearch Kibana** dashboard or **Prometheus** dashboard.

HPE Ezmeral ML Ops Enhancements

HPE Ezmeral Runtime Enterprise 5.6.0 or higher version includes the following enhancements and changes to the HPE Ezmeral ML Ops features:

Enhanced User Experience	Users can now view detailed information about deployed models through the HPE Ezmeral Runtime Enterprise UI. For information, see Viewing Model Information on page 182.

Apache Spark Analytics Enhancements

HPE Ezmeral Runtime Enterprise 5.6.0 or higher version includes the following enhancements and changes to Apache Spark on HPE Ezmeral Runtime Enterprise:

Support for Apache Spark 3.3.1 compatible with HPE Ezmeral Data Fabric on Bare Metal version 6.2.0 to 7.2.0	Includes secure access to read and write data from HPE Ezmeral Data Fabric based on the user identity.
Version Updates	• Spark is upgraded to version 3.3.1 • Hive Metastore is upgraded to version 3.1.3 • Spark History Server is upgraded to version 3.3.1 • Spark Thrift Server is upgraded to version 3.3.1.
Support for Hadoop 3 and enhanced S3 features	Spark 3.3.1 supports enhanced S3 features introduced in Hadoop 3.x.

Prepackaged Applications

HPE Ezmeral Runtime Enterprise includes the following applications out of the box. You can choose to enable or deploy them. These reference applications are included to help you get up and running quickly. However you are not limited to these applications.

For commercial and open-source applications and tools that have been validated for the HPE Ezmeral ecosystem, visit the HPE Ezmeral Marketplace.

Application Support

All the out-of-the-box applications listed in this topic are officially supported by Hewlett Packard Enterprise. Customizations and modifications to these applications, including the installation of different application versions, is not supported.

Supported applications in the HPE Ezmeral applications catalog display the following statement in their tiles: `Supported by HPE`

Prepackaged ML Ops Applications

- Airflow operator 2.3.4 (For HPE Ezmeral Runtime Enterprise 5.6.2 or earlier only)
- Kubeflow operator 1.6 (For HPE Ezmeral Runtime Enterprise 5.6.2 or earlier only)
- KubeDirector Notebook application 3.4

Prepackaged Ezmeral Analytics for Spark Applications

- Livy 0.7.0-2.4.7
- Livy 0.7.0
- Hive Metastore 3.1.3
- Spark History Server 3.3.1
- Spark Thrift Server 3.3.1
- Spark operator, supporting both 3.3.1 and 2.4.7 (For HPE Ezmeral Runtime Enterprise 5.6.2 or earlier only)
- Spark 3.3.1 images
- Spark 2.4.7 images

Prepackaged OSS KubeDirector Applications

- Ubuntu 18.04 utility
- CentOS 7.x utility

On-Premises, Hybrid, and Multi-Cloud Deployments

HPE Ezmeral Runtime Enterprise can be deployed standalone on-premises and/or on a public cloud, as well as in a hybrid deployment where some of the hosts reside on-premises (including in multiple data centers) while other hosts reside on one or more public clouds. In each case, the network requirements described in Networks and Subnets on page 111 must be satisfied. HPE Ezmeral Runtime Enterprise supports all of the major cloud providers:

- **Amazon Web Services (AWS):** Either a configurable AWS CloudFormation or a purpose-built `cepicctl` command line tool can be used to configure HPE Ezmeral Runtime Enterprise on EC2 instances.

  EC2 instances that contain NVMe SSDs are not supported in HPE Ezmeral Runtime Enterprise 5.3.1. However, EC2 instances that contain NVMe SSDs are supported in HPE Ezmeral Runtime Enterprise 5.3.5 and later releases.

- **Google Cloud Platform (GCP):** A configurable YAML deployment script in conjunction with the `glcloud` deployment tool is used to configure HPE Ezmeral Runtime Enterprise on Google Cloud compute instances (VMs).
- **Microsoft Azure:** A configurable Azure Resource Manager template is used to configure HPE Ezmeral Runtime Enterprise on Azure instances.

Benefits

HPE Ezmeral Runtime Enterprise offers the following benefits regardless of the deployment model used:

- The same general approach may be used to install and run HPE Ezmeral Runtime Enterprise regardless of how it is deployed. See Deployment Models on page 103.
- The code base and management experience are identical for a fully on-premises deployment, fully cloud-only deployment, or hybrid deployment.
- You retain complete control over your on-premises and/or cloud infrastructure. For example, you can leverage your existing application machine images (e.g. AWS AMI) for your certified RHEL operating system as well as any cloud specific features like disaster recovery. You may leverage cloud-specific features for the hosts in your deployment, such as availability zones, spanning hosts across subnets, and instance types when using virtual machines as hosts.

NOTE: This documentation uses the term "host" to refer to the physical hosts and/or virtual machines that run HPE Ezmeral Runtime Enterprise virtual nodes/containers.

- HPE Ezmeral Runtime Enterprise supports vCPU over-provisioning and can place multiple virtual nodes/containers on each host, thereby increasing host utilization and reducing costs.
- HPE Ezmeral Runtime Enterprise supports host tags that can be used to control the placement of virtual nodes/containers among on-premises hosts and/or cloud instances from AWS, Azure or GCP. This enables the placement of virtual nodes/containers based on workload (e.g. Spark, TensorFlow etc), SLA and data gravity considerations. See About Tags on page 545.
- HPE Ezmeral Runtime Enterprise provides common benefits regardless of the deployment model used (see Deployment Models on page 103), including strict tenant isolation without cloud-specific networking constructs.
- Gateway hosts use HAProxy to control access (ingress) to application service endpoints. See Gateway Hosts on page 106.

Deployment Models

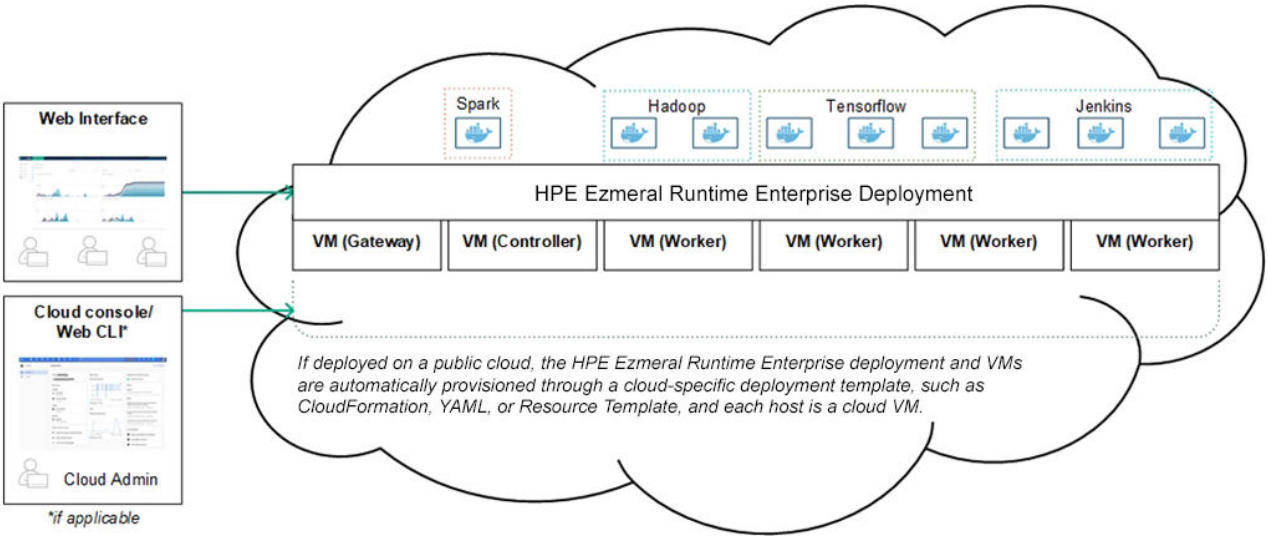
As described above, HPE Ezmeral Runtime Enterprise may be deployed as follows:

- **Entirely on-premises or on a single public cloud:** See Single Cloud on page 103.
- **Across on-premises and one or more public clouds:** See Hybrid and Multi-Cloud on page 104.

NOTE: The following diagrams illustrate generic HPE Ezmeral Runtime Enterprise deployments. See Software Components on page 113, Kubernetes on page 319, and Kubernetes Physical Architecture on page 320 for additional details.

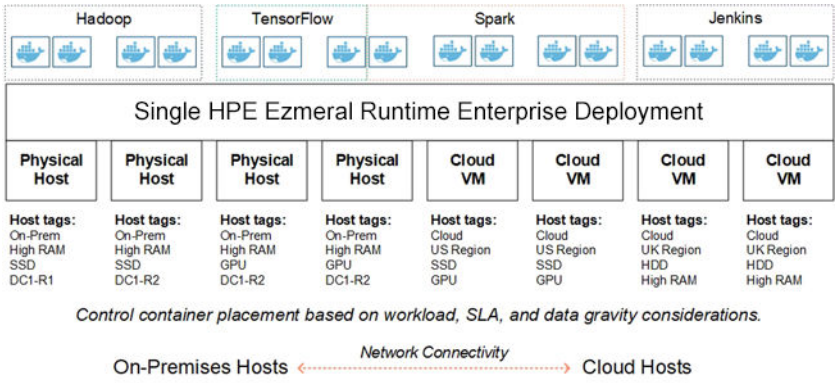
Single Cloud

This diagram depicts a deployment that has been installed either on a single public cloud or on-premises on virtual machines in a private cloud.



Hybrid and Multi-Cloud

This diagram depicts a deployment that has been installed across one or more public clouds and physical hosts on-premises.



Third-Party Licenses

Please click to download a spreadsheet that lists the third-party components in HPE Ezmeral Runtime Enterprise in Microsoft Excel format (.xlsx). In addition,HPE Ezmeral Runtime Enterprise installs a GPL-licensed Linux device driver as part of the support for the HPE Ezmeral Runtime Enterprise DataTap implementation. You may obtain the source code for this driver at no charge by contacting Hewlett Packard Enterprise Technical Support.

Universal Concepts

The articles in this section provide high-level descriptions of architecture and functions that apply across the entire deployment.

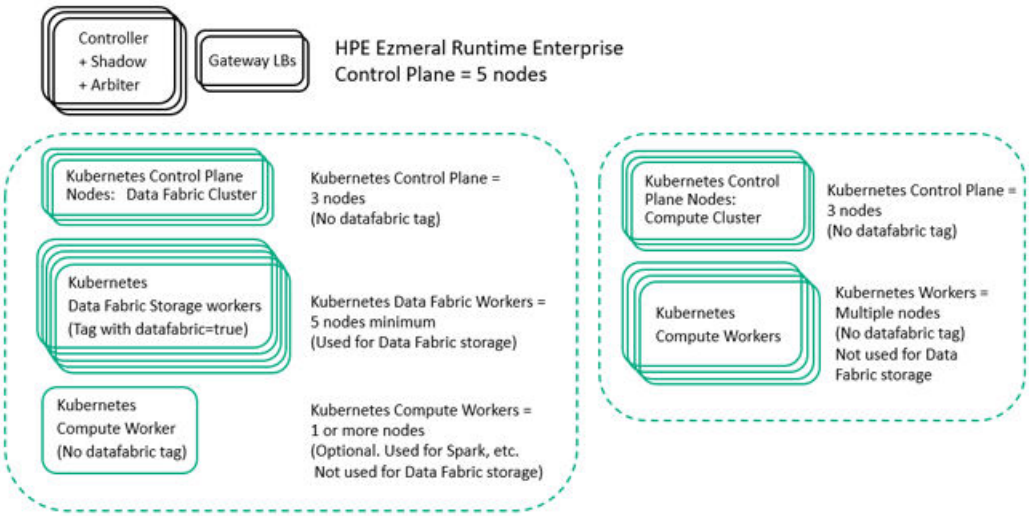
Controller, Gateway, and Worker Hosts

A host is either a physical server or a virtual server, located on your premises or in a public cloud, that is available to HPE Ezmeral Runtime Enterprise. The term **host** and **node** are often used interchangeably. Nodes are hosts that are part of a cluster.

You must have a supported operating system installed on hosts before they can be used in HPE Ezmeral Runtime Enterprise. Hosts have different requirements depending on their functions. See Host Requirements on page 813.

Logical Diagram of Hosts

The following diagram illustrates a Kubernetes deployment of HPE Ezmeral Runtime Enterprise that includes a Kubernetes cluster used for compute jobs and a separate Kubernetes cluster that is an implementation of HPE Ezmeral Data Fabric on Kubernetes.



Controller Hosts

The Controller host is the host where you initially install HPE Ezmeral Runtime Enterprise. This host controls the rest of the hosts in the deployment.

In high-availability (HA) deployments, there is also a Shadow Controller host and an Arbiter host, for a total of three (3) Controller hosts.

Controller hosts are part of the HPE Ezmeral Runtime Enterprise control plane. HPE Ezmeral Runtime Enterprise control plane hosts are not part of any Kubernetes cluster.

Gateway Hosts

Gateway load balancer (Gateway LB) hosts enable access to pods or container services from an external network. See Gateway Hosts.

In high-availability (HA) deployments, there are a minimum of two (2) Gateway LB hosts. For more information about Gateway host requirements, see Gateway Hosts on page 106.

Gateway LB hosts are part of the HPE Ezmeral Runtime Enterprise control plane. HPE Ezmeral Runtime Enterprise control plane hosts are not part of any Kubernetes cluster. Gateway hosts are added to HPE Ezmeral Runtime Enterprise in a separate installation procedure.

Kubernetes Control Plane Hosts

Hosts become Kubernetes nodes in a two-phase process. First the host is added to the HPE Ezmeral Runtime Enterprise deployment. Then the host is added to the Kubernetes cluster as a Kubernetes control plane node or as a worker node.

The Kubernetes control plane nodes manage the Kubernetes worker nodes and pods in the Kubernetes cluster. For detailed information about what a Kuberentes control plane does, see Control Plane

Components in the Kubernetes documentation (links opens an external website in a new browser window or tab).

In high-availability (HA) deployments that implement Kubernetes, there are a minimum of three (3) Kubernetes control plane nodes.

Kubernetes control plane nodes, formerly known as master nodes, control the Kubernetes cluster, but are not considered part of the HPE Ezmeral Runtime Enterprise control plane.

Worker hosts

Hosts become Kubernetes nodes in a two-phase process. First the host is added to the HPE Ezmeral Runtime Enterprise deployment. Then the host is added to the Kubernetes cluster as a Kubernetes control plane node or as a worker node.

Worker nodes run the containers and pods that process jobs in HPE Ezmeral Runtime Enterprise. Worker nodes are managed by the Kubernetes control plane for that cluster.

See also Data Fabric worker hosts

Data Fabric Worker Hosts

Data Fabric hosts are the hosts that have `Datafabric` tag enabled. These hosts can become the Data Fabric worker storage nodes in an implementation of **HPE Ezmeral Data Fabric on Kubernetes**.

Related concepts

Gateway Hosts on page 106
Gateway hosts run the HAproxy service and are part of the HPE Ezmeral Runtime Enterprise control plane. You can access the web UI of a HPE Ezmeral Runtime Enterprise deployment through any Gateway host. For high availability and load balancing, configure multiple Gateway hosts.

Related reference

Kubernetes Physical Architecture on page 320

High Availability on page 132
High availability (HA) in deployments of HPE Ezmeral Runtime Enterprise is divided into platform controller HA, gateway HA, and cluster HA.

More information

Host Requirements on page 813
This topic lists the minimum host requirements for HPE Ezmeral Runtime Enterprise for production environments and for non-production environments, such as for development and testing.

Gateway Hosts

Gateway hosts run the HAproxy service and are part of the HPE Ezmeral Runtime Enterprise control plane. You can access the web UI of a HPE Ezmeral Runtime Enterprise deployment through any Gateway host. For high availability and load balancing, configure multiple Gateway hosts.

Gateway hosts are part of the HPE Ezmeral Runtime Enterprise control plane. Gateway hosts must conform to the applicable requirements listed in Gateway Host Requirements and Limitations on page 108 and in Host Requirements on page 813.

Gateway hosts do not run pods. Instead, they enable access to user-facing services such as Notebooks, Hue console, and SSH running on pods through an instance of the High Availability Proxy service (HAproxy service on page 110). The management Web UI is accessible through any Gateway host in the HPE Ezmeral Runtime Enterprise deployment.

Gateway hosts allow for clear separation of network zones by providing the following:

- A simple, secure, and fully-managed control path for end users and admins alike to access Kubernetes API servers (such as when handling `kubectl` commands) as well as the service endpoints of multiple Kubernetes clusters. HPE Ezmeral Runtime Enterprise dynamically manages endpoints, including `kubeconfig` contents, as clusters and services are created, deleted, or updated.
- Automated load balancing for Kubernetes masters and services. Kube API traffic (via kubectl etc) is load-balanced to both multi-master highly-available Kubernetes clusters and multi-replica NodePort services.
- SSL termination to container service access points.
- Interoperability with any ingress controller and NodePort service definition for maximum flexibility.

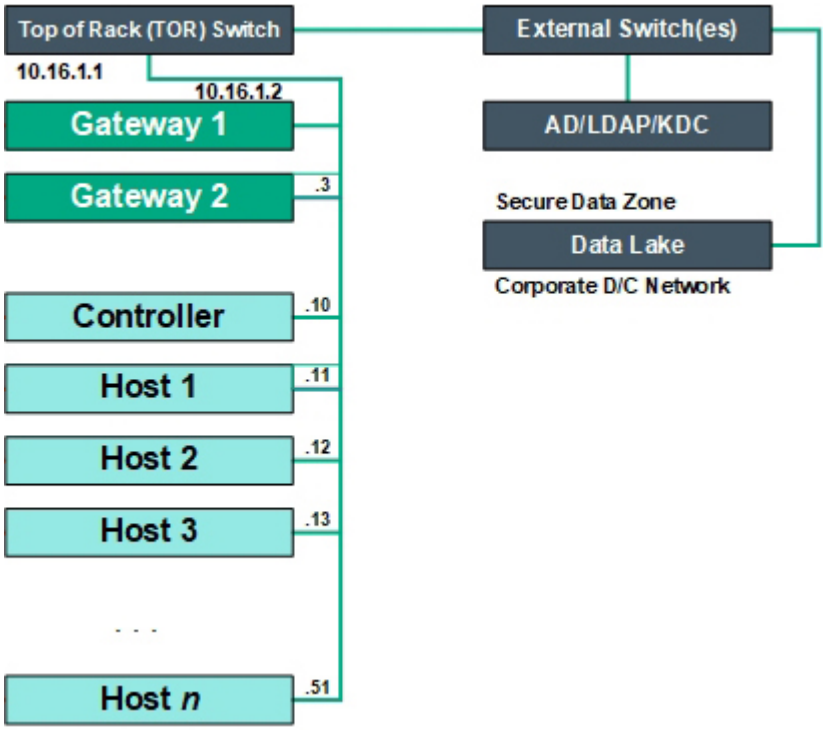
You can configure multiple Gateway hosts one or more Gateway sets that have a common Fully Qualified Domain Name (FQDN) for round-robin load balancing and High Availability. You can also use a hardware load balancer in front of the multiple Gateway hosts, and you can also configure one or more custom port ranges between 10000 and 50000 for use as proxies.

All control traffic **to** the pods from end-user devices (browsers and command line), such as HTTPS, SSH, or AD/KDC, goes through the Gateway hosts, while all traffic **from** the pods is routed through the hosts on which those pods reside.

Support for multiple subnets increases Gateway host flexibility. For example, you can use "small" virtual machines that meet all Gateway host requirements located on different racks or in different areas of your network, instead of having to place these hosts on the same rack as Kubernetes cluster hosts. This configuration can help optimize resource usage.

Physical Architecture

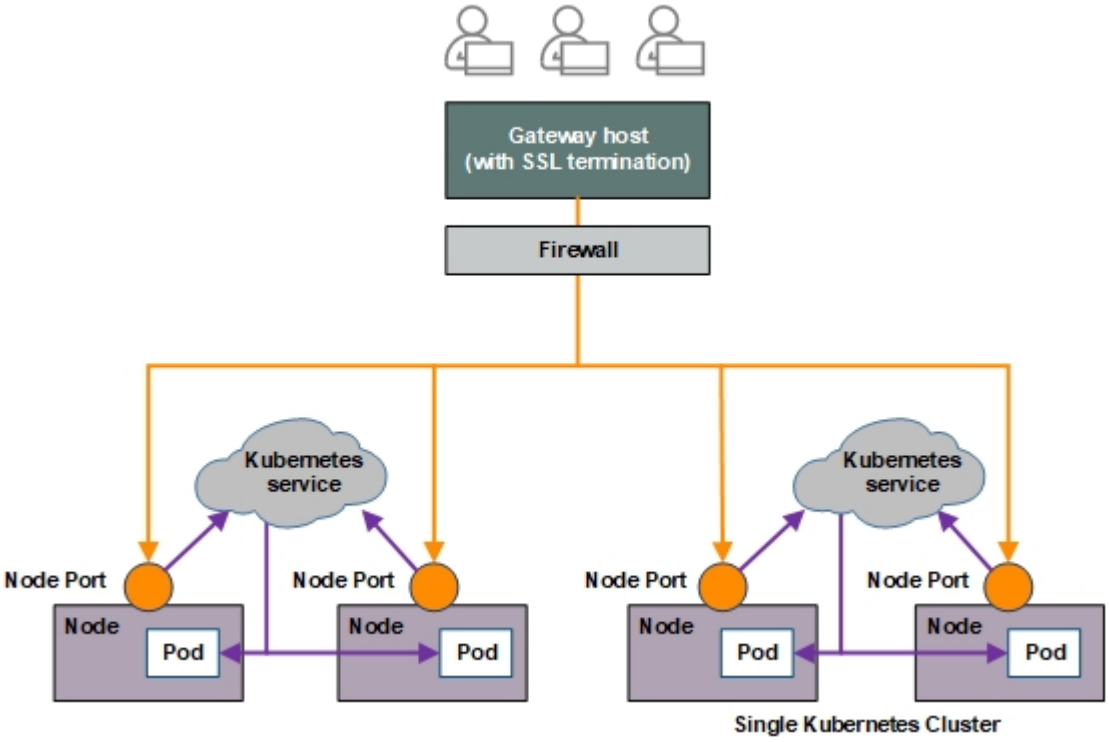
This diagram displays the physical architecture of a deployment that has two Gateway hosts.



Logical Architecture

You can access the web UI of a HPE Ezmeral Runtime Enterprise deployment through any Gateway host.

The following diagram displays the logical architecture of Kubernetes clusters and Gateway hosts within a deployment:



Gateway Host Requirements and Limitations

A deployment of HPE Ezmeral Runtime Enterprise must include at least one Gateway host. For high availability and load balancing, a deployment can include multiple Gateway hosts in one or more Gateway sets.

Gateway hosts only run the HAproxy service on page 110 and cannot be included in Kubernetes clusters. Gateway hosts must meet the following minimum requirements:

- For information about the CPU, memory, and storage requirments for Gateway hosts, see the information about Gateway load balancer (Gateway LB) hosts in Host Requirements on page 813.
- Gateway hosts need not be on the same subnet as the Controller, Shadow Controller, or other hosts.
- Ports 10000-50000 on the Gateway hosts are used for port mapping. The `sysctl` utility is configured on the Gateway hosts to prevent misallocation of these port ranges. The `HAproxy` service binds to ports in this range.
- The `iptables` service is automatically disabled on the Gateway hosts during the installation.
- For both DataTap access and for Kerberos access from the containers, physical hosts must be on a routable network/standard corporate subnet.

Gateway Sets

A Gateway set is a set of Gateway hosts that share a common DNS server name. You create a Gateway set during Gateway host installation when you specify multiple IP addresses for the same hostname. You can create more than one Gateway set by performing multiple Gateway host installation operations, specifying a different hostname and a different list of IP addresses during each operation.

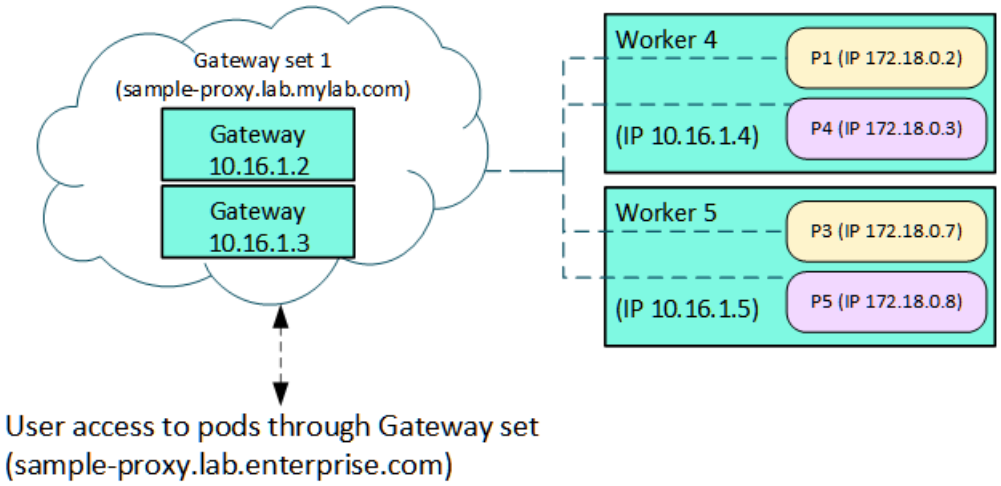
Multiple Gateway sets can be used to create a larger number of port mappings than are allowed for a single set.

Gateway sets have the following requirements and limitations:

- All of the hosts in a Gateway set must have both an individual hostname and an externally-resolvable common hostname. The **Gateway Hostname** must be **all lower case** as per the Linux hostname naming convention. The DNS server must be configured to do this. For example, the following two Gateway hosts have the common hostname `sample-lab-proxy.enterprise.com`:
  - `10.32.2.94 hostname-1.enterprise.com sample-lab-proxy.enterprise.com`
  - `10.32.2.96 hostname-2.enterprise.com sample-lab-proxy.enterprise.com`
- A new Gateway host can be added to an existing set at any time. The new Gateway host will automatically be configured with the port instance mappings for that set.
- A new Gateway host can be added to start a new set at any time. However, the newly added Gateway host will be used only for future service mappings.
- At the time of instance launch, all Gateway hosts in the set must be accessible to create mappings. If they are not available, then cluster creation will fail.
- A Gateway host that belongs to a Gateway set can be decommissioned provided the following conditions are met:
  - No active pod port mappings are present.
  - Pod port mappings are present and there is at least one other Gateway host available in the set.

Proxy Mapping Management

Configuring one or more Gateway hosts enables users to access pods using a set of service endpoint proxy mappings, as shown in the following illustration:



Each Gateway set can consist of one or more Gateway hosts. When a cluster is created, information is collected about the service endpoints configured for that cluster. These service endpoints will be those defined for services being deployed within that cluster, and each service endpoint will be mapped to a specific port regardless of whether that service uses HTTP, HTTPS, or TCP.

The Controller uses a scheduling algorithm to decide which Gateway set to use for creating port mappings. This is simply based on which Gateway set has fewer port mappings and whether or not all of the hosts in the Gateway set are accessible. If the Controller cannot find an available Gateway set, then cluster launch will fail with an error message returned to the user.

Once the Controller identifies an available Gateway set, it allocates the necessary ports from the reserved range. A message is sent to the service running on each Gateway host to create the appropriate proxy mappings. Mappings will be automatically deleted when a cluster is deleted, and the ports will be freed up for use by the next cluster.

Gateway hosts within a Gateway set have failover ability. If a Gateway host is down, the other hosts provide the port mappings. However, all active hosts within a set must be available at the time of cluster creation. The user's DNS server determines which specific host will receive the traffic within a single set. Typically, a round-robin configuration on the DNS server serves this purpose.

Mappings between Gateway sets are not shared. If an entire Gateway set is disabled, those mappings will no longer function.

HAproxy service

The HAproxy service enables load balancing and SSL termination. After the Gateway hosts are set up, the HPE Ezmeral Runtime Enterprise Controller automatically starts using the Gateway proxy set for container port mappings.

Service endpoints for a Kubernetes cluster in a deployment appear in the **Gateway Mappings** column of the table on the **Kubernetes Service Endpoints** screen. See Endpoints Tab.

HAproxy Stats UI

HAProxy provides a "Stats" web UI for monitoring the health of all back-end service endpoints. The UI is available over HTTP on port 8081 of each Gateway host, with username `haproxy` and password `haproxy`. The Gateway host uses the Stats web UI to provide the HAProxy service status information on the HPE Ezmeral Runtime Enterprise Dashboard.

- Beginning with HPE Ezmeral Runtime Enterprise 5.4.1, the HAProxy Stats web page is configured by default to accessible from the local host only. To enable temporary access from other hosts, do the following on each Gateway host:

  1. Open the file `/opt/bluedata/common-install/scripts/haproxy/haproxy_globals.cfg` for editing.

  2. In the `listen stats` section, find the following line:

     ```
     bind 127.0.0.1:8081
     ```

  3. Change this line to the following:

     ```
     bind 0.0.0.0:8081
     ```

  4. Restart the service by executing the following command on the Gateway host:

     ```
     systemctl restart bds-controller
     ```

  This change does not persist across software upgrades, and is not replicated onto new hosts.

- For HPE Ezmeral Runtime Enterprise 5.4.0 and prior releases, the HAProxy Stats web page is configured by default to be accessible from any host on the network. To restrict access to the local host only, do the following on each Gateway host:

  1. Open the file `/opt/bluedata/common-install/scripts/haproxy/haproxy_globals.cfg` for editing.

2. In the `listen stats` section, find the following line:

```
bind 0.0.0.0:8081
```

3. Change this line to the following:

```
bind 127.0.0.1:8081
```

4. Restart the service by executing the following command on the Gateway host:

```
systemctl restart bds-controller
```

This change does not persist across software upgrades, and is not replicated onto new hosts.

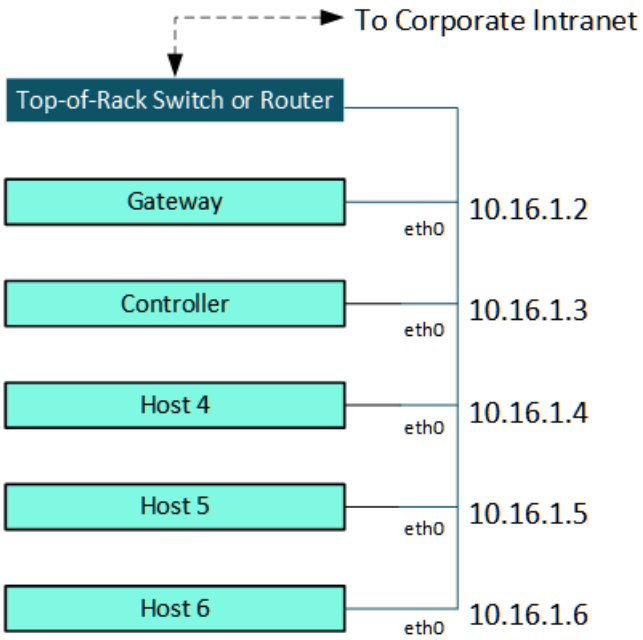
Networks and Subnets

HPE Ezmeral Runtime Enterprise hosts are connected through an external switch that is accessible to the network that your organization manages. Internally within HPE Ezmeral Runtime Enterprise, Kubernetes pods communicate through a private nonroutable virtual network that is not visible to the host network.

Routable Host Network

Hosts in a deployment of HPE Ezmeral Runtime Enterprise are typically deployed as one or more racks of servers that are connected to an external top-of-rack (ToR) switch. Each host (Controller, Shadow Controller, Arbiter, Gateway, and all Kubernetes hosts) has an IP address (such as **10.16.1.5**) and an FQDN. This network must be both routable and part of the network that the IT department of your organization manages.

HPE Ezmeral Runtime Enterprise hosts can be deployed across multiple subnets, subject to certain requirements. See Multiple Subnets on page 112.

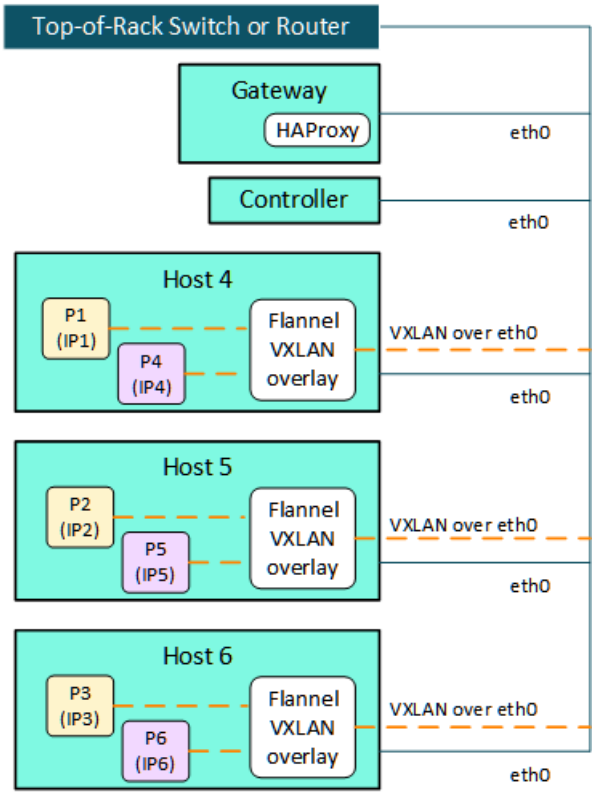


Private Nonroutable Pod Network

Kubernetes pods communicate with each other through an internal virtual network that is private (nonroutable) and managed by HPE Ezmeral Runtime Enterprise. This internal network is not accessible from the top-of-rack switch and is separate from the routable host network.

The virtual network for pods uses a VxLAN overlay. Canal, which combines Calico and Flannel, is used as the pod Network (CNI) Network Provider. This private, nonroutable pod network keeps the pod IP addresses hidden within the private network.

In Kubernetes, every container in a pod shares a network namespace, including the IP address and network ports. Kubernetes assigns an internal IP address to each pod, which appears in the following diagram as **P1 (IP1)**, **P2 (IP2)**, and so forth. Tenant network isolation is achieved by using Kubernetes Network Policies (link opens an external website in a new browser tab or window).



The Gateway hosts act as a proxy for accessing services that are running inside the pods. IP masquerading replaces the IP addresses of outgoing packets:

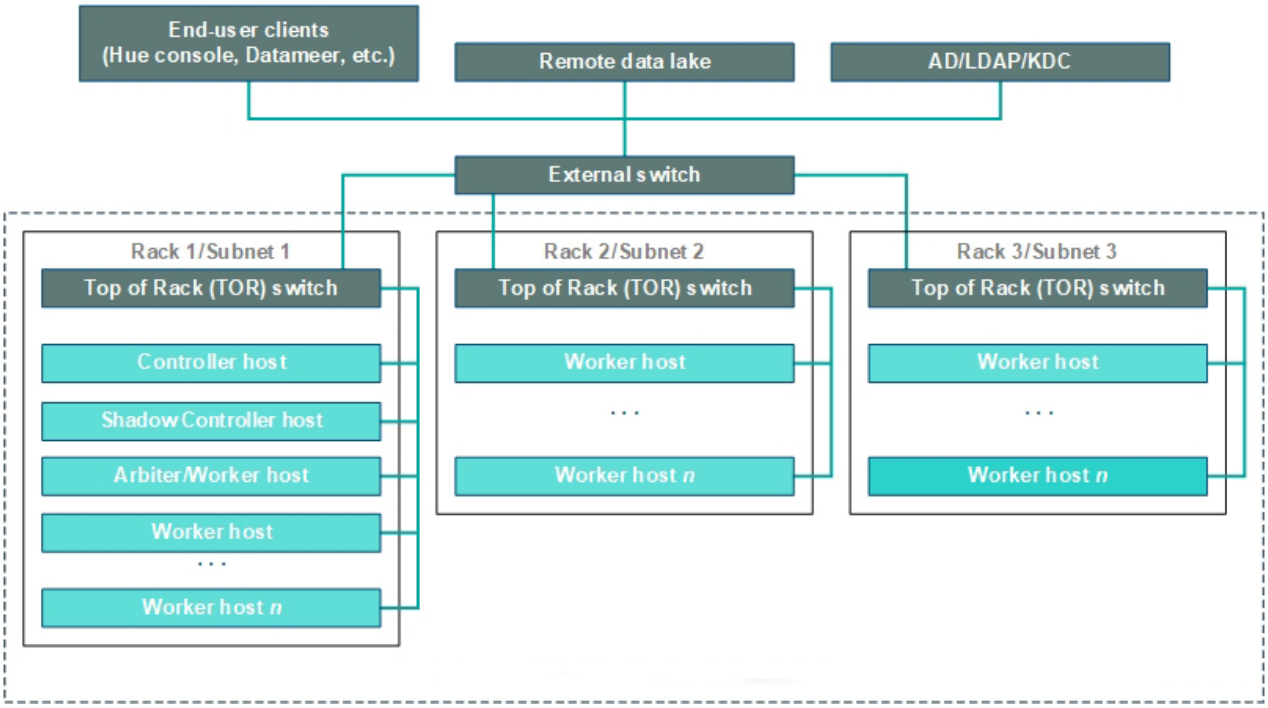
- End-user access to services in the pods (such as Jupyter Notebook or web applications) is routed through a Gateway host that runs the HAProxy service.
- Traffic that originates from pods is routed through the host network interface masquerading. Examples of traffic from pods includes accessing a remote HDFS, accessing enterprise systems such as Active Directory (AD), MIT KDC (Kerberos provider), SSO (Identity providers), and Certificate Authority (CA), and so forth.

Multiple Subnets

HPE Ezmeral Runtime Enterprise can be deployed across multiple subnets.

The use of multiple subnets is subject to the requirements described in Network Requirements on page 825.

The following diagram illustrates a sample deployment that uses multiple subnets.

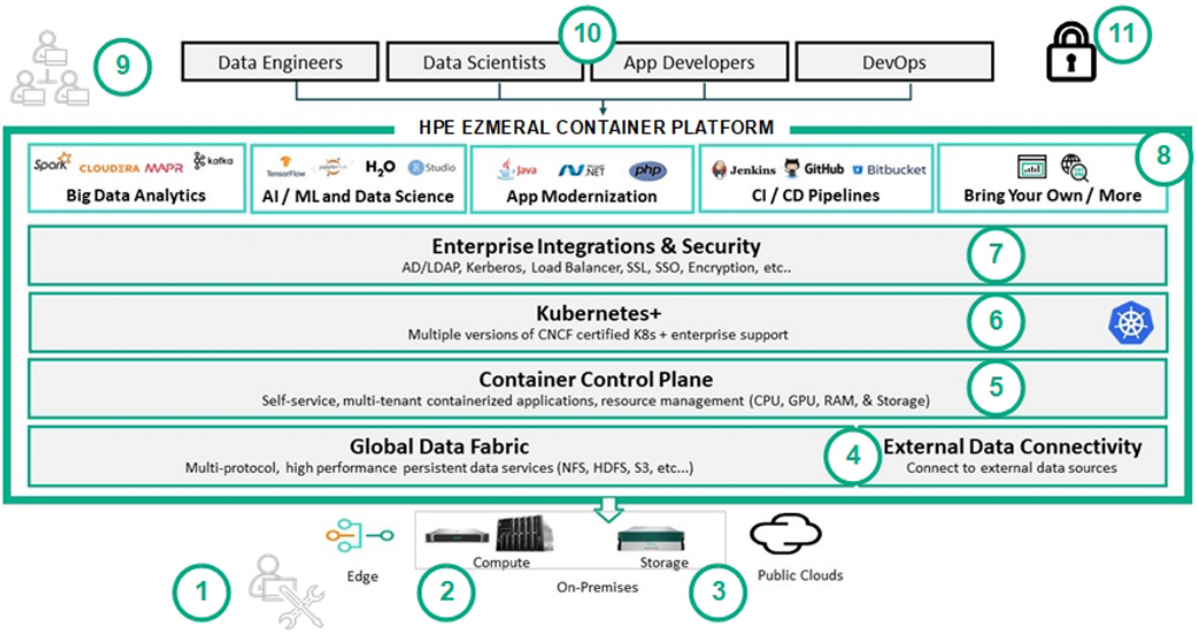


Related reference

Network Requirements on page 825

Software Components

HPE Ezmeral Runtime Enterprise is an enterprise-grade software platform that forms a layer between the underlying infrastructure and applications, transforming that infrastructure into an agile and flexible platform for virtual clusters running on containers.



last-updated: May 06, 2024

NOTE:

See the following articles for additional information about the following scenarios:

- Getting Started with General Kubernetes Functionality, when running Big Data tenants in Kubernetes clusters.
- AI and ML Project Workflow on page 150, when running AI/ML projects in Kubernetes clusters.

The high-level architecture is as follows (numbers correspond to the callouts in the preceding image):

- **Platform Administrator (1):** One or more Platform Administrators handle overall administration, including managing hosts and creating tenants or projects. A Kubernetes Administrator can create Kubernetes clusters.
- **Hosts (2):** Physical and/or virtual machines. See Controller, Gateway, and Worker Hosts and Gateway Hosts.
- **Data Storage Resources (3):** Available on-premises and/or cloud-based storage resource. This comprises the following:
  - **Data Source:** This is where persistent job data required by the tenants/projects and virtual clusters is read and written. A data source is typically a DataTap: a shortcut that points to existing remote data storage locations on your network. A special **TenantStorage** DataTap is constructed from local storage to the hosts. DataTaps reduce or even eliminate the need to copy large volumes of data to and from the virtual clusters before and after running jobs, thus saving time and reducing network traffic. Please see About DataTaps.
  - **Cluster file system:** This is the storage where temporary data that is generated while running jobs within a given cluster is read and written. The cluster file system is built within the virtual cluster, on storage taken from the node storage space of the underlying host (on-premises and/or a remote storage resource).
  - **Unique file directories for each tenant/project:** Each tenant or project has its own sandboxed shared-storage area within the tenant/project storage space, whether on-premises or on the public cloud. This per-tenant storage can be used to isolate data that should be accessible by only one tenant or project. Optionally, it can be used to enforce a quota on the tenant's/project's use of that storage capacity.
- **Data Resources (4)**: A wide variety of storage protocols used by high-performance persistent data services, such as NFS, HDFS, and S3, are supported. Connectivity to existing external data sources is supported via both DataTaps and FS Mounts. See About DataTaps and About FS Mounts, respectively.
- **Container Control Plane (5):** The control plane consists of the services that are installed on each of the hosts. HPE Ezmeral Runtime Enterprise automatically handles the back-end virtual cluster management, thereby eliminating the need for complex, time-consuming IT support. Platform and Tenant/Project Administrator users can perform all of these tasks in moments using the web interface.
- **Kubernetes (6):**HPE Ezmeral Runtime Enterprise includes built-in support for Kubernetes clusters, tenants/projects, and pods. See Kubernetes Physical Architecture for an overview of the Kubernetes implementation.
- **Enterprise Integrations and Security (7):** Built-in features help ensure a seamless integration between HPE Ezmeral Runtime Enterprise and your existing enterprise infrastructure, including:
  - Built-in user roles (Platform Administrator, Tenant Administrator, and Member) that allow you to control who can see certain data and perform specific functions. Roles are granted on a per-tenant or per-project basis, meaning that you can either restrict users to a single tenant/project or grant access to multiple tenants/projects. Each user may have at most one role per tenant/project.

  - Authenticating users via either the internal user database or your existing AD/LDAP setup.
  - Kerberos encryption for data traveling within the deployment and between the deployment and your existing infrastructure.
  - Load balancing for optimal resource usage.
  - SSL connections to the web interface for added protection.
  - SSO support to simplify user access.
- **Tenants or Projects (8):** Tenants and/or AI/ML projects allow you to restrict access as needed, such as by department. Each tenant or project has its own unique sets of authorized users, DataTaps, applications, and virtual clusters that are never shared with other tenants/projects. Users with access to one tenant or project cannot access or modify any aspect of another tenant/project unless they have also been assigned a role (Tenant/Project Administrator or Member) on that tenant or project. Each tenant/project runs one or more virtual clusters that are created to run a wide variety of Big Data or AI/ML/DL applications, services, and jobs.
- **Tenant/Project Administrators (9):** A Tenant or Project Administrator manages the resources assigned to that tenant or project. Each tenant or project must have at least one user with the Tenant Administrator or Project Administrator role, as appropriate.
- **End users (10):** Tenant/Project Member users access virtual clusters within tenants to perform jobs.

Virtual Cores, RAM, Storage, and GPU Devices

NOTE: This article uses the term "tenant" to refer to both tenants and projects.

Virtual CPU (vCPU) cores are modeled as follows:

- The license specifies the maximum number of CPU cores that can be surfaced by the set of on-premises and/or public cloud hosts in a given HPE Ezmeral Runtime Enterprise deployment. Starting with Container Platform version 3.8, the use or non-use of CPU hyperthreads does not impact the license and vCPU count.
- The number of available vCPU cores is the number of physical CPU cores multiplied by the CPU allocation ratio specified by the Platform Administrator. For example, if the hosts have 40 physical CPU cores and the Platform Administrator specifies a CPU allocation ratio of 3, then a total of 120 vCPU cores will be displayed as available. You can allocate an unlimited number of vCPU cores to each tenant or project. The collective core usage for all nodes (containers) within a tenant/project will be constrained by either the tenant's assigned quota or the available cores in the system, whichever limit is reached first. The tenant quotas and the CPU allocation ratio act together to prevent tenant members from overloading the system's CPU resources.
- When two or more nodes are assigned to the same host, they contend for the same physical CPU resources of that host. CPU resources are allocated to such nodes in a ratio determined by their vCPU core count. For example, a node with 8 cores will receive twice as much CPU time as a node with 4 cores.
- The Platform Administrator can also specify a Quality of Service (QOS) multiplier for each tenant or project. In the case of CPU resource contention, the node vCPU count is multiplied by the tenant/project QOS multiplier when determining the physical CPU time that will be allotted to each container running within a given tenant or project. For example, a node with 8 vCPU cores in a tenant or project with a QOS multiplier of 1 will receive the same physical CPU time as a node with 4 vCPU cores in a tenant or project with a QOS multiplier of 2. The QOS multiplier is used to describe relative tenant/project priorities when CPU resource contention occurs; it does not affect the overall cap on CPU load established by the CPU allocation ratio and tenant/project quotas.

RAM is modeled as follows:

- The total amount of available RAM is equal to the amount of unreserved RAM. Unreserved RAM is the amount of RAM remaining after reserving some memory in each host for platform services. For example, if your deployment consists of four hosts that each have 128GB of physical RAM with 110GB of unreserved RAM, the total amount of RAM available to share among tenants or projects will be 440GB.
- You may allocate an unlimited amount of RAM to each tenant/project. The collective RAM usage for all nodes within a tenant or project will be constrained by either the tenant's or project's assigned quota or the available RAM in the system, whichever limit is reached first.

Storage is modeled as follows:

- Root disk storage space is allocated from the disks on each Worker host that are assigned as Node Storage disks when adding the Worker to the platform. Each node consumes Node Storage space equivalent to its root disk size on the Worker host where that node is placed.

If compatible GPU devices are present, then they are modeled as follows:

- You must install the NVIDIA drivers on the hosts before deploying HPE Ezmeral Runtime Enterprise, as described in GPU Driver Installation on page 838.
- The total number of available GPU resources is equal to the number of physical GPU devices. For example, if your deployment consists of four hosts that each have 8 physical GPU devices, then there will be a total of 32 GPU devices available to share among tenants and/or projects.
- Quotas on (tenant) namespaces for GPUs are applied by the `nvidia.com/gpu` specifier, which applies to physical GPUs and MIG instances in `single` strategy only. For example, specifying a quota of three devices of 1g.5gb is not supported.
- You may allocate an unlimited number of GPU resources to each tenant or project. The collective GPU resource usage for all nodes within a tenant or project will be constrained by either the tenant's or project's assigned quota or the available GPU devices in the system, whichever limit is reached first.
- GPU devices are expensive resources, and their usage is maximized as follows:
  - If a container requires GPU resources, then HPE Ezmeral Runtime Enterprise attempts to place that container in such a way as to maximize GPU resource utilization on a given host and to reduce or eliminate wasted resources.
  - HPE Ezmeral Runtime Enterprise does not have the concept of a virtual GPU. This means that a container deployed on one host cannot access the GPU resources of another host. Containers are limited to accessing GPUs only on the host where they are deployed.
  - HPE Ezmeral Runtime Enterprise does not allow sharing the same GPU device between multiple containers simultaneously. Once a GPU device is allocated to a given container, that container has exclusive access to that GPU.

Default values will appear in the various quota fields when you are creating a tenant/project. These default values will be 25% of the total system resources for most fields. The exception to this rule is the quota for GPU devices where the default value is 0. When configuring each resource quota, the web interface displays the total available amount of that resource for comparison. You may edit these quota values or delete a value and leave the field blank to indicate that the tenant does not have a quota defined for that resource.

Assigning a quota of resources to a tenant does not reserve those resources for that tenant when that tenant is idle (not running one or more clusters). This means that a tenant may not actually be able to acquire system resources up to the limit of its configured quota.

- You may assign a quota for any amount of resources to any tenants regardless of the actual number of available system resources. A deployment where the total amount of configured tenant resource quotas exceeds the current amount of system resources is called over-provisioning. Over-provisioning occurs when one or more of the following conditions are met:
  - You have a tenant which has resource quotas that either exceed the system resources or are undefined. This tenant will only be able to obtain the amount of resources that are actually available. This arrangement is typically a convenience to make sure that the tenant is always able to fully utilize the platform, even if you add more hosts in the future.
  - You have multiple tenants where none have overly large or undefined quotas, but where the sum of their quotas exceeds the resources currently available. In this case, you are not expecting all tenants to attempt to use all their resource quotas simultaneously. Still, you have given each tenant the ability to claim more than its “fair share” of resources when these extra resources are available. In this case, you must balance the need for occasional bursts of usage that may exceed quota resources against the need to restrict how much a “greedy” tenant can consume. A larger quota gives more freedom for burst consumption of unused resources while also expanding the potential for one tenant to prevent other tenants from fully utilizing their quotas.

NOTE: Over-provisioning is useful in certain situations; however, avoiding over-provisioning prevents potential resource conflicts by ensuring that all tenants are guaranteed to be able to obtain their configured quota of virtual CPU cores, RAM, and GPU devices.

Tenants and Projects

Tenants are created by the Platform Administrator after the Controller host has been installed. The infrastructure resources (e.g. CPU, memory, GPU, storage) available on the Worker hosts are split among the tenants on the platform. Each tenant is allocated a set of resources and restricts access to a set of data to only those users authorized to access the tenant. Resources used by one tenant cannot be used by another tenant. All users who are members of a tenant can access the resources and data objects available to that tenant.

Tenants are isolated by default, meaning that the resources in one tenant cannot view or access the resources in any other tenant.

You will need to decide how to create tenants to best suit your organizational needs, such as by:

- **Office location:** If your organization has multiple office locations, you could choose to create one or more tenants per location. For example, you could create a tenant for the San Francisco office and one for the New York office. Location is not a factor when creating tenants; this is just an example of how you could use a tenant.
- **Department:** You could choose to create one or more tenants for each department. For example, you could create one tenant each for the Manufacturing, Marketing, Research & Development, and Sales departments.
- **Use cases, application life cycle, or tools:** Different use cases for Big Data analytics and data science may have different image/resource requirements.
- **Combination:** You could choose to create one tenant by department for each location. For example, you could create a tenant for the Marketing department in San Francisco and another tenant for the Marketing department in New York.

Some of the factors to consider when planning how to create tenants may include:

- **Structure of your organization:** This may include such considerations as the departments, teams, and/or functions that need to be able to run jobs.
- **Use cases/tool requirements:** Different use cases for Big Data analytics and data science may have different image and resource requirements.

- **Seasonal needs:** Some parts of your organization may have varying needs depending on the time of year. For example, your Accounting department may need to run jobs between January 1 and April 15 each year but have little to no needs at other times of the year.
- **Amount and locations of hosts:** The number and locations of the hosts on which you will deploy HPE Ezmeral Runtime Enterprise may also be a factor. If your hosts are physically distant from the users who need to run jobs, then network bandwidth may become an important factor as well.
- **Personnel who need access:** The locations, titles, and job functions of the people who will need to be able to access the deployment at any level (Platform Administrator, Tenant Administrator, or Tenant Member) may influence how you plan and create tenants.
- **IT policies:** Your organization's IT policies may play a role in determining how you create tenants and who may access them.
- **Regulatory needs:** If your organization deals with regulated products or services (such as pharmaceuticals or financial products), then you may need to create additional tenants to safeguard regulated data and keep it separate from non-regulated data.

These are just a few of the possible criteria you must evaluate when planning how to create tenants. HPE Ezmeral Runtime Enterprise has the power and flexibility to support the tenants you create regardless of the schema you use. You may create, edit, and delete tenants at any time. However, careful planning for how you will use your deployment that includes the specific tenants your organization will need now and in the future will help you better plan your entire deployment from the number and type of hosts to the tenants you create.

Namespaces

This article describes Kubernetes namespaces in HPE Ezmeral Runtime Enterprise.

Kubernetes Namespaces

All Kubernetes resources, other than nodes and persistent storage volumes, exist within a namespace.

Namespaces are partially isolated environments that run inside a single physical Kubernetes cluster. This allows different teams, projects, and customers to share a Kubernetes cluster. Namespaces have separate pods and resources, but cannot be nested and can still communicate with each other.

Kubernetes namespaces have the following uses:

- **Isolation:** Teams, projects, and customers exist in their own environment within a cluster, and do not impact each other's work.
- **Security:** Use access controls to limit users or processes to certain namespaces.
- **Resource control:** Use resource quotas to divide a cluster's resources between teams and users.
- **Organization:** Separate development, testing, and production environments into different namespaces on one cluster.
- **Performance:** Use multiple namespaces on the same cluster to reduce the number of items the Kubernetes API must search when performing operations.

For more information on using namespaces, see the Namespaces page in the Kubernetes documentation.

kubectl Commands for Namespaces

- Create a namespace:

```
kubectl create namespace
```

- View namespaces:

```
kubectl get namespace
```

- Set a different namespace as default:

```
kubectl config set-context --current --namespace=<namespace>
```

- Delete a namespace:

  CAUTION: This action cannot be undone.

```
kubectl delete namespace
```

Reserved Namespaces

Reserved namespaces are already in use by HPE Ezmeral Runtime Enterprise. You cannot adopt reserved namespaces for your Kubernetes tenants.

If you deploy optional add-ons, HPE Ezmeral Runtime Enterprise reserves additional namespaces.

Platform and Cluster Administrators can query a deployed cluster for the reserved namespaces with the following command:

```
kubectl get hpecpconfig -n hpecp -o
jsonpath='{.items[0].spec.reservedNamespaceNames}' | tr , ' ' | tr -d '[]'
```

The HPE Ezmeral Runtime Enterprise default reserved namespaces are:

- `airflow-base`
- `airflowop-system`
- `argocd`
- `auth`
- `cert-manager`
- `default`
- `ezctl`
- `ezmysql`
- `ezml-model-mgmt`
- `gatekeeper-system`
- `hpe-csi`
- `hpe-externalclusterinfo`
- `hpe-ldap`
- `hpe-nfscsi`

- `hpe-nodesvc`
- `hpe-secure`
- `hpe-sparkoperator`
- `hpe-storage`
- `hpe-system`
- `hpe-templates-compute`
- `hpecp`
- `hpecp-bootstrap`
- `hpecp-cert-manager`
- `hpecp-falco`
- `hpecp-observability`
- `istio-system`
- `kd-apps`
- `kd-mlops`
- `kd-spark`
- `kiali-operator`
- `knative-eventing`
- `knative-serving`
- `kube-node-lease`
- `kube-public`
- `kube-system`
- `kubeflow`
- `kubeflow-jobs`
- `kubeflow-operator`
- `kubeflow-user-example-com`
- `kubernetes-dashboard`
- `mapr-external-info`
- `prism-ns`
- `velero`

If a Kubernetes Data Fabric cluster is deployed, HPE Ezmeral Runtime Enterprise also reserves the namespace corresponding to the name of the Data Fabric cluster.

For example, if a Data Fabric cluster is named `df-cluster`, HPE Ezmeral Runtime Enterprise reserves the `df-cluster` namespace.

Tenant/Project Storage

NOTE: This article uses the term "tenant" to refer to tenants and AI/ML projects.

Tenant storage is an optional storage location that is shared by all nodes within a given tenant. Tenant storage can be configured to use HPE Ezmeral Data Fabric on Bare Metal, HPE Ezmeral Data Fabric on Kubernetes, or a remote NFS system. To use an HPE Ezmeral Data Fabric implementation as tenant storage, you must register HPE Ezmeral Data Fabric. See HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579. Alternatively, you can create a tenant without dedicated storage.

NOTE: If all tenants are created using the same tenant storage service settings, then no tenant can access the storage space of any other tenant.

When a new tenant is created, that tenant automatically receives a DataTap called **TenantStorage** that points at a unique directory within the tenant storage space. This DataTap can be used in the same manner as other DataTaps, but it cannot be edited or deleted. This does not apply if tenant storage has not been defined (meaning that you selected **None** for Tenant Storage during installation, as described in Platform Controller Setup).

The **TenantStorage** DataTap points at the top-level directory that a tenant can access within the Tenant Storage service. The Tenant Administrator can create or edit additional DataTaps that point at or below that directory.

If the tenant storage is based on a local HDFS, then the Platform Administrator can specify a storage quota for each tenant. The HDFS back-end is used to enforce this quota, meaning that the quota applies to storage operations that originate from either the DataTap browser or the nodes within that tenant.

Root tenant storage folders are created under the deployment global tenant storage root. For example, given a global tenant storage root of `/a/b`, the tenant-specific tenant storage root directories will be `/a/b/1` for Tenant 1 and `/a/b/2` for Tenant 2.

You may create DataTaps that point to any subdirectory within the global tenant storage root, so long as that location cannot access another tenant's tenant storage root directory, nor the global tenant storage root. For example:

- You could create a DataTap in Tenant 1 that points to `/a/b/SharedStorage`, because that directory is not part of any existing tenant's Tenant Storage root.
- You will also be able, as Tenant 2, to create another DataTap that points to `/a/b/SharedStorage`, thereby allowing data sharing between Tenant 1 and Tenant 2.

NOTE: Tenant 2 cannot create a DataTap to the `/a/b/1/SharedStorage` directory, because the `/a/b/1` directory is the root tenant storage directory for Tenant 1.

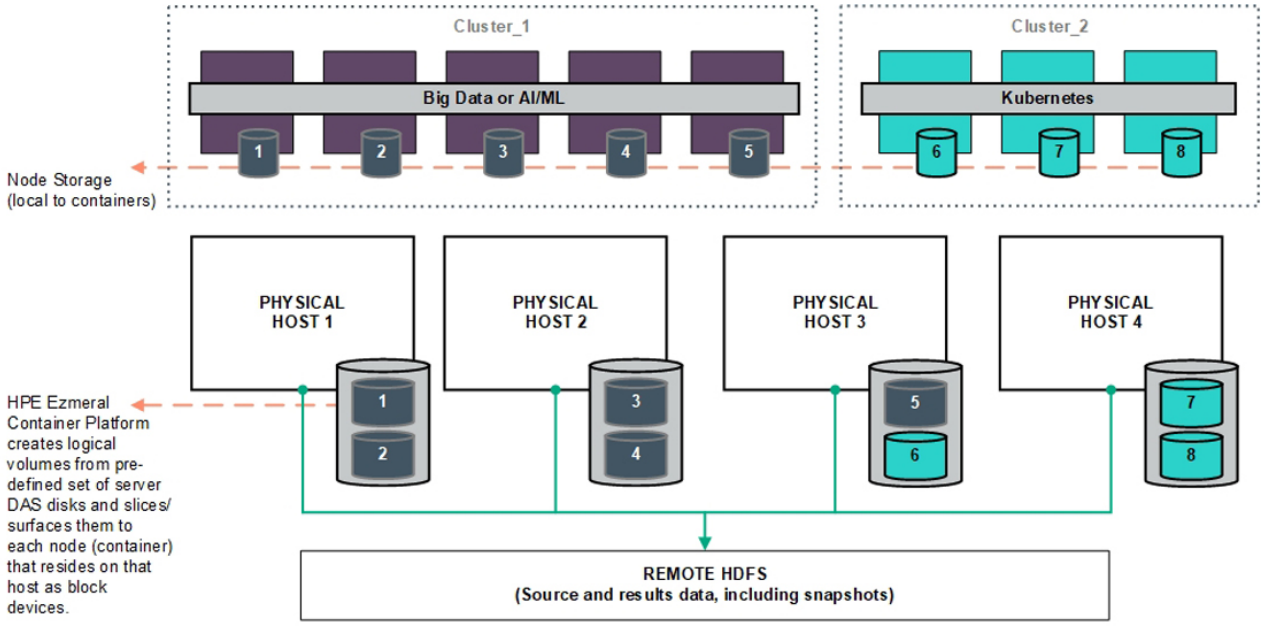
Users who have a Tenant Administrator role may view and modify detailed DataTap information. Members may only view general DataTap information and are unable to create, edit, or remove a DataTap.

NOTE: Data conflicts may occur if more than one DataTap points to a location being used by multiple jobs at once.

Node Storage

Node storage (referred to as *ephemeral storage* in Kubernetes clusters) is built from the local storage in each host and is used for the disk volumes that back the local storage for each virtual node. Using

SEDs (Self-Encrypting Drives) will ensure that any data written to node storage is encrypted on write and decrypted on read by the OS. A tenant can optionally be assigned a quota for how much storage the nodes in that tenant can consume.



Virtual nodes/containers running on public cloud VMs (such as AWS EC2) utilize storage within the instance (such as AWS Elastic Block Storage, or EBS) as node storage.

About DataTaps

DataTaps expand access to shared data by specifying a named path to a specified storage resource. Applications running within virtual clusters that can use the HDFS filesystem protocols can then access paths within that resource using that name, and DataTap implements Hadoop File System API. This allows you to run jobs using your existing data systems without the need to make time-consuming copies or transfers of your data. Tenant/Project Administrator users can quickly and easily build, edit, and remove DataTaps using the **DataTaps** screen, as described in The DataTaps Screen (Admin). Tenant Member users can access DataTaps by name.

Each DataTap requires the following properties to be configured, depending on the type of storage being connected to (MapR, HDFS, HDFS with Kerberos, or NFS):

- **Name:** A unique name for each DataTap. This name may contain letters (A-Z or a-z), digits (0-9), and hyphens (-), but may not contain spaces. You can use the name of a valid DataTap to compose DataTap URIs that you pass to applications as arguments. Each such URI maps to some path on the storage system that the DataTap points to. The path indicated by a URI might or might not exist at the time you start a job, depending on what the application wants to do with that path. Sometimes the path must indicate a directory or file that already exists, because the application intends to use it as input. Sometimes, the path must not currently exist, because the application expects to create it. The semantics of these paths are entirely application- dependent, and are identical to their behavior when running the application on a physical Hadoop or Spark platform.

- **Description:** Brief description of the DataTap, such as the type of data or the purpose of the DataTap.

- **Type:** Type of file system used by the shared storage resource associated with the DataTap (**MAPR**, **HDFS**, or **NFS**). This is completely transparent to the end job or other process using the DataTap.

The following fields depend on the DataTap type:

- MapR

- HDFS
- NFS on page 124
- GCS on page 125

MapR

NOTE: All of the links to MapR articles in this section will open in a new browser tab/window.

A MapR DataTap is configured as follows:

- **Cluster Name:** Name of the MapR cluster. See the MapR articles Creating the Cluster and Creating a Volume articles.
- **CLDB Hosts:** DNS name or address of the container location database of a MapR cluster. See the MapR article Viewing CLDB Information.
- **Port:** Port for the namenode service on the host used to access the MapR file system. See the MapR article Specifying Ports.
- **Mount Path:** Complete path to the directory containing the data within the specified MapR file system. You can leave this field blank if you intend the DataTap to point at the root of the MapR cluster. See the MapR articles Viewing Volume Details and Creating a Volume.
- **MapR Secure:** Checking this check box if MapR cluster is secured. When the MapR cluster is secured, all network connections require authentication, and moving data is protected with wire-level encryption. MapR allows applying direct security protection for data as it comes into and out of the platform without requiring an external security manager server or a particular security plug-in for each ecosystem component. The security semantics are applied automatically on data being retrieved or stored by any ecosystem component, application, or users. See the MapR article Security.
- **Ticket Source:** Select the ticket source. This will be one of the following:
  - **Upload Ticket File:** This is enabled when Ticket source is selected as **Use Existing File**.
  - **Use the existing one:** To use the existing ticket details.
- **Ticket file:** This will be one of the following:
  - When **Upload Ticket File** is selected, **Browse** button is enabled to select the tiket file.
  - When **Use the Existing One** is selected, it is the name of the existing ticket file.
- **Enable Impersonation:** When you enable impersonation, when a user signs into the container and creates a file in the MapR cluster through the DataTap connection, ownership of that file is assigned to that user. If the user does not exist in the MapR cluster, then the connection between the DataTap and the MapR cluster is rejected. Typically, administrators ensure that the same users exist in both the container and the MapR cluster by configuring both the container and the MapR cluster with the same AD/LDAP settings.
- **Select Ticket Type:** Select the ticket type. This will be one of the following:
  - **User:** Grants access to individual users with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service**: Accesses services running on client nodes with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.

last-updated: May 06, 2024

  - **Service (with impersonation):** Accesses services running on client nodes to run jobs on behalf of any user. The ticket cannot be used to impersonate the `root` or `mapr` users.
  - **Tenant:** Allows tenant users to access tenant volumes in a multi-tenant environment. The ticket can impersonate any user.
- **Ticket User:** Username to be included in the ticket for authentication.
- **MapR Tenant Volume:** Indicates whether or not the mount path is a MapR tenant volume. See the MapR article Setting Up a Tenant.
- **Enable Passthrough**: Select this box to enable Passthrough mode.

See the following examples for additional information:

- Sample MAPR DataTap - No Impersonation
- Sample MAPR DataTap - Impersonation

HDFS

An HDFS DataTap is configured as follows:

- **Host:** DNS name or IP address of the server providing access to the storage resource. For example, this could be the host running the namenode service of an HDFS cluster.
- **Standby NameNode:** DNS name or IP address of a standby namenode host that an HDFS DataTap will try to reach if it cannot contact the primary host. This field is optional; when used, it provides high-availability access to the specified HFDS DataTap.
- **Port:** For HDFS DataTaps, this is the port for the namenode server on the host used to access the HDFS file system.
- **Path:** Complete path to the directory containing the data within the specified HDFS file system. You can leave this field blank if you intend the DataTap to point at the root of the specified file system.
- **Kerberos parameters:** If the HDFS DataTap has Kerberos enabled, then you will need to specify additional parameters. HPE Ezmeral Runtime Enterprise supports two modes of user access/ authentication.
  - Proxy mode permits a “proxy user” to be configured to have access to the remote HDFS cluster. Individual users are granted access to the remote HDFS cluster by the proxy user configuration. Mixing and matching distributions is permitted between the compute Hadoop cluster and the remote HDFS.
  - Passthrough mode passes the credentials of the current user to the remote HDFS cluster for authentication.
- HDFS file systems configured with TDE encryption as well as cross-realm Kerberos authentication are supported. See HDFS DataTap TDE Configuration and HDFS DataTap Cross-Realm Kerberos Authentication for additional configuration instructions.

NFS

NOTE: This option is not available for Kubernetes tenants.

An NFS DataTap is configured as follows:

- **Host:** DNS name or IP address of the server providing access to the storage resource.

- **Share:**This is the exported share on the selected host.
- **Path:** Complete path to the directory containing the data within the specified NFS share. You can leave this field blank if you intend the DataTap to point at the root of the specified share.

GCS

An GCS DataTap is configured as follows:

- **Bucket Name:** Specify the bucket name for GCS.
- **Credential File Source:** This will be one of the following:
  - When **Upload Ticket File:** is selected, **Browse** button is enabled to select in the **Credential File**. The credential file is a JSON file that contains the service account key.
  - When **Use the Existing One:** is selected, enter the name of the previously uploaded credential file. The credetial file is a JSON file that contains the service account key.
- **Proxy:** This is optional. Specify http proxy to access GCS.
- **Mount Path:**Enter a path within the bucket that will serve as the starting pointfor the DataTap. If the path is not specified, the starting point will default to the bucket.

Using a DataTap

The storage pointed to by a DataTap can be accessed via a URI that includes the name of the DataTap.

A DataTap points to the top of the “path” configured for the given DataTap. The URI has the following form:

```
dtap://datatap_name/
```

In this example, `datatap_name` is the name of the DataTap that you wish to use. You can access files and directories further in the hierarchy by appending path components to the URI:

```
dtap://datatap_name/some_subdirectory/another_subdirectory/some_file
```

For example, the URI `dtap://mydatatapr/home/mydirectory` means that the data is located within the `/home/mydirectory` directory in the storage that the DataTap named `mydatatap` points to.

DataTaps exist on a per-tenant basis. This means that a DataTap created for Tenant A cannot be used by Tenant B. You may, however, create a DataTap for Tenant B with the exact same properties as its counterpart for Tenant A, thus allowing both tenants to access the same storage resource. Further, multiple jobs within a tenant may use a given DataTap simultaneously. While such sharing can be useful, be aware that the same cautions and restrictions apply to these use cases as for other types of shared storage: multiple jobs modifying files at the same location may lead to file access errors and/or unexpected job results.

Users who have a Tenant Administrator role can view and modify detailed DataTap information. Members can only view general DataTap information and are unable to create, edit, or remove a DataTap.

CAUTION: Data conflicts can occur if more than one DataTap points to a location being used by multiple jobs at once.

CAUTION: Editing or deleting a DataTap while it is being used by one or more running jobs can cause errors in the affected jobs.

More information

Troubleshooting DataTap Issues on page 944

FS Mounts

The filesystem mount feature allows the automatic addition of NFS v3 or v4 volumes or mounts to virtual nodes/containers. This allows virtual nodes/containers to directly access NFS shares as if they were local directories. You can use this feature to provide common files across all of the virtual nodes/containers of a given tenant, such as a common configuration file that will be used by each of the virtual nodes/containers in the Marketing tenant. This eliminates the need to manually copy common files to individual virtual nodes/containers.

All virtual nodes/containers include a root directory called `/bd-fs-mnt`. If one or more filesystems have been mounted, then this directory will contain the mounted filesystems. Each mounted filesystem in this directory will have the same name as the **Mount Name** that was assigned when creating the FS mount (see Creating a New FS Mount).

Filesystems are mounted on a per-tenant basis, meaning that a given filesystem mount will be applied to each of the virtual nodes/containers in the tenant where that filesystem was created. For example, if you create a filesystem mount in the Marketing tenant, then each of the virtual nodes/containers created in the Marketing tenant will include that filesystem mount. Tenant Administrator users can create, modify, and delete filesystem mounts. Tenant Member and Platform Administrator users may view filesystem mounts but cannot modify them.

A filesystem may be mounted as either:

- **Read Only:** Users can view (read) objects in the filesystem but cannot create, modify, or delete objects.
- **Read/Write:** Users can view, create, modify, and/or delete objects.

FSmount is backed by a POSIX-based filesystem, such as the HPE Ezmeral Data Fabric POSIX client or NFS server. When HPE Ezmeral Runtime Enterprise is configured with HPE Ezmeral Data Fabric storage as its tenant storage, then FSmount points to HPE Ezmeral Data Fabric POSIX clients by default.

Inside every container:

- When a new filesystem is mounted, the **Name** property will be populated in the `/bd-fs-mnt` directory.
- The contents of the NFS share will be accessible in either read only or read/write fashion, depending on the settings provided when creating the mount.
- Users will not be able to write files to or create new folders in `/bd-fs-mnt`.

See the following articles for additional information:

- The FS Mounts Screen
- Creating a New FS Mount
- Editing an Existing FS Mount
- Deleting an FS Mount

User Authentication

Each user has a unique username and password that must be provided in order to log in. Authentication is the process by which the user-supplied username and password are matched against the list of authorized users to determine whether to grant access (stored either in the local user database server or in the remote LDAP/Active Directory server). Authentication is the process that determines what exact access to allow, in terms of the specific roles granted to that user.

User authentication information is stored on a secure server. Users can be authenticated using any of the following methods:

- An internal user database.

last-updated: May 06, 2024

- One or more existing LDAP or Active Directory servers that you can connect to using Direct Bind or Search Bind. The following configurations are supported when using multiple LDAP/AD servers:
  - Servers located across multiple domains. In this case, a user may specify the domain to use when accessing the web interface, as described in Launching and Logging In.
  - Multiple servers in the same domain. This allows authentication to occur in a failover mode when one or more of the servers is down or otherwise unreachable.
  - Both of the above. In this case, multiple LDAP/AD domains can be used where one or more of those domains includes multiple servers set up to allow failover.

HPE Ezmeral Runtime Enterprise can also apply user authentication settings as follows:

- **Platform:** The same authentication settings apply to every tenant/project. HPE Ezmeral Runtime Enterprise may be configured to use local authentication or one or more LDAP/AD domains. Further, each domain may be configured to use multiple servers in order to provide failover protection.

Non-SSO Access

The non-SSO user authentication process is identical when using either the internal user database or an external LDAP/AD server:

1. A user accesses the **Login** screen using a Web browser.
   - If tenant-independent authentication is not enabled, the URL will be `http://<ip_address>`, where `<ip_address>` is either:
     - The IP address of the Controller host (if platform HA is not enabled).
     - The cluster IP address (if platform HA is enabled and you provide a cluster IP address).
     - The IP address of a Gateway host (if platform HA is enabled but no cluster IP address is provided). When using a private (non-routable) virtual node network, the Primary Controller and Shadow Controller need not be on the same subnet unless a Cluster IP address is specified.
     - Hostname of the Controller host.
     - A DNS-mapped URL.

     NOTE: Replace `http` with `https` if a secure connection is required.
2. The user enters her or his username and password in the appropriate fields and attempts to login. If multiple LDAP/AD domains are configured, then the user must either specify the domain to use via the **Domain for Authentication** pull-down menu or enter their username as `user@domain`. See Launching and Logging In.
3. The user-supplied username and password is securely to the authentication server, if TLS is enabled.
4. The authentication server returns a response that indicates either a valid (allow user to login) or invalid (prevent user from logging in) login attempt.
5. If the login attempt is valid, then the user will be matched with the roles granted to that user and allowed the proper access.

Using the internal user database is fast and convenient from an IT perspective. However, it may complicate user administration for various reasons, such as:

- The user may be required to change their password on the rest of the network but this change will not be reflected in HPE Ezmeral Runtime Enterprise.

- A user who is removed from the network (such as when they leave the organization) must be independently removed from the HPE Ezmeral Runtime Enterprise user database.

Connecting to your existing user authentication server requires you to supply some information about that server during installation. Contact your user administrator for the following information:

- **LDAP:** LDAP Host, User Attribute, User Subtree DN
- **Active Directory:** AD Host, User Attribute, User Subtree DN

SSO Access

Single Sign On (SSO) allows users to supply login credentials once, and then gain access to all authorized resources and applications without having to log in to each application separately. When SSO is configured, authorized users will proceed directly to the **Dashboard** screen without having to log in, by navigating to either of the following, as appropriate:

- `http://<ip_address>`, if tenant independent authentication is not enabled.

SSO configuration requires both of the following, which you specify in the **User Authentication** tab of the **System Settings** screen, or the **External Authentication** tab of the **Create Tenant** or the **Edit Tenant** screen. See Configuring User Authentication Settings:

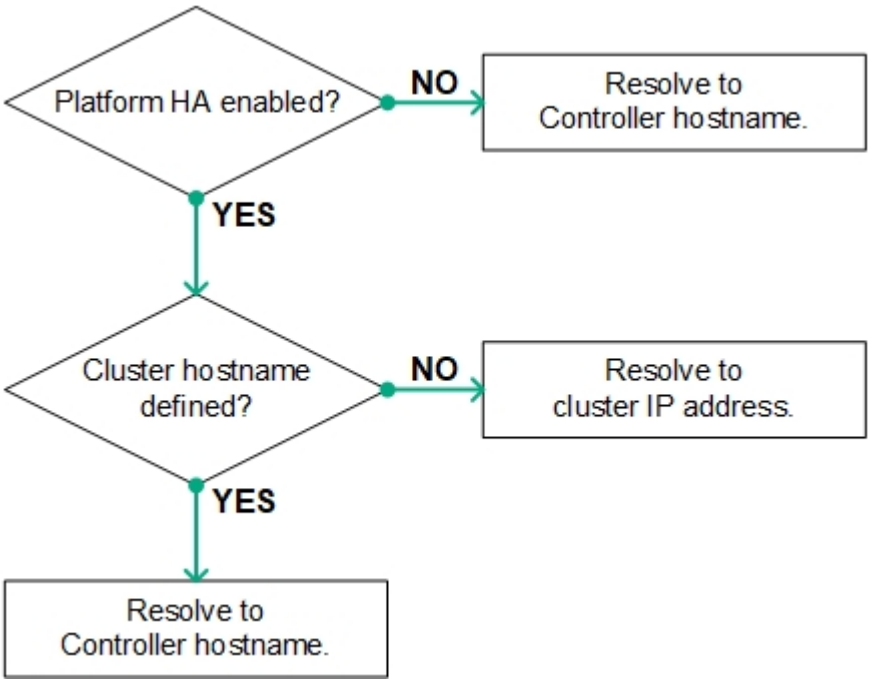
- A metadata XML file that is provided by the Identity Provider (IdP)
- Configuring the `XPath` parameter to refer to a location in the SAML Response that contains the LDAP/AD username of the authenticated user. This will commonly be `//saml:Subject/saml:NameID/text()`, but this value may be passed in the SAML Attributes as well. In that case, the XPath may be similar to the following:

  ```
  //saml:AttributeStatement/saml:Attribute[@Name="PersonImmutableID"]/
  saml:AttributeValue/text()
  ```

You can then use LDAP/AD groups to assign roles to users. See Assigning/Revoking User Roles (LDAP/AD). Groups can also be assigned to users from the SAML Assertion. In order to point to the group field in the SAML Assertion, fill in the **Group XPath** field. This value will probably be similar to `//saml:AttributeStatement/saml:Attribute[@Name="Groups"]/saml:AttributeValue/text()`. If all of the groups are included in a single XML node with a separator character, then the Group Separator can be used to specify the character that separates the groups. A user will not receive groups from two different sources; they will only receive groups from one of the following:

- The SAML Assertion
- The LDAP/AD server

If platform High Availability is not enabled, then the hostname of the Controller host must be mapped to the IP address of the Controller host via a DNS server that can be accessed by the user. This allows a user-initiated browser GET request to correctly resolve to the Controller host. For deployments where platform HA is enabled, this will be a hostname that resolves to the cluster IP address.



The IdP must be configured with the following information:

- **Audience:** This field is not required; however, providing the base URL of the SAML server is more secure than a blank entry. If you do enter a URL, then this URL must exactly match the SAML Application Name that you will specify in HPE Ezmeral Runtime Enterprise.

- **Recipient:** `[<hostname>|<ip_address>]/bdswebui/login`, where `<hostname>` is the name of the Controller host, the HA cluster, or the Controller gageway FDQN as appropriate, and `<ip_address>` is the Controller, cluster IP address, or Gateway host IP address.

  For HPE Ezmeral Runtime Enterprise 5.3.5 and later releases, to use SAML SSO with Jupyterhub Notebooks, you must specify the Controller gateway FDQN for `<hostname>`. Do not specify an IP address.

  Use either a hostname or an IP address, but not both. For example, `controllername/bdswebui/login` or `10.32.1.10/bdswebui/login`.

- **Consumer URL Validator:** Enter `<platform_info>/bdswebui/login/`, where `<platform_info>` is one of the following:

  - `.*` - This is a valid generic entry, but is less secure. For example, `.*/bdswebui/login/`.

  - `<name-or-ip>`, which will be either the FQDN or IP address of either the Controller host or HA cluster, as described above. This entry is more secure than the generic entry. For example, `10.32.0.75/bdswebui/login/` or `platform-01.organization.com/beswebui/login/`.

- **Consumer URL:** `<platform_info>/bdswebui/saml_login/`, where `<platform_info>` is either a generic or specific entry, as described above.

- Further, the IdP must send the user's LDAP/AD username in the body of the SAML assertion, either as the subject or as a field in the SAML attributes.

The IdP will provide a SAML IdP XML metadata file that you will use when configuring HPE Ezmeral Runtime Enterprise for SSO.

Users and Roles

Components of a User

A user consists of the following components:

- Login credentials (user name and password)
- One or more roles.

Number of Roles Per User

Users that are not Platform Administrators can have a maximum of one assigned role per tenant or HPE Ezmeral ML Ops project.

A user with more than one role may be a Member of some tenants and a Tenant Administrator of other tenants.

Platform Administrators can access all tenants and projects. While they are accessing a tenant or project, the Platform Administrator automatically assumes the role of Tenant Administrator or Project Administrator.

Planning Considerations

Some of the planning considerations related to users and tenants in a deployment of HPE Ezmeral Runtime Enterprise include the following:

Tenants	The number of tenants and the functions each tenant performs determines how many users with the Tenant Administrator role are needed and, by extension, the number of users with the Tenant Member role that are needed for each tenant. The reverse is also true, because the number and functions of users that need to run jobs can influence how you define tenants. For example, different levels of confidentiality might mandate separate tenants. See also Tenants and Projects on page 117.
Job functions	The specific work performed by a given user will directly impact the role they are assigned. For example, a small organization might designate a single user as the Tenant Administrator for multiple tenants, while a large organization might designate multiple Tenant Administrators per tenant.
Security clearances	You might need to restrict access to information based on the security clearance of a user. The need for this kind of restriction can impact both the tenants a user has access to and the role configured for that user within a given tenant.

Role-Based Access Control in Kubernetes Tenants

For detailed information about role-based access control within Kubernetes tenants, see Kubernetes Tenant RBAC on page 325.

Roles and Privileges

The privilege to perform an action is associated with one or more predefined user roles. Roles differ in the scope of the platform or tenant resources that they can affect.

last-updated: May 06, 2024

Tenant Members and Project Members

Tenant Members are users that have been assigned the `Member` role for a specific tenant.

In ML Ops contexts, tenants that are configured as HPE Ezmeral ML Ops projects are called **projects**, and users that are assigned the `Member` role are **Project Members**.

Tenant Administrators and Project Members:

- Operate within the tenant-specific or project-specific UI.
- Can view metrics in the tenant or project context.
- Can view, create, and delete workloads within the tenant or project.
- Can view and use DataTaps and FS Mounts. However, Members cannot view the detailed information about the connected storage services, and cannot create, edit, or delete DataTaps or FS Mounts.
- Have access to a kubectl configuration associated with tenant or project member privileges in the tenant or project namespace.

Tenant Administrators and Project Administrators

Tenant Administrators are users that have been assigned the `Admin` role for a specific tenant.

In ML Ops contexts, tenants that are configured as HPE Ezmeral ML Ops projects are called **projects**, and users that are assigned the `Admin` role are **Project Administrators**.

Tenant Administrators and Project Administrators:

- Operate within the tenant-specific UI.
- Have all the capabilities of Tenant Members or Project Members.
- For DataTaps and FS Mounts, can also view the connected storage service details, and can create, edit, and delete DataTaps and FS Mounts.
- Can assign and revoke tenant or project users.
- Have access to a kubectl configuration associated with tenant administrator privileges in the tenant or project namespace.

Kubernetes Cluster Administrator

Kubernetes Cluster Administrators are users that have been assigned the `K8S Admin` role for a specific cluster.

Kubernetes Cluster Administrators:

- Can view services status, usage totals, alerts, and metrics in the context of the Kubernetes cluster.
- Have access to the Kubernetes dashboard of the cluster.
- Can view detailed information about the hosts that are acting as Kubernetes cluster nodes.

	• Can view detailed information about the Kubernetes tenants or projects associated with the cluster. • Can assign and revoke users for those associated tenants or projects. • Have access to the administrative kubectl configuration for the cluster.
Platform Administrator	A user that has been assigned the `Site Admin` role is known as a Platform Administrator. This role is also called the **Kubernetes Administrator** in the context of managing Kubernetes hosts, clusters, tenants, and users. Platform Administrators: • Can operate as tenant or project administrator without needing an explicit role assignment. • Have all the capabilities of Kubernetes Cluster Administrators for each Kubernetes cluster. • Can view services status, usage totals, alerts, and metrics in a sitewide context. • Can create, edit, and delete tenants or projects. • Can add hosts to and remove hosts from the deployment. • Can create, edit, resize, delete, and upgrade Kubernetes clusters. • Can modify sitewide user authentication settings (for AD/LDAP group-based users) and manage local user accounts. • Can assign and revoke all user roles. • Can control other sitewide configuration, such as security policies, High Availability, gateways, licensing, air gap, and platform upgrades.

High Availability

High availability (HA) in deployments of HPE Ezmeral Runtime Enterprise is divided into platform controller HA, gateway HA, and cluster HA.

Different types of high availability (HA) protection are available:

- **Platform High Availability**. This protection applies to HPE Ezmeral Runtime Enterprise Controller and services.
- **Gateway Host High Availability**. This protection applies to the Gateway hosts.
- **Kubernetes Cluster High Availability** **on page 134**. This protection applies to all Kubernetes clusters.

Platform High Availability

Platform high availability protects against the failure of the Controller host. When Platform HA is enabled, three different hosts are used:

- The Controller host

- The Shadow Controller host
- The Arbiter host

Under normal circumstances:

- The Controller host manages HPE Ezmeral Runtime Enterprise.

If any of the three hosts fails, the following actions occur:

- Host-specific failure actions:

Controller host failure	If the Controller host fails, the Arbiter host switches management to the Shadow Controller host. This process usually takes two to three minutes. After a failover to the Shadow Controller host, the deployment continues to run, but in a degraded state. In that degraded state, there is no protection against the failure of the Shadow Controller host. During a failover, all user web sessions will be terminated. Users must sign in again after the failover process completes.
Shadow Controller host failure	If the Shadow Controller host fails but the Controller host is running, the deployment continues to run, but in a degraded state. In that degraded state, there is no protection against the failure of the Controller host.
Arbiter host failure	If the Arbiter host fails but the Controller host is running, the deployment continues to run, but in a degraded state. In that degraded state, there is no protection against a Controller host failure. Failover to the Shadow Controller cannot occur if the Arbiter host has failed.

- A message is displayed in the upper right corner of the web interface warning you that the deployment is running in a degraded state. If the Shadow Controller or Arbiter host fails, the message is displayed even if the Controller host is functioning properly.

  If SNMP/SMTP is configured, service alerts are sent.

  You can use the **Service Status** tab of the Platform Administrator **Dashboard** (see Dashboard - Platform Administrator on page 570) to see which host has failed and which services are down.

- HPE Ezmeral Runtime Enterprise analyzes the root cause of the host failure and attempts to recover the failed host automatically. If recovery is possible, the failed host comes back up, and normal operation resumes.

- If the problem cannot be resolved, the affected host is left in an error state.

  You must manually diagnose and repair the problem (if possible) and then reboot that host. If rebooting solves the problem, then the failed host will come back up, and normal operation will resume.
  If rebooting the host does not solve the problem, contact Hewlett Packard Enterprise Support for assistance.

Each host has its own IP address. If the Controller host fails, attempting to access the Shadow Controller host using the same IP address will fail. Similarly, accessing the Shadow Controller host using that host IP address will fail after the Controller host recovers. To avoid this problem, you must do one of the following:

- Access the web interface using one of the following:

  - If configured, you can use the host name of the Gateway host or Gateway set (if there are multiple Gateway hosts).
  - If configured when HA was enabled, you can use the cluster host name.
  - You can use the IP address, without a port number, of any Gateway host.
- Specify a cluster IP address that is bonded to the node acting as the Controller host, and then sign into the web interface using that cluster IP address. You will automatically connect to the Controller host (under normal circumstances) or to the Shadow Controller host with a warning message (if the Controller host has failed and triggered the High Availability protection). In this case, the Primary Controller and Shadow Controller hosts must be on the same subnet. You can access the web interface by using either the cluster IP address or a Gateway host IP address.

Gateway Host High Availability

You can add redundancy for Gateway hosts by mapping multiple Gateway host IP addresses to a single hostname. When this mapping is done, then either the DNS server or an external load balancer will load-balance requests to the hostname among each of the Gateway hosts on a round-robin basis. This configuration ensures that there is no single point of failure for the Gateway host. For more information, see The Gateway/Load Balancer Screen.

Kubernetes Cluster High Availability

You provide High Availability protection for a Kubernetes cluster by configuring a minimum of three hosts as the Kubernetes control plane (formerly called Kubernetes "masters"). You can specify additional control plane hosts, in odd number increments, for additional HA protection.

In Kubernetes, the state of the cluster is stored in a distributed key-value data store called **etcd**. Kubernetes clusters created by HPE Ezmeral Runtime Enterprise use kubeadm tools and a stacked controller etcd topology. In a stacked controller etcd topology, there is an instance of etcd in each control plane node.

Because of quorum requirements for etcd, two Kubernetes cluster control plane hosts are not sufficient. Although an even number of Kubernetes control plane nodes is supported for situations such as migrating a cluster, To maintain a quorum, HPE Ezmeral Runtime Enterprise recommends that you configure an odd number of Kubernetes control plane nodes.

Clusters with an even number of control plane nodes risk losing quorum permanently with a so-called "split brain." For example, consider a three-node Kubernetes control plane in which one node is down. The cluster can continue to operate because the quorum is two nodes. However, if you expand the control plane to four nodes, the quorum becomes three nodes. If you add the fourth node while one node remains down, and the addition of the fourth node fails because of an error, quorum is permanently lost: Your four-node control plane now has two nodes up and two nodes down, but requires a majority of three nodes to undo the failed membership change.

For more information about quorums, failure tolerance, and etcd clusters, see Failure Tolerance in the etcd documentation (link opens an external website in a new browser tab or window).

Public Key Infrastructure

A Public Key Infrastructure (PKI) is used to secure Remote Procedure Calls (RPC) between hosts. In this infrastructure:

- The Controller host knows which public server keys reside on each Worker host.

- Each Worker host knows which public keys can contact that host from the Controller host (or Shadow Controller, if platform HA is enabled).

This feature manifests itself in the following ways:

- **Adding a Worker using the Agent:** If you are adding a new Worker host using the agent as described in Agent-Based Kubernetes Host Installation, then you must copy the file `/opt/bluedata/keys/authorized_keys` from the Controller host to the same location on the new Worker host after installing the agent, and with the same owner/group, permissions, and SELinux context. See Kubernetes Worker Installation Overview. This is not needed for Gateway hosts. Copying the `authorized_keys` file is not necessary for Gateway hosts.

- **Non-agent based Worker installation**: `/opt/bluedata/keys/authorized_keys` will be securely transmitted to the Worker host using the credentials given for the Worker-add process. See Kubernetes Worker Installation Overview, Gateway Installation Tab. No manual action is needed for the keys.

NOTE: When PKI is used, the **Details** column of the **Installation** screen will include a **Fingerprint** column that displays an MD5 sum such as `f7:60:1f:45:fb:a7:e4:47:82:e2:38:19:a3:ff:08:bd` for each Worker host. This is the MD5 fingerprint contained in the file `/opt/bluedata/keys/ssh_host_rsa_key.pub` on the Worker host. This allows the Platform Administrator to confirm that they are adding the correct Worker host. You can verify this MD5 fingerprint by logging in to the Worker host and then executing the command `ssh-keygen -E md5 -lf /opt/bluedata/keys/ssh_host_rsa_key.pub`, followed by comparing the returned value to that displayed in the **Details** column.

CAUTION: Clicking **Install** means that you trust that you are installing HPE Ezmeral Runtime Enterprise on the correct, intended worker host.

Monitoring and Alerting

This article describes monitoring and alerting. Also see the following articles for additional information:

- Support Bundles Tab
- Config Checks Tab
- Troubleshooting Overview

Monitoring

Metricbeat collects data from the containers by running the container `stats` or other container commands. It also retrieves system-level information by reading `cgroup` data from the `OS/proc` files. Metricbeat then provides the collected metrics to Elasticsearch, where the data can be visualized on dashboards or through the Kibana dashboard.

NOTE: Kibana is only available for Kubernetes clusters running HPE Ezmeral Data Fabric. See HPE Ezmeral Data Fabric Introduction on page 578.

When platform-level HA is enabled (see High Availability), Elasticsearch will run on three hosts to ensure data replication and backup. Metricbeat is a lightweight service with minimal memory requirements.

The high-level workflow is as follows:

1. Metricbeat captures monitoring information and provides this data to Elasticsearch.
2. When platform HA is enabled, Elasticsearch replicates this data across the Controller, Shadow Controller, and Arbiter hosts.
3. Elasticsearch data can be visualized using either a **Dashboard** screen or through Kibana.

To access Kibana, see the following:

- If this is a Kubernetes deployment of HPE Ezmeral Runtime Enterprise, open the The Kubernetes Clusters Screen on page 457 screen. The **Details** column of the cluster contains a link to the Kibana service. Links to services are not shown when HPE Ezmeral Runtime Enterprise is in Lockdown mode.

  For default user name and password information for Kibana and Grafana on Data Fabric clusters, see Managing HPE Ezmeral Data Fabric on Kubernetes on page 627.

Alerting

Nagios runs as a container on the Controller host. The Nagios implementation is open source with no customization; however, a few Nagios scripts are included to monitor and provide alerts for some specific services. These scripts are located in the `/usr/lib64/nagios/plugins` directory.

There are two ways to configure Nagios alerts:

- **Web interface:** You may configure SNMP traps and SMTP email alerts through the web interface. See The Notification Settings Screen.

- **Within Nagios:** You may configure email alerts directly within Nagios, as described in Setting up Nagios Email Alerts on page 918.

Accessing HPE Ezmeral Runtime Enterprise Applications and Services

The articles in this section describe how to access the web interface and change your password. They also describe how to directly access virtual nodes/containers and other scenarios for accessing HPE Ezmeral Runtime Enterprise:

- **Enabling SSL Connections:** If you want to enable SSL connections after deploying HPE Ezmeral Runtime Enterprise, see Enabling SSL Connections. If you added an SSL certificate during the installation process, as described in Adding an SSL Certificate, this procedure is not needed and SSL connections are already enabled.

- **Launching and Signing In:** Accessing the web interface. See Launching and Signing In on page 136.

- **Changing Your Password:** How to change your password. See Changing Your Password.

- **Accessing Kubernetes Containers:** Specific instructions for accessing Kubernetes containers. See Accessing Kubernetes Containers.

- **API Access:** Describes API access. See API Access.

- **Updating External Service Passwords:** Some services running in a container may require additional authentication, meaning that you must authenticate with the service after logging in to the container. This article describes how to change the passwords for these services. See Updating External Service Passwords.

Launching and Signing In

The method used to launch and sign in to the web interface will vary slightly depending on the authentication configuration.

- **Platform Authentication:** If users are authenticated at the platform level, then see Platform Authentication on page 137.

- **SSO:** If Single Sign On (SSO) is enabled, then see Single Sign On (SAML SSO) on page 138.

In general:

- You cannot access the web interface from inside an IFRAME.
- Script injection and other common security loopholes are blocked.
- User access requires https:// access. Attempts to log in using http:// result in an error.

Platform Authentication

To launch and log into the web interface when platform authentication (non-SSO) is enabled:

1. In a Web browser, navigate to `https://<ip-address>`, where `<ip-address>` is one of the following:
   - The IP address of the Controller host.
   - The cluster IP address, which will automatically route you to either the Controller host (under normal circumstances) or the Shadow Controller host (if platform High Availability is enabled and the Controller host has failed). Please see Host Requirements on page 813 and High Availability on page 132 for information on enabling platform High Availability.
   - If HPE Ezmeral Runtime Enterprise is installed on a non-routable network with one or more Gateway hosts installed, then you may navigate to the IP address of any Gateway host without a port number to be automatically redirected to the Primary Controller or Shadow Controller, as appropriate. You may specify Port 80 for HTTP, 443 for HTTPS, or 8080 for RESTful API access. Adding a port number other than 80, 443, or 8080 to the IP address of a Gateway host will access the mapped service within one of the containers.

   Alternatively, if you have a DNS service on the network that maps the Controller IP address to the Controller hostname or cluster FQDN, then you can navigate to `https://<hostname>`, as appropriate.

   The **Sign In** screen appears.
2. Enter your username and password in the appropriate fields. If multiple authentication domains are configured, then you may either enter your username as `<username>@<domain>` (where `<username>` is your username, and `<domain>` is the name of the domain to use to authenticate your login), or simply enter your username and then proceed to Step 3.
   - The default Platform Administrator credentials are: `admin/admin123`.

   NOTE: You must have at least one role assigned in one tenant or project in order to be able to log in to the web interface.
3. If HPE Ezmeral Runtime Enterprise is configured for either local authentication or local authentication and a single LDAP/AD login, then skip to Step 5, otherwise proceed to Step 4.
4. If multiple authentication domains are configured, then you may use the **Domain for Authentication** menu to select the domain to use to authenticate your login. This menu does not appear if either local authentication or a single authentication domain has been configured, or if the Platform Administrator has disabled it, as described in Configuring User Authentication Settings on page 778.
   - You may do this instead of entering your username as `<username>@<domain>`, as described in Step 2.
   - If you entered your username as `<username>@<domain>` in Step 2, then that entry will override any selection you make in the **Domain for Authentication** menu.
     - If multiple domains are configured and the Platform Administrator has disabled the **Domain for Authentication** pull-down menu, then you may simply enter your username to search all available authentication domains.

5. Click the **Sign In** button.

Yo will be signed in to the tenant or project you last accessed before signing out of your previous session, and the **Dashboard** screen appropriate to the role you have in that tenant or project will appear. The content of the main menu also varies depending on your role. See Navigating the GUI on page 143.

You may switch to any tenant or project that you have access to by clicking the **User Actions** icon (down arrow) to the right of the **Role** display at the top of the screen to open the **User Actions** menu, and then selecting the desired tenant or project.

Single Sign On (SAML SSO)

To launch and log in when SAML SSO is enabled, launch a web browser and then navigate to one of the following, where `<ip-address>` is the IP address of the controller:

- `http://<ip-address>/bdswebui/login`
- `https://<ip-address>/bdswebui/login`

This action bypasses the **Sign In** screen.

You will be signed in to the tenant or project you last accessed before signing out of your previous session, and the **Dashboard** screen appropriate to the role you have in that tenant or project will appear. If you are a Kubernetes Cluster Administrator or a Platform Administrator, you will be signed into the deployment instead of a particular tenant or project. The content of the main menu also varies depending on your role. See Navigating the GUI on page 143.

You may switch to any tenant or project that you have access to by clicking the **User Actions** icon (down arrow) to the right of the **Role** display at the top of the screen to open the **User Actions** menu, and then selecting the desired tenant or project.

Related reference

Navigating the GUI on page 143
Describes the screen layout of the HPE Ezmeral Runtime Enterprise graphical user interface (GUI).

HPE Ezmeral Runtime Enterprise new UI on page 146
Introduces the HPE Ezmeral Runtime Enterprise UI that is the primary interface used to access machine learning (ML Ops) projects, and tenants that use analytics applications, such as Spark.

Users and Roles on page 130

Changing Your Password

NOTE: This article only applies to local user authentication. If your organization uses LDAP/AD or SSO, then please follow the appropriate procedures for your organization.

Clicking the **User Actions** icon (down arrow) to the right of the **Role** display at the top of the screen to open the **User Actions** menu, and then selecting the desired **Change Password** opens the **Update User Password** popup.

To change your password:

1. Enter your current (old) password in the **Current Password** field.
2. Enter your new password in the **New Password** field. Passwords are case-sensitive.
3. Confirm your new password in the **Confirm Password** field.

When you have finished entering your new password, click **Submit** to save your changes or **Cancel** to clear your changes without changing your password.

Accessing Kubernetes Containers

You can use Kubectl or SSHD to access Kubernetes containers in HPE Ezmeral Runtime Enterprise deployments.

There are two ways to access Kubernetes containers:

- SSHD
- Kubectl

SSHD

If the Kubernetes container is running the SSHD service, then you may use an SSH client to log in normally. The following considerations apply to this method:

- Automatic LDAP/AD integration is not provided for Kubernetes containers. To SSH into a container, you must therefore know of a valid user account within that container and know the login password for that account.
- The port for the SSHD service must be exposed through a Kubernetes service. For access from outside the Kubernetes cluster, this should be a NodePort service that is mapped to a port on a Gateway host. You can then use an external SSH client to connect to that port on the Gateway host.

Kubectl

Container access via the `kubectl` plugin is available via either an LDAP/AD directory server or via SAML. To authenticate with the plugin:

1. Verify that the computer to you are using to access the container is able to access the requisite ports described in Port Requirements and Kubernetes Port Requirements.
2. Verify HTTP access to the Controller host by executing the following command:

```
curl -k https://<gateway_ip_address>:8080/config
```

3. Verify HTTP access to the ports required for the authenticating proxy by executing the following command:

```
curl -k https://<gateway_ip_address>:9500/api\?timeout\=32s
```

4. Verify that both `kubectl` and `kubectl-hpecp` are installed on the computer you are using to access the container, and they are both on the PATH of your computer. You may download both plugins from a web interface Kubernetes **Dashboard** screen. See any of the following, as appropriate:
   - Dashboard - Kubernetes Tenant Member
   - Dashboard - Kubernetes Tenant/Project Administrator
   - Dashboard - Kubernetes Cluster Administrator
   - Dashboard - Kubernetes Administrator
5. Execute the following command, being sure to add the `--insecure` flag if the API is not protected by TLS.

```
kubectl hpecp refresh <gateway_ip_address>
```

6. When prompted, authenticate to the platform as instructed.
7. If prompted, select the tenant or cluster that the current context should be in.
8. Follow all printed instructions.

You may use the `kubectl exec` command to execute commands or open a shell inside the container.

- `kubectl` must be set up to access the Kubernetes cluster with privileges that include the `create` verb on the `pods/exec` resource. A Kubernetes Cluster Administrator will typically have this privilege, but other users typically will not, unless they get that privilege allowed for a restricted list of pods. See Kubernetes Tenant RBAC.
- For more details about using `kubectl exec`, see standard documentation such as https://kubernetes.io/docs/tasks/debug-application-cluster/get-shell-running-container/ (link opens an external website in a new browser tab/window).

API Access

This topic describes how to access the HPE Ezmeral Runtime Enterprise REST API and the REST API documentaton.

To access the REST API documentation for HPE Ezmeral Runtime Enterprise, on your controller host server, see:

`https://<controller-ip-address>:8080/apidocs`

To access the API, when platform authentication is configured, send the following POST request:

```
POST <http_or_https>://<ip_address_or_hostname>:<port>/api/v2/session
{
"name": "<username>",
"password": "<password>"
}
```

Where:

- `<http_or_https>`

  is the URL prefix, which will be either `http` or `https`.

  Hewlett Packard Enterprise strongly recommends using https to enhance organizational security.

- `<ip_address_or_hostname>` is either the IP address or hostname of the virtual node you are logging in to.
- `<port>` is the port number you are accessing on the virtual node.
- `<username>` is a valid username.
- `<password>` is the password for the specified username (case sensitive).
- `<tenant_key>` is the 10-character tenant key (case sensitive).

The current version of the API is v2. Every v2 endpoint begins with: `/api/v2/`

For information about the legacy v1 version of the API, see Legacy v1 API Documentation (link opens an external website in a new browser tab or window). The v1 version of the API is not recommended for new applications.

> **TIP:**
>
> If you are interested in developing software, connect to the HPE Developer Community to build, communicate, and collaborate. One of the many resources available through the community is the Hack Shack, which features on-demand workshops, such as the Introduction to the HPE Ezmeral Container Platform REST API.

Updating External Service Passwords

This article describes how to change the default passwords for the following included external services:

- Nagios
- HAProxy
- HACluster

Nagios

To change the Nagios password:

1. On the Controller host, execute the command `docker ps` to view the Nagios container.

   This command returns a table with a list of the containers running on the Controller host.

2. Look in the `NAMES` column for an entry similar to the following:

   ```
   epic-nagios-10.32.1.112
   ```

   This is the Nagios container running on the Controller.

3. Execute the following commands to view the current password:

```
# docker exec -it epic-nagios-10.32.1.112 bash
        # cat /etc/nagios/passwd
```

   The system will return the current administrator password:

```
nagiosadmin:nagiosadmin
```

4. To update the password, execute the following command:

```
# htpasswd -c /etc/nagios/passwd nagiosadmin
```

5. You will be prompted to `Enter the New password` and then to `Re-type new password for confirmation`.

   The system confirms that the password is being changed.

```
Adding password for user nagiosadmin
```

6. Execute the following command to verify that the password was changed:

```
# cat /etc/nagios/passwd
```

   The system displays the new administrator password:

```
nagiosadmin:$apr1$/5sis9Al$3ncyFom6EUXRnfymJf9Yo
```

7. Validate that the system has changed the password by accessing the Nagios interface and then typing `<controller_ip>:8443 (e.g. 10.32.1.112:8443)`.

   The system asks for a username and password.

8. Enter `nagiosadmin` as the username, and then enter the new password.

9. Verify successful login.

HAProxy

To change the HAProxy password, use the following procedure on each Gateway host:

1. Open the file `/opt/bluedata/common-install/scripts/haproxy/haproxy_globals.cfg` for editing.

2. In the `listen stats : 8081` section, find the following line:

```
stats auth haproxy:haproxy
```

3. Change this line to reflect the new password. For example, to change the password from `haproxy` to `haproxy1`, edit the line to read `haproxy:haproxy1`.

4. Execute the command `bds-controller restart` on the Gateway host.

5. After few minutes, access the HAProxy service on the Gateway host by navigating to `<gateway_ip>:8081`, and then attempt to log in as user `haproxy` with the new password.

HACluster

If you have forgotten the current HACluster password or you want to change the HACluster password, reset the HACluster password and re-authenticate the cluster nodes:

1. On the primary Controller, execute the following command:

```
passwd hacluster
```

2. If the controller host is running RHEL or CentOS, on the Shadow Controller, execute the following command:

```
passwd hacluster
```

3. On the primary Controller, re-authenticate the HACluster nodes by executing the following command:

```
pcs cluster auth <node1> <node2> --force
```

   where `<node1>` and `<node2>` are the hostnames or IP addresses of the nodes being re-authenticated.

Navigating the GUI

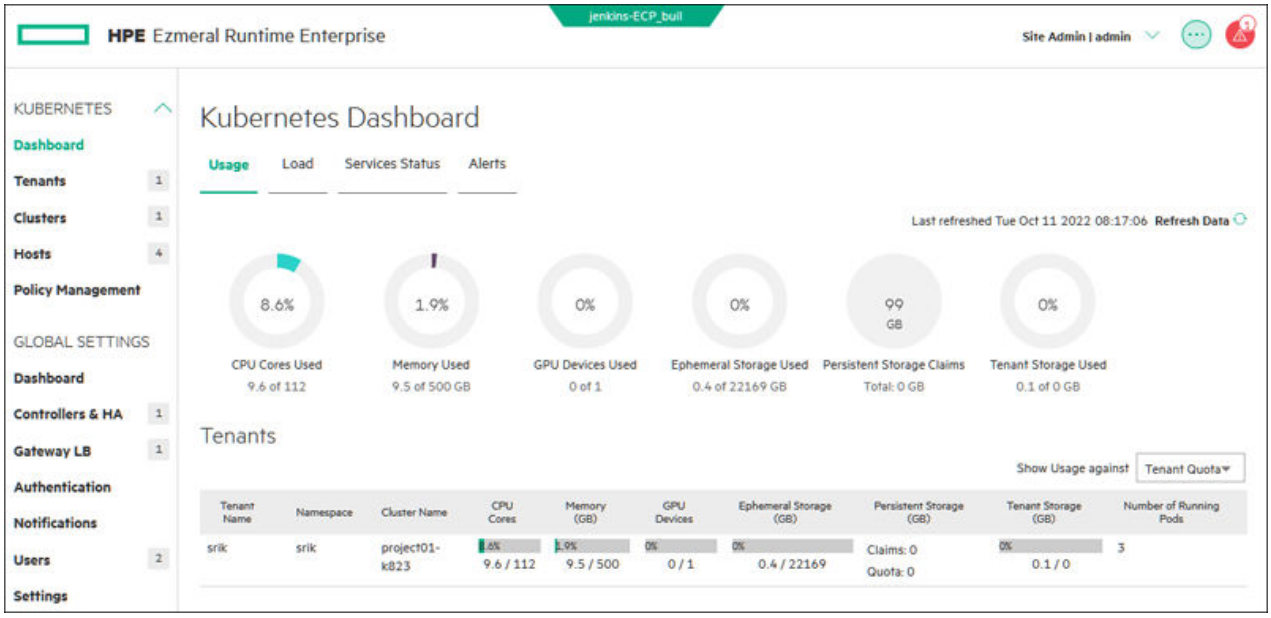
Describes the screen layout of the HPE Ezmeral Runtime Enterprise graphical user interface (GUI).

Graphical User Interface Orientation

The content of the Graphical User Interface (GUI)—also referred to as the web interface—varies according to factors such as the following:

- The product licenses that are active in the deployment
- The applications that are deployed
- The access rights and roles assigned to user that is signed into HPE Ezmeral Runtime Enterprise

The following image illustrates a typical layout of the interface for a user that is a Platform Administrator.



Toolbar

The toolbar is also called the application header.

Title	The application header displays the title: HPE Ezmeral Runtime Enterprise
Custom installation name	If a custom installation name was provided during installation, the name is displayed between the title and the other menus.
User menu	The user menu displays information about your user account: • If you can access only one tenant or project, your role in the current project is displayed. • If you can access multiple tenants or projects, both the current tenant or project name and your current role in that tenant or project is displayed. From the user menu, you can select the following: **Change Password** — Opens the **Change Password** dialog, which allows you to modify your password. This option does not appear if an external LDAP/AD server is being used to authenticate you. **Logout** — To sign out of HPE Ezmeral Runtime Enterprise, select **Logout**.
⋯ **Quick Access menu**	Click ⋯ to display the **Quick Access** menu. The items in the menu vary according to the user role and type of tenant. For a list of commonly included items, see Quick Access Menu - Common Items on page 145

Alerts Icon

If one or more alert conditions exist, then the **Alerts** icon appears on the right side of the **Toolbar**, along with the number of current alerts. Clicking this icon opens the **Alerts** dialog, which displays the current alerts.

For example:

Quick Access Menu - Common Items

The following items in the **Quick Access** menu are common to most users and tenants or projects:

User Info	Opens the **Current User Information** dialog, which lists your role, current project, and username.
User Guide	Opens this *User and Administrator Guide*.
Privacy	Opens the **Hewlett Packard Enterprise Privacy Statement** web page in a new browser tab or window.
Version	Displays version and build information about the HPE Ezmeral Runtime Enterprise deployment.
Ezmeral Runtime Enterprise New UI	Opens the home page of the HPE Ezmeral Runtime Enterprise new UI in a new browser tab or window. The interface that is displayed is the primary interface you use to access machine learning (ML Ops) projects, and analytics applications, such as Spark. This item appears only when you access a Kubernetes tenant or ML Ops project.

Main Menu

The main menu is also called the navigation sidebar. The entries in this sidebar vary according to the user role and the type of tenant.

For example:

- The **ML Workbench** menu item appears in the main menu only in tenants that are HPE Ezmeral ML Ops projects. Users click this item to access the project page of the HPE Ezmeral Runtime Enterprise new UI.
- The main menu and quick access menu contents related to managing the HPE Ezmeral Runtime Enterprise deployment appear when the the user is signed in as a **Platform Admistrator**.
- **Tenant Administrators** and **Project Administrators** have access to items that are not available to Tenant Members or Project Members.

Work Area

This area is where the screens are displayed. Each screen has a title. Screens might have tabs or pages. Actions that apply to the screen or page appear in the upper right area of the screen. SeeUsing the Work Area.

Related reference

HPE Ezmeral Runtime Enterprise new UI on page 146
Introduces the HPE Ezmeral Runtime Enterprise UI that is the primary interface used to access machine learning (ML Ops) projects, and tenants that use analytics applications, such as Spark.

Users and Roles on page 130

More information

Launching and Signing In on page 136

Using the Work Area

The work area is where HPE Ezmeral Runtime Enterprise displays each web interface screen.

Various generic functions will be available in the work area, depending on the screen you are accessing. These generic functions might include some or all of the following:

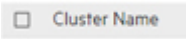
- Use the **Rows** menu to select how many records you want to see displayed on a single screen.

- Clicking a check box in a table selects that item. You may select one or more items and then perform an action on the selected items.

- Clicking the check box in a table header selects all of the items in that table.

- Clicking the arrows in a table column sorts the table by the information in that column. For example, clicking the arrows in the **Login Name** column of the **User Management** screen sorts the list of users by their login names. Repeatedly clicking a column header toggles the display between ascending (A-Z) and descending (Z-A) order.

- Clicking the **Search** icon and then entering one or more keywords in the field returns all records containing the supplied keywords in real time as you type; the work area refreshes as you type.

- If a screen contains too many records to display on a single page, you may use the page numbers and arrows to move between pages.

  - Clicking a page number opens the selected page of the current screen.
  - Clicking **Previous** button takes you to the previous page of the current screen.
  - Clicking the **Next** button takes you to the next page of the current screen.

HPE Ezmeral Runtime Enterprise new UI

Introduces the HPE Ezmeral Runtime Enterprise UI that is the primary interface used to access machine learning (ML Ops) projects, and tenants that use analytics applications, such as Spark.

Accessing the HPE Ezmeral Runtime Enterprise new UI

The HPE Ezmeral Runtime Enterprise new UI is the primary interface you use to access machine learning (ML Ops) projects, and tenants that use analytics applications such as Spark. The UI is distinct from the administrative UI that is displayed when you sign in.

From the administrative UI, you can access the HPE Ezmeral Runtime Enterprise new UI in one of the following ways:

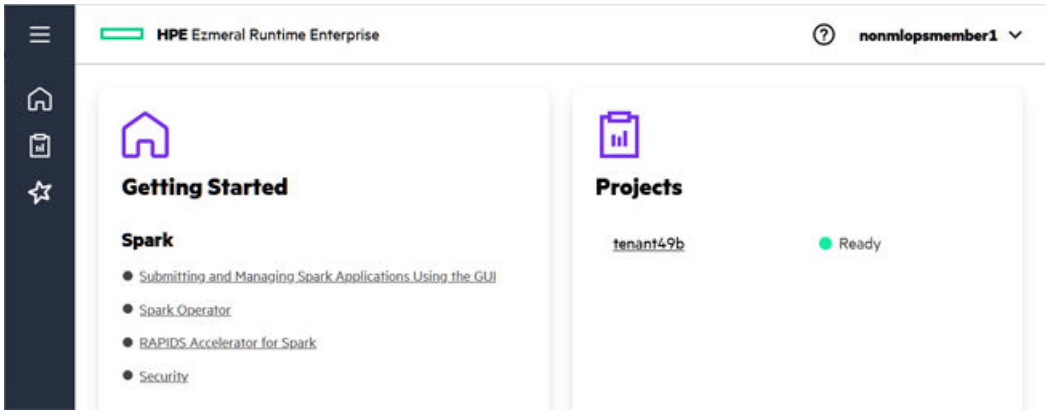
- If you have signed into an ML Ops project, in the main menu, click **ML Workbench**. This link opens the HPE Ezmeral Runtime Enterprise new UI in a new browser tab or window. The HPE Ezmeral Runtime Enterprise new UI displays the **Overview** tab of **Project Details** screen of the project.
- If you have signed into either an ML Ops project or a Kubernetes tenant, open the **Quick Access** menu and select **Ezmeral Runtime Enterprise New UI**. This link opens the home page of the HPE Ezmeral Runtime Enterprise new UI in a new browser tab or window.

Orientation

The contents of the interface varies according to factors such as the following:

- The applications that are deployed
- The access rights and roles assigned to user that is signed into HPE Ezmeral Runtime Enterprise

The following image illustrates the typical home page of the HPE Ezmeral Runtime Enterprise new UI.



Application Header

The application header is also called the toolbar.

Title	The application header displays the title: HPE Ezmeral Runtime Enterprise
Help Icon	Opens this *User and Administrator Guide*.
User menu	The user menu displays information about your user account. From the user menu, you can select the following: **Dark Mode** — Changes the interface to use a dark background. **Light Mode** — Changes the interface to use a light background.

last-updated: May 06, 2024

Sign Out	Signs you out of HPE Ezmeral Runtime Enterprise.

Main Menu

The main menu is also called the navigation sidebar. The items in this sidebar can include the following:

Home	Opens the home screen of the HPE Ezmeral Runtime Enterprise new UI.
Projects	Opens the **Projects** screen, which lists the ML Ops projects and Kubernetes tenant projects to which you have access. To open a specific project, click on the link for that project.
Spark	Opens the **Spark Applications** screen, from which you can view, manage, and create Spark applications.

Work Area

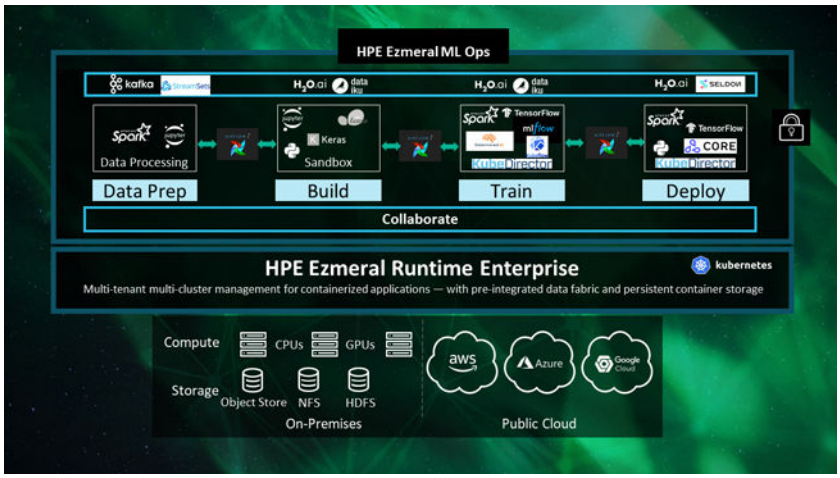
This area is where the screens are displayed. Each screen has a title. Screens might have tabs or tiles. See Using the Work Area.

HPE Ezmeral ML Ops

The topics in this section provide information about machine learning operations (ML Ops/MLOps) using HPE Ezmeral ML Ops in HPE Ezmeral Runtime Enterprise. (Not available with HPE Ezmeral Runtime Enterprise Essentials.)

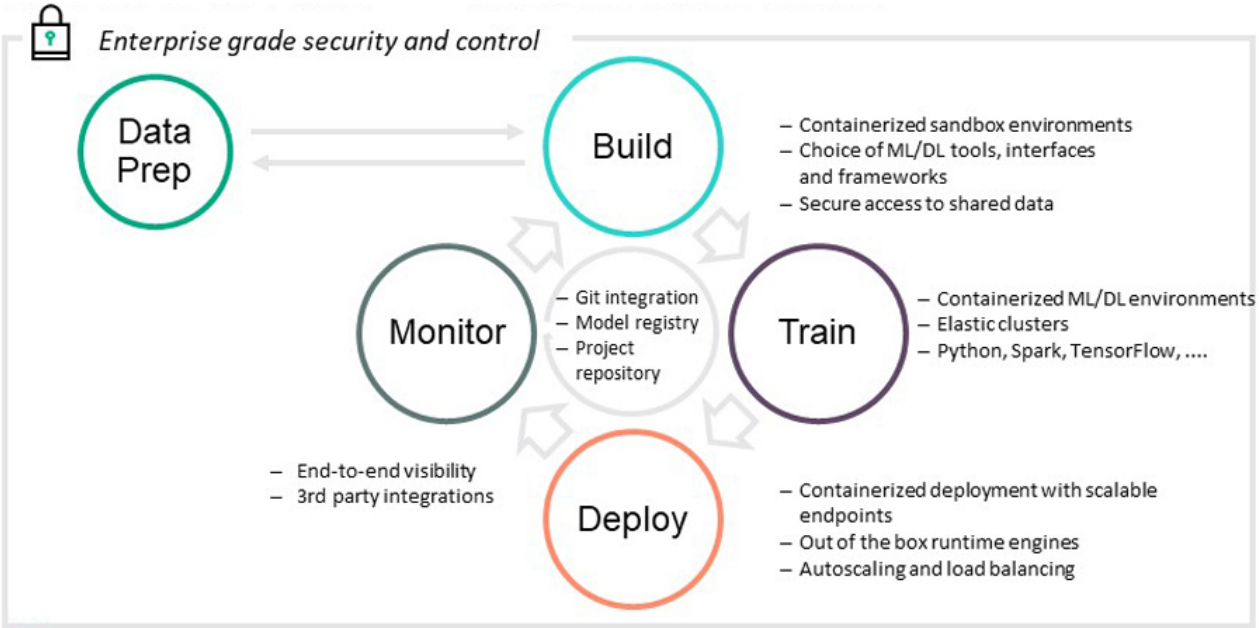
About HPE Ezmeral ML Ops

HPE Ezmeral ML Ops brings the power of Kubernetes pods and containers to the entire machine learning lifecycle to allow you to build, train, deploy, and monitor machine learning (ML) and deep learning (DL) models. HPE Ezmeral ML Ops supports sandbox development (notebooks), distributed training, and the deployment and monitoring of trained models in production. Project repository, source control, and model registry features allow seamless collaboration.



Features by ML Lifecycle Stage

With HPE Ezmeral ML Ops, data scientists can spin up containerized environments on scalable compute clusters with their choice of machine learning tools and frameworks for model development. When the model is ready for deployment, containerized endpoints with automatic scaling and load balancing are provided to handle variable workloads and optimize resource usage.



Some of the specific features supplied at each stage of the machine learning lifecycle include:

- **Build:**
  - Containerized sandbox environments
  - Choice of ML/DL tools, interfaces, and frameworks
  - Secure access to shared data
- **Train:**
  - Containerized, distributed ML/DL environments
  - Auto-scale capabilities
  - Prepackaged images for Python, Spark, and TensorFlow
- **Collaborate:**
  - Project Repository
  - Model Registry
  - Github integration
- **Deploy:**
  - Support for multiple runtime engines
  - REST endpoints with token-based authorization
  - Auto-scaling and load balancing

last-updated: May 06, 2024

- **Monitor:**
  - Notebook resource utilization
  - Training cluster resource monitoring
  - Deployment resource monitoring
  - REST input and output logs

Licensing

HPE Ezmeral ML Ops requires a separate license. See HPE Ezmeral ML Ops on page 95.

AI and ML Project Workflow

This topic describes getting started with the AI and ML workflows in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

The AI/ML workflow enables you to build, train, and deploy a model, and then send API requests to that model in order to make predictions. This workflow consists of the following high-level steps, which users must perform in the following order in accordance with their roles:

- Kubernetes Administrator
- LDAP/AD Administrator (For Jupyter Notebook KDapp Use)
- Project Administrator
- Project Member (Data Scientist)

You can then make predictions, as described in Making Prediction Calls With Deployed Models on page 186.

Kubernetes Administrator

1. Verify that the Platform Administrator has done the following:
   - Verified that HPE Ezmeral Runtime Enterprise is licensed for at least the number of CPU cores that will be used for the new Kubernetes cluster.
   - Configured LDAP/AD authentication.

     LDAP must be configured in order to run HPE Ezmeral ML Ops in a Kubernetes cluster. All AI/ML project users (Project Members and Project Administrators) must be LDAP/AD users. They cannot be authenticated using local authentication.
   - Configured and registered tenant storage on the HPE Ezmeral Runtime Enterprise deployment.
2. Log into the web interface as a Kubernetes Administrator, as described in Launching and Signing In on page 136.
3. Create a Kubernetes cluster, as described in Creating a New Kubernetes Cluster on page 463.

   IMPORTANT:

   Be sure to provide LDAP server information in the **Step 3: Authentication** screen; LDAP must be configured in order to run HPE Ezmeral ML Ops in a Kubernetes cluster.

4. Assign at least one user to be a Kubernetes Administrator for the Kubernetes cluster you just created. See Managing Kubernetes Admin Users on page 489 (to assign a user role using local authentication) or Updating External Kubernetes Cluster Admin Groups on page 490 (to assign a user role using LDAP/AD groups).

5. Note the hostname or IP address of the Kubernetes control plane hosts. Control plane hosts have the role `master` in the **Host(s) Info** tab of the Kubernetes **Cluster Details** screen (see The Kubernetes Cluster Details Screen on page 437).

6. Create a new Kubernetes AI/ML project, as described in Creating a New Kubernetes Tenant or Project on page 452. Ensure that you do the following:
   - Check the **AI/ML Project** check box.
   - Enter the external LDAP/AD user group in the External Authentication tab (see Kubernetes Tenant/Project External Authentication on page 456).

7. Assign at least one user to be a Kubernetes Project Administrator for the project you just created. See Viewing and Assigning Kubernetes Cluster Users on page 436.

LDAP/AD Administrator (For Jupyter Notebook KDapp Use)

If the environment will include the ability to use the Jupyter Notebook KubeDirector application (kdapp), LDAP server group settings must be changed for all members of the group.

The LDAP/AD Administrator must add member user IDs to user groups manually:

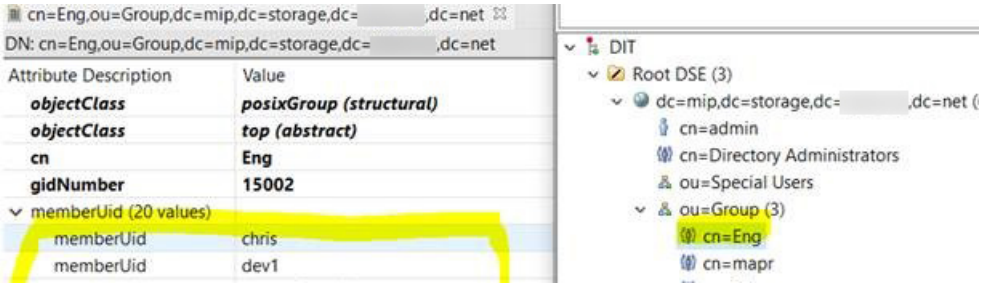
1. Connect to the LDAP server.

2. Access the Groups.

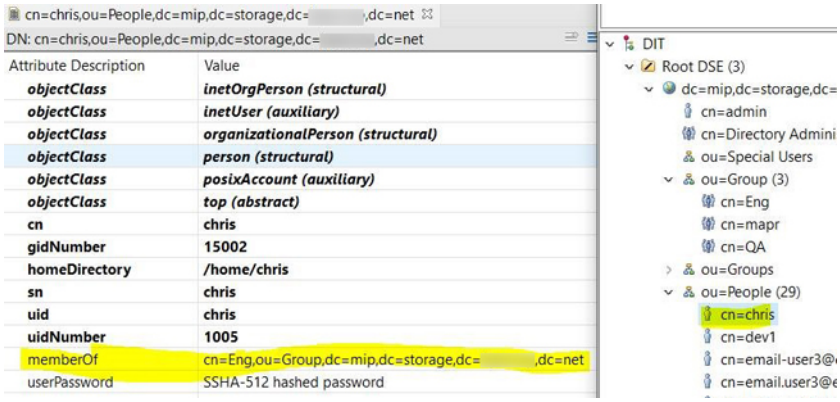
3. For each group that has members that will log in to a Jupyter notebook, do the following:

   a. For each member, create a `memberUid` attribute that has a value of the member's user ID.

   The following example shows the entry for the `Eng` group after members have been added.



4. You can verify which groups a member belongs to by selecting the entry for the member in `People`. For example:



Kubernetes Project Administrator

1. Confirm that the Kubernetes Administrator has completed all of the steps described in Kubernetes Administrator on page 150, above.
2. Create a data source. See Adding Data Sources on page 161.
3. If needed, create a new LDAP/AD user who will be assigned a role in the new project. If you create new users, The LDAP/AD Administrator might need to perform additional tasks (see LDAP/AD Administrator (For Jupyter Notebook KDapp Use) on page 151).

   NOTE: All AI/ML project users (Project Members and Project Administrators) must be LDAP/AD users. They cannot be authenticated using local authentication.

4. Assign at least one user to the new project as described in Assigning/Revoking User Roles (LDAP/AD/SAML) on page 774.
5. Configure one or more source control configuration templates, as described in Creating Source Control Configurations on page 164.

Kubernetes Project Member

Follow the instructions described in ML Ops Tasks on page 160 to perform the following:

- Create a source control configuration template.
- Create a notebook server.
- After running experiments in your notebook, view experiment results.
- Register and deploy models.
- Make prediciton calls with deployed models.

Installing HPE Ezmeral ML Ops

This topic describes how to install HPE Ezmeral ML Ops on Kubernetes clusters in HPE Ezmeral Runtime Enterprise.

Installing Shared RDBMS

This topic describes how to install the Shared RDMBS feature for HPE Ezmeral Runtime Enterprise. The Shared RDBMS feature is a common backend database service for application components across Kubernetes compute clusters.

Prerequisites:

- **Required access rights:** Kubernetes Administrator
- **Kubernetes cluster software requirements**:
  - Host OS is a minimum of RHEL 8 or SUSE 15 (SP2 or SP3)
  - Storage Class is configured (Data Fabric or any other CSI)
  - Helm 3

About Shared RDBMS

IMPORTANT: Install the Shared RDBMS feature (MySQL CE Operator 8.0.30) only **one time** per HPE Ezmeral Runtime Enterprise installation.

The Shared RDBMS feature provides common storage for ML components to manage metadata, and is accessible across Kubernetes clusters within HPE Ezmeral Runtime Enterprise.

The following HPE Ezmeral Runtime Enterprise components use the Shared RDBMS feature:

- Secure Model Management
- EzSQL

Installing Shared RDBMS

1. On a Kubernetes cluster master node, download the MySQL application bundle:

   ```
   # wget https://ezml-release.s3.amazonaws.com/5.6.0/
   mysql-cluster-ere560.tar.gz

   # tar  xvzf mysql-cluster.tar.gz && cd mysql-cluster
   ```

2. **(Optional)**: Before installing the Shared RDBMS feature, you can customize parameters for your InnoDB cluster. You can perform customization based on factors such as size of Kubernetes cluster, number of applications, or load of read/write operations.

   To customize parameters, review and update `values.yaml` under the InnoDB chart. See Customizing InnoDB Values on page 154.

3. On a HPE Ezmeral Runtime Enterprise Kubernetes cluster master node, run the `ezmysql` installation script:

   ```
   # ./ezmysql_install.sh
   ```

   The script prompts you to enter a password. Make a note of the password that you use.

   The script performs the following:

   - Creates a namespace `ezmysql`
   - Installs MySQL Operator
   - Installs MySQL InnoDB Cluster
   - Configures the router for HA
   - Configures HPA for auto-scaling
   - Provides a Gateway endpoint of MySQL instance for connectivity

4. To check that the MySQL server is accessible after successful install, you can verify MySQL cluster connectivity. To verify connectivity, use one of the following options:
   - **Option 1**: Use the MySQL endpoint details provided from the step 3 output and the same password given during installation in step 3:

     ```
     mysql -u <user-name> -p -h <host> -P <port>
     ```

     For example:

     ```
     # mysql -u root -p -h example.hpecorp.net -P 10022
     ```

   - **Option 2**: Use a different client or application with `root` as the username and password.
5. **(Optional)** You can upgrade the MySQL version for your Shared RDBMS installation. Proceed as follows to upgrade the MySQL version from version 8.0.30 to version 8.0.31:
   a. Download and extract the latest MySQL version:

      ```
      # wget https://ezml-release.s3.amazonaws.com/5.6.0/
      mysql-cluster-ere560.tar.gz;tar -zxvf mysql-cluster-ere560.tar.gz;cd
      mysql-cluster;chmod +x ezmysql_upgrade.sh
      ```

   b. Start the MySQL version upgrade:

      ```
      # sh ezmysql_upgrade.sh
      ```

      The upgrade completes in about 10 minutes.

Customizing InnoDB Values

You can customize the following parameters by using `./mysql-innodbcluster/values.yaml`:

- **User name**: The default value is `root`.

  ```
  credentials:
    root:
      user: root
  ```

- **Instances**: You can change server and router instances.
  - Server instances refers to the number of MySQL servers required.
  - Routers are used to control the communication between MySQL servers for load balancing.

  ```
  serverInstances: 3
  routerInstances: 1
  ```

- **Storage size**: You can update the storage size for MySQL InnoDB clusters. Data volume is used for persistent storage of application data.

  The default value is `40Gi`.

  ```
  datadirVolumeClaimTemplate:
    accessModes: ReadWriteOnce
    resources:
      requests:
        storage: 40Gi
  ```

- **Resource limits:** You can update resource parameters such as CPU and memory according to your requirements.

  The default value for memory is `3G` and the default value for CPU is `400m`.

  ```
  resource:
      request:
        memory: "2G"
        cpu: "200m"
      limits:
        memory: "3G"
        cpu: "400m"
  ```

- **Replicas**: You can increase the number of replicas in case of high load.

  The default setting is `minReplicas: 3` and `maxReplicas: 5`.

  ```
  hpa:
    spec:
      maxReplicas: 5
      minReplicas: 3
      metricsCpuAverageUtilization: 70
      metricsMemoryAverageUtilization: 70
  ```

  For best performance, do not change the values for `metricsCpuAverageUtilization` and `metricsMemoryAverageUtilization`.

Creating Backups and a Backup Schedule

Update backup PVC size and schedule: The default PVC value is `100Gi` and default backup frequency is one time per day. For example:

```
backup:
  schedule: "0 1 * * *"
  pvcSize: 100Gi
```

Editing the backup schedule:

NOTE: Before editing the backup schedule, ensure that MySQL is running and has backups scheduled.

- If you want to stop the backup scheduler or change the backup frequency, you can edit `cronjob.batch/sqlapp-backup-job` with the following command:

  ```
  #kubectl edit cronjob.batch/sqlapp-backup-job -n ezmysql
  ```

- To suspend the backup scheduler, look for the `suspend` key and set it to `True`.

- To change the frequency of backup, edit the `schedule` key values.

For example:

```
schedule: "0 2 * * *"
suspend: false
```

Restoring the MySQL database from backup:

You can restore a database from a backup dump by running the following script:

```
# ./ezmysql_restore.sh
```

This script restores the MySQL database from the backup taken on the PVC. You can fetch the full backup from the following tenant share path:

```
path: /opt/bluedata/mapr/mnt/<df-cluster>/<pvc-volume-path>/<backup-dump>
```

For example:

```
/opt/bluedata/mapr/mnt/df01/844d9e48-k8s-6--sklyrckwmf/EZML_14102022_06_40/
```

Uninstalling MySQL Operator and MySQL InnoDB Cluster

IMPORTANT: Do not uninstall MySQL Operator and MySQL InnoDB Cluster unless you are performing a POC or other test.

NOTE: The uninstallation script is customized for `ezmysql` namespaces only. Namespaces are cleaned after each uninstall, and PVCs are deleted.

To uninstall MySQL Operator and MySQL InnoDB Cluster, proceed as follows:

1. Run the uninstallation script:

   ```
   # ./ezmysql_uninstall.sh
   ```

   The following actions are performed:

   - Uninstalls MySQL InnoDB Cluster
   - Uninstalls MySQL Operator
   - Deletes PVCs

2. On successful uninstallation, the following message appears:

   ```
   No resources found in ezmysql namespace.
   ```

Deploying the Model Management Service

This topic describes steps to install the Model Management Service for HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops. While multiple tenants are able to share a model registry and experiment tracker at the cluster level, the Model Management Service employs rule-based access controls to ensure that users can access only their own models and metadata. The Model Management Service is required for Project Members to run Experiments.

Deploying the Model Management Service

NOTE: The Model Management Service is required for Project Members to create Experiments.

Prerequisites:

- **Required access rights:** You must have bucket creation rights.
- An object store for storing model artifacts. You can use HPE Ezmeral Runtime Enterprise Data Fabric Object Store, or an object store from another source such as AWS S3.

  For information on Data Fabric Object Store, see Object Store (S3 Gateway) Overview on page 665.
- A MySQL RDBMS for storing experiment runs. For best results, Hewlett Packard Enterprise recommends using the HPE Ezmeral Runtime Enterprise Shared RDBMS feature (see Installing Shared RDBMS).

  NOTE: HPE Ezmeral Runtime Enterprise does not support the Shared RDBMS feature on CentOS. If you are using CentOS as your host OS, you can use a MySQL RDBMS other than Shared RDBMS to store experiment runs.

Installation steps:

1. Download the application bundle on the Kubernetes master node:

```
# wget https://ezml-release.s3.amazonaws.com/5.6.0/
model-mgmt-ere560.tar.gz

# tar -xvzf model-mgmt-ere560.tar.gz && cd model-mgmt
```

2. **(Optional)**: If you are using Data Fabric Object store, you can create a secure connection to the Object Store by copying the certificates from the Data Fabric master:

```
# scp root@<DF-ClusterMaster-IP>:/opt/mapr/conf/ca/chain-ca.pem ~/
model-mgmt/mlflowtrack/
```

3. Log in to the Kubernetes master node and execute the install script:

```
# sh ezml_model_mgmt_install.sh
```

4. The following information is required for installation. If the database is hosted on the same cluster you are installing the Model Management Service on, the script fetches the database details automatically.

```
Below Object store configuration required to install the ModelMGMT.
   object_store_host:
   object_store_access_key:
   object_store_secret_key:
Press enter to continue...

Press Enter to continue with installation.
```

   Press **Enter** to continue with installation.

5. The install script scans for a MySQL endpoint in an existing cluster.

- If the MySQL endpoint is found, the script proceeds with this endpoint, and asks whether or not to use a signed certificate for communication:

```
MYSQL end point identified in ezmysql NameSpace:
  mip-bd-vm184.mip.storage.hpecorp.net:10017

  Proceeding with above mysql endpoint.....
```

- If no MySQL endpoint is found, the script asks for a MySQL hostname and port number. After you have entered the hostname and port number for a MySQL endpoint, the script asks for the database password:

```
Mysql operator is not available in ezmysql namespace. Please
provide the endpoint.
Enter the DB host name:
mip-bd-vm1094.example.net
Enter the DB port Number:
10006

Enter the PASSWORD for database user:
Note: password will be hidden when typing
```

6. Select the communication type (secure or insecure):

- To create a secure connection to the Data Fabric Object Store with certificates, enter **Y**:

```
Are you using the self signed certificate for communication (Y/N):
Y
```

The script prompts you to enter your certificate file path:

```
Then provide absolute path for self signed certificate.
  Please provide the Certificate file(*.pem) with absolute path.
```

NOTE: When creating a secure connection, copy the certificate file from the Data Fabric master (`/opt/mapr/conf/ca/chain-ca.pem`) to the local path. For example:

```
scp root@<DF-ClusterMaster-IP>:/opt/mapr/conf/ca/chain-ca.pem
<local-path>
```

- To create a secure connection to the Data Fabric Object Store without certificates, or to use a different object store type (for example, AWS S3), enter **N**:

```
Are you using the self signed certificate for communication (Y/N):
N
```

7. The script asks for the Object Store bucket configuration:

```
Please enter the Object Store bucket configuration.
Below is an example for these entries

object_store_host: https://XXXXXXXXXXXXX:9000
object_store_access_key: XXXXXXXXXXXXX
object_store_secret_key: XXXXXXXXXXXXX

Bucket name: ezmodel-mgmt-k8s-4
Enter the object_store_host
URL: https://m2-bd-vm2118.example.net:9000
Enter the object_store_access_key:
Note: Access_key will be hidden when typing
Enter the object_store_secret_key:
Note: Secret_key will be hidden when typing
```

After entering the required information, the script begins installation:

```
Installing the mlflowtrack in namespace ezml-model-mgmt ....

Installing the MODELMGMT in namespace ezml-model-mgmt ....

ModelMGMT installation is in Progress ......!!!!

ModelMGMT installation has been completed successfully.
Below are endpoint & bucket that will be used to interact with Model
Mgmt
MLFLOW_S3_ENDPOINT_URL: s3://ezmodel-mgmt-k8s-4
MODELMGMT_BACKEND_URL: http://mip-bd-vm1094.example.net:10035
*** Please apply the generated secret model-mgmt-secret.yaml to tenant
namespace

Usage: kubectl create -f model-mgmt-secret.yaml -n <tenant-namespace>
```

8. The script creates the YAML file `model-mgmt-secret.yaml` for the secret. Use the following command to create the secrets with `tenant-namespace`:

```
# kubectl create -f model-mgmt-secret.yaml -n <tenant-namespace>
```

Toolbar & Main Menu - ML Ops Project Member

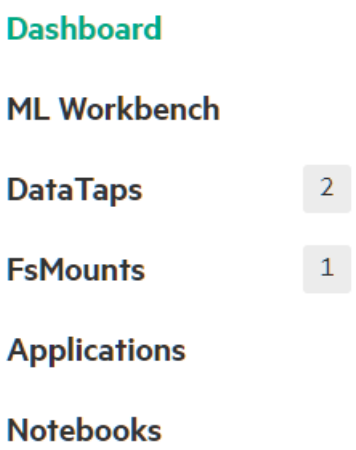
Describes the toolbar and navigation sidebar available to users with Tenant Member access rights to an ML Ops project in HPE Ezmeral Runtime Enterprise.

Toolbar

The layout of the Toolbar is the same as described in Navigating the GUI on page 143.

Main Menu (ML Ops Project Member)

The main menu for ML Ops projects appears as shown in the following image:



Dashboard	Opens the Kubernetes **Dashboard** screen. See Dashboard - Kubernetes Tenant Member on page 357
ML Workbench	Opens the **HPE Ezmeral Runtime Enterprise new UI** in a separate browser tab or window, and displays the **Overview** tab of **Project Details** screen of of this project. **The interface that is displayed is the primary interface you use to access machine learning (ML Ops) projects.**
DataTaps	Opens the **DataTaps** screen, which enables you to upload and download files.
FS Mounts	Opens the **FS Mounts** screen, which enables you to upload and download files.
Applications	Opens the **Kubernetes Applications** screen, which enables you to launch applications within Kubernetes pods and access service endpoints and virtual endpoints.
Notebooks	Opens the **Notebooks** screen, from which you can launch notebook servers and view notebook endpoints.

Related reference

Toolbar & Main Menu - Tenant or Project Administrator on page 389
Describes the toolbar and navigation sidebar available to users with Kubernetes Tenant/Project Administrator access rights in HPE Ezmeral Runtime Enterprise.

More information

ML Ops Tasks on page 160
This topic describes Project Administrator and Project Member Tasks in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

ML Ops Tasks

This topic describes Project Administrator and Project Member Tasks in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Data Sources

The topics in this section describe using data sources. Data sources are available for use in both HPE Ezmeral ML Ops projects and non-HPE Ezmeral ML Ops projects.

Adding Data Sources

This topic describes adding data sources in HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Project Administrator

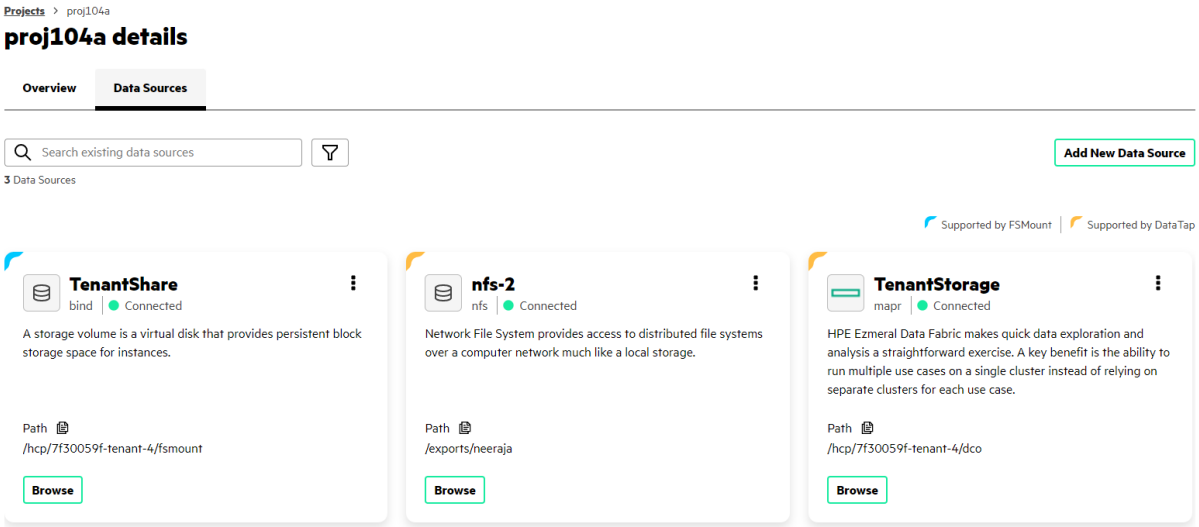
About this task

Connect data sources to your project to allow Project Members access to data required for experiments.

Procedure

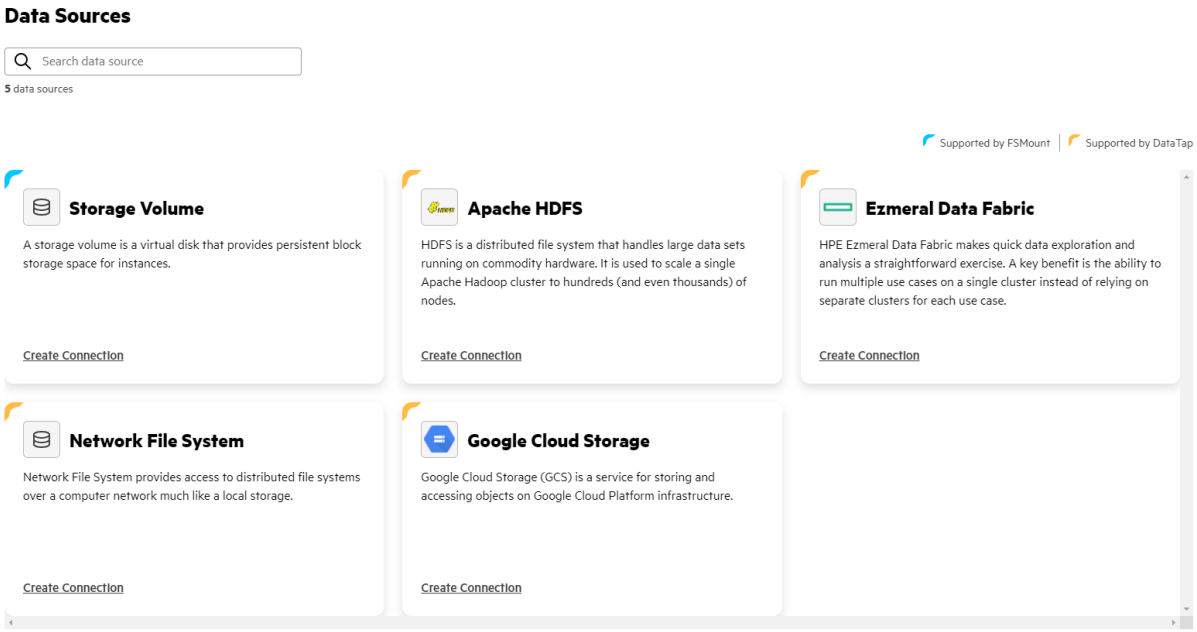
1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Perform one of the following:
   - Select the **Data Sources** tab.
   - Select a data source name or **View All** on the **Data Sources** panel.

   The **Data Sources** tab opens.



3. To add an additional data source, select **Add New Data Source**.

   The data source creation screen opens:



4. Select **Create Connection** from the panel of the data source type that you want to create:
   - Storage Volume
   - Apache HDFS
   - Ezmeral Data Fabric
   - Network File System
   - Google Cloud Storage

   Depending on the selected data source type, a side-drawer with different input fields opens. Enter your connection information in the provided fields.

5. When you are finished, select **Connect**.

Editing Data Sources

This topic describes editing data sources in HPE Ezmeral Runtime Enterprise.

Prerequisites

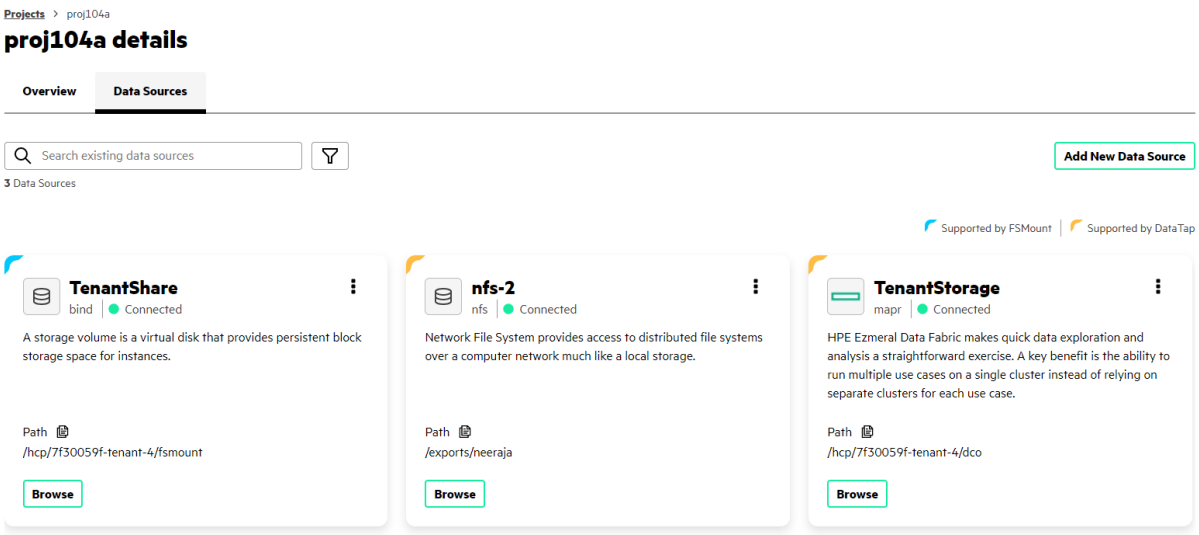
Required access rights: Project Administrator or Project Member

Procedure

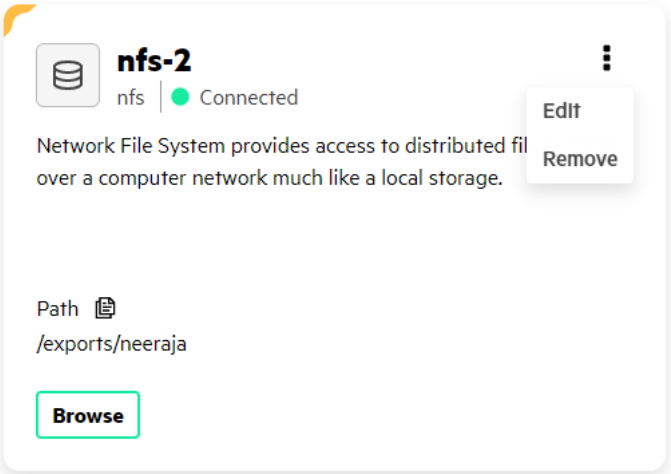
1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. Perform one of the following:
   - Select the **Data Sources** tab.
   - Select a data source name or **View All** on the **Data Sources** panel.

   The **Data Sources** tab opens.



3. On the data source you want to edit, open the **actions** menu and select **Edit**.



NOTE: If a data source is unavailable for editing or removal, the action menu options are disabled, and appear dimmed.

4. The **Edit** menu opens.

   Depending on the data source type, a side-drawer with different input fields opens. Enter your updated connection information in the provided fields.

5. When you are finished, select **Save**.

Deleting Data Sources

This topic describes deleting data sources in HPE Ezmeral Runtime Enterprise.

Prerequisites

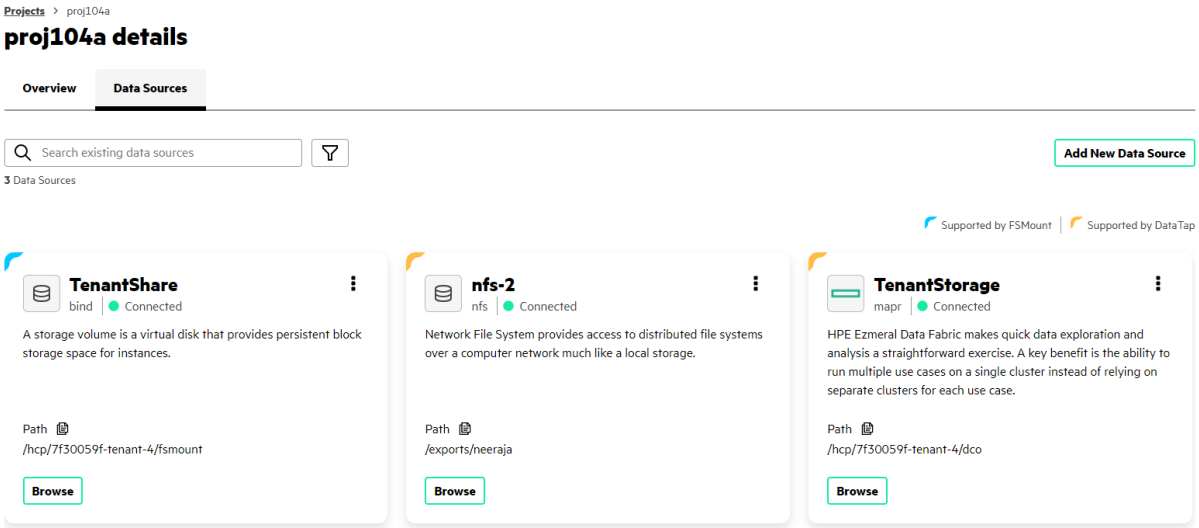
Required access rights: Project Administrator or Project Member

Procedure

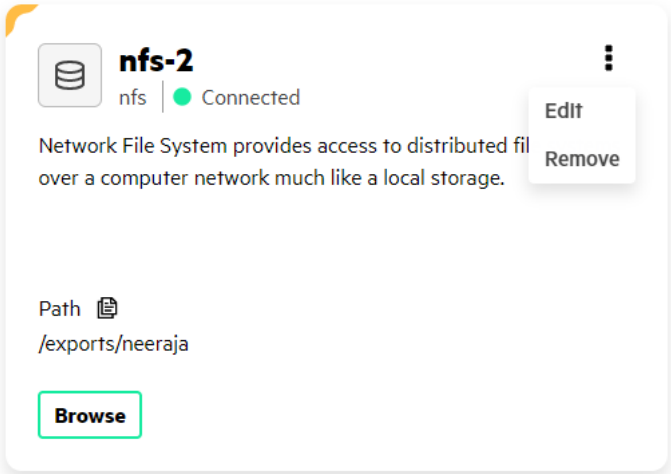
1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. Perform one of the following:
   - Select the **Data Sources** tab.
   - Select a data source name or **View All** on the **Data Sources** panel.

   The **Data Sources** tab opens.



3. On the data source you want to edit, open the **actions** menu and select **Remove**.



NOTE: If a data source is unavailable for editing or removal, the action menu options are disabled, and appear dimmed.

Source Control Configurations

The topics in this section describe using source control configurations. Source control is available for use in both HPE Ezmeral ML Ops projects and non-HPE Ezmeral ML Ops projects.

Creating Source Control Configurations

This topic describes adding source control configuration templates or instances in HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights:

- To create Source Control Configuration templates, Project Administrator access is required.
- To create Source Control Configuration instances, Project Administrator or Project Member access is required.

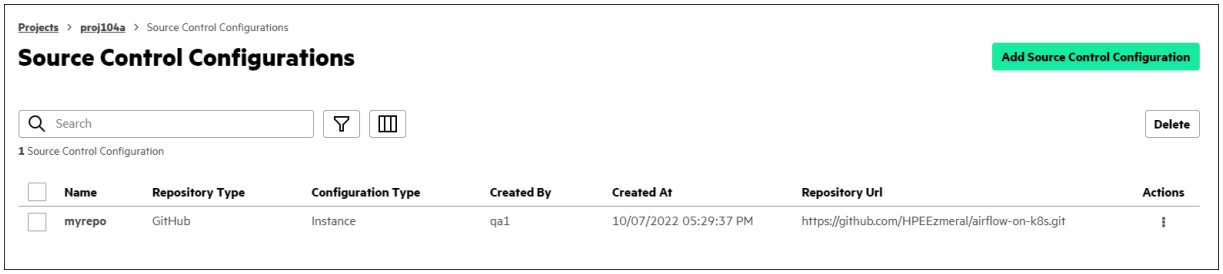
About this task

You must set up source control for a project before creating Kubernetes Notebook clusters in that project. Kubernetes Notebook clusters do not detect source control configurations that are added after the notebook cluster is deployed.

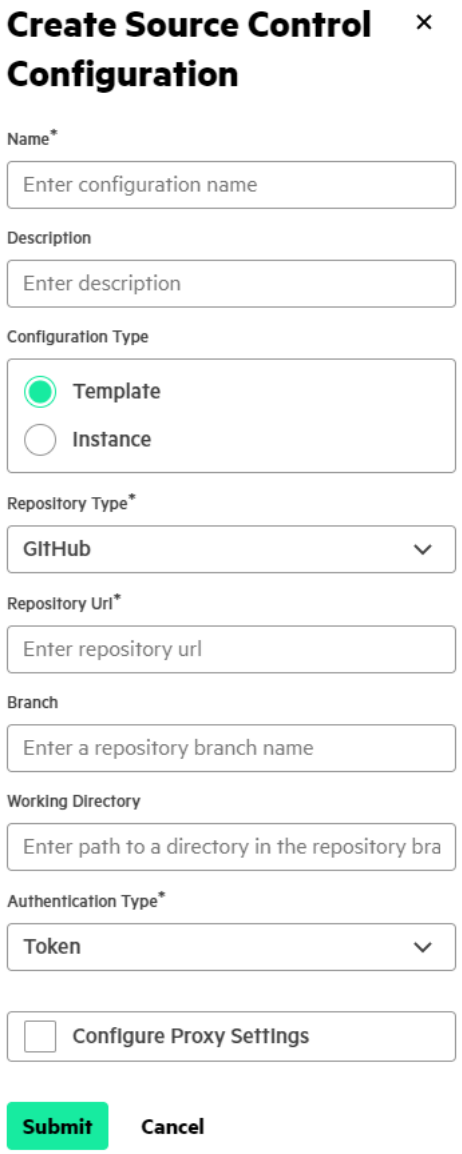
At least one configuration template must be added before Project Member users can create individual source control instances.

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Select **View All** on the **Source Control Configurations** panel. The **Source Control Configurations** screen opens.



3. Select **Add Source Control Configuration**. The **Create Source Control Configuration** side-drawer opens.



4. Enter the information for your Source Control Configuration:
   - **Name**
   - **Description**
   - **(Project Administrator only) Configuration Type**:
     - Select **Template** to create a Source Control Configuration template. At least one Source Control Configuration template must be available for Project Members to create Source Control Configuration instances.
     - Select **Instance** to create a Source Control Configuration instance.
   - If you select **Template** for the **Scope**, the menu has the following fields:
     - **Repository Type**
     - **Repository URL**

- **Branch**
- **Working Directory**
- **Authentication Type**
- **Configure Proxy Settings**

- If you are creating a source control configuration as a Project Member, or you select **Instance** for the **Scope**, the menu has the following fields:
  - **Template**: Select the Source Control Configuration template to use as the basis for your Source Control Configuration instance.
  - **Branch**
  - **Working Directory**
  - **Username**
  - **Email**
  - **Token** or **Password**

  Select **Submit** to create your Source Control Configuration.

Editing Source Control Configurations

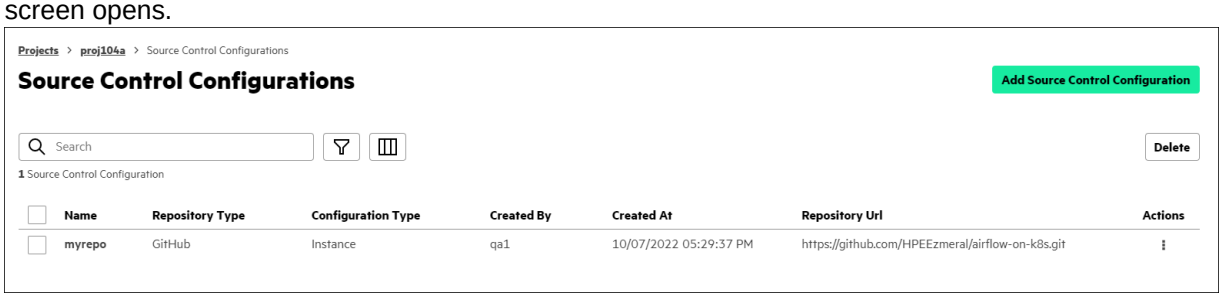
This topic describes editing source control configuration templates and instances in HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Project Administrator or Project Member

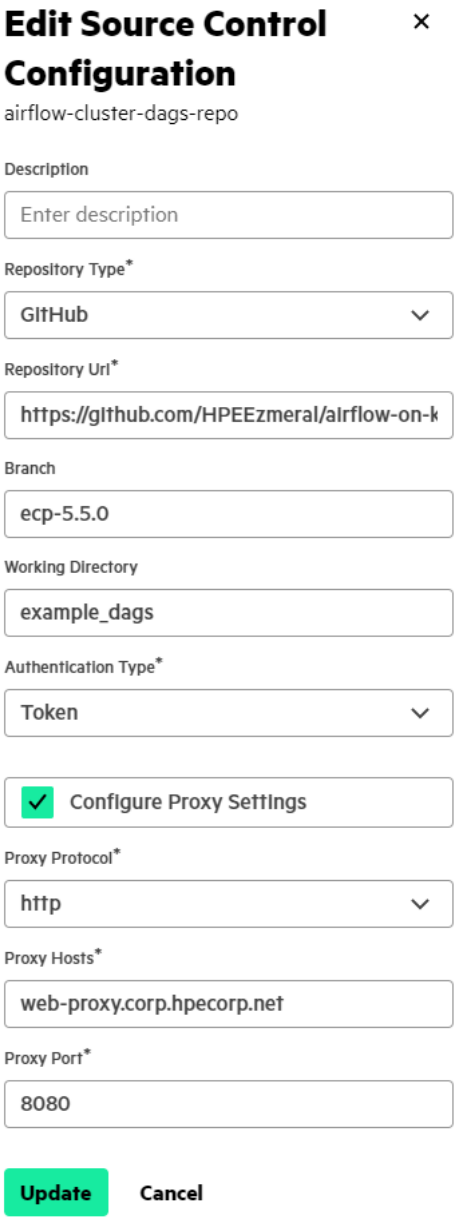
Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Select **View All** on the **Source Control Configurations** panel. The **Source Control Configurations** screen opens.



3. Open the **actions** menu for the source control you want to edit, and select **Edit**.

   The **Edit Source Control Configuration** side-drawer opens.



4. Enter the new information for the source control configuration, and select **Update**.

Deleting Source Control Configurations

This topic describes deleting source control configuration templates and instances in HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Project Administrator or Project Member

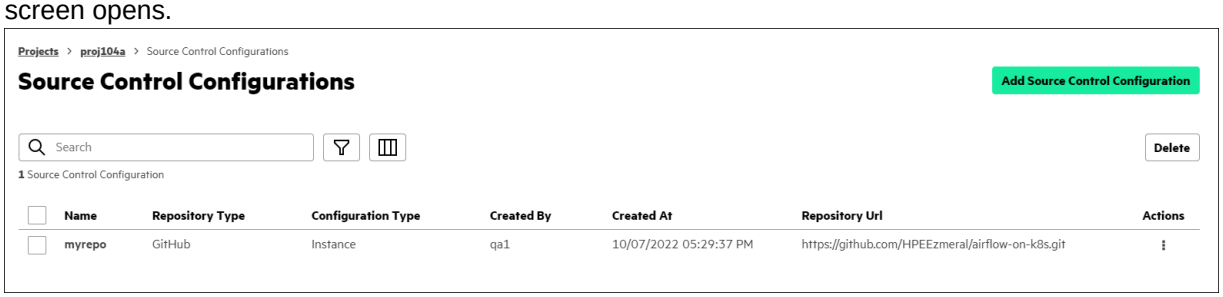
About this task

Project Administrators can delete Source Control Configurations created by other Project Members.

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. Select **View All** on the **Source Control Configurations** panel. The **Source Control Configurations** screen opens.



3. Perform one of the following:
   - Select the check box next to the source control you want to delete, and then select the **Delete** button.
   - Open the **Actions** menu next to the source control you want to delete, and then select the **Delete** action item.

Notebook Servers

The topics in this section describe using Notebook Servers in HPE Ezmeral ML Ops.

Creating Notebook Servers

This topic describes creating notebook servers in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

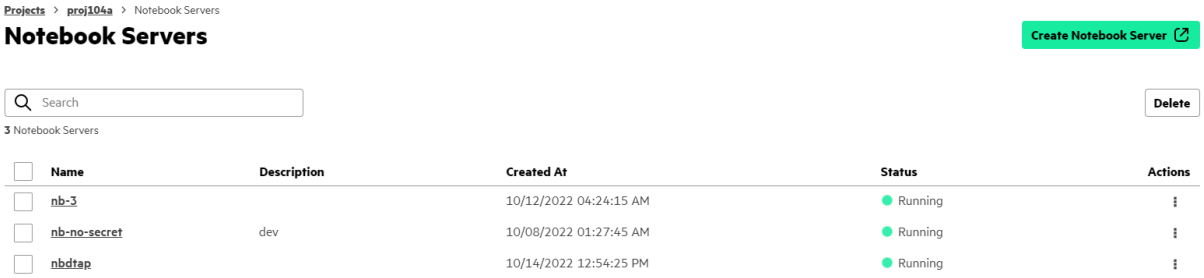
Required access rights: Project Administrator or Project Member

Procedure

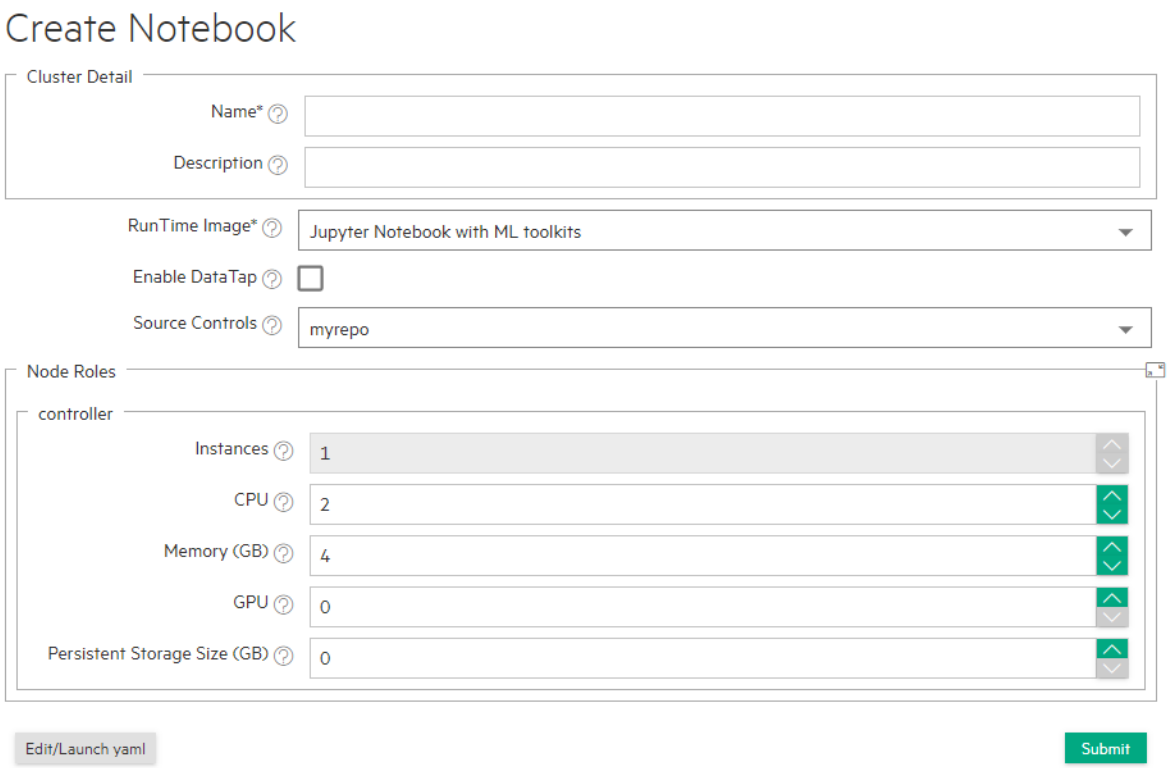
1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. se

3. Select **View All** on the **Notebook Servers** panel. The **Notebook Servers** screen opens.



4. Select **Create Notebook Server**. The **Create Notebook** screen opens.



5. Enter your information into the form.
   - Select **Enable DataTap** to enable DataTap for this notebook. For information about DataTap, see About DataTaps on page 122.
   - Optionally, select or more source controls. Use commas to separate the source controls.
   - You can expand or collapse the information in `Node Roles` by clicking the icon in the upper right corner of the box.
     - The correct role name and number of instances are entered by default, so there is usually no need to display or alter these values.

       Default values for resources are provided, but you can change these values.
     - To request MIG resources, you must edit the YAML file manually. Complete the rest of the entries in the **Create Notebook**, and then click **Edit/Launch yaml**.
6. **(Optional)**: If needed, open the YAML file for editing by clicking **Edit/Launch yaml**.
   - You might need to edit the YAML file in the following circumstances:
     - To request MIG resources: You must edit the YAML file manually to change all `nvidia/gpu:` entries to specify the MIG configuration. For example: `nvidia.com/mig-3g.20gb:`.

       For more information about requesting MIG resources, see Using GPUs in Kubernetes Pods on page 727.
     - If you specify a nonzero value for GPU, the required NVIDIA_DRIVER_CAPABILITIES environment variable setting is added to the YAML file automatically. If you edit the YAML file manually, ensure that the NVIDIA_DRIVER_CAPABILITIES environment is set to `"compute,utility"` as follows:

```
env:
  -
    name: "NVIDIA_DRIVER_CAPABILITIES"
    value: "compute,utility"
```

- If the notebook you are using to build models uses the Model Management service, then you must include the Model Management secret in the YAML file.

  Ensure that Model Management secret (default: `model-mgmt-secret`) appears under `secrets:`. For example:

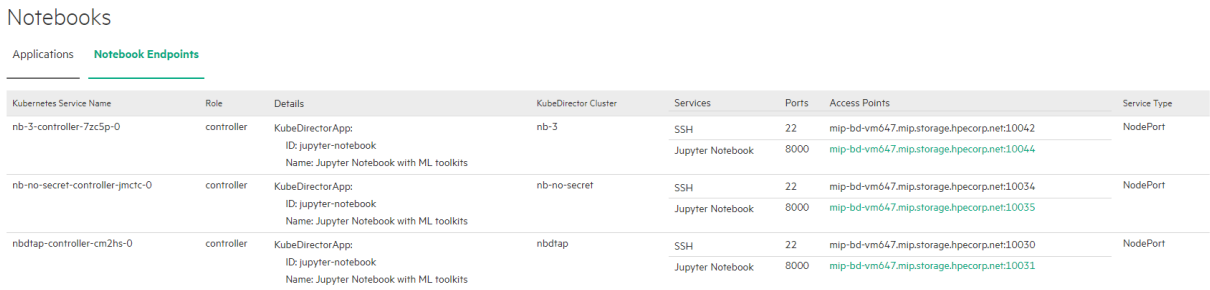
```
apiVersion: "kubedirector.hpe.com/v1beta1"
kind: "KubeDirectorCluster"
metadata:
  name: "jupyter-notebook-instance"
  namespace: "aiml1"
  labels:
    description: ""
spec:
  app: "jupyter-notebook"
  appCatalog: "local"
  connections:
    clusters:
    secrets:
      - hpecp-kc-secret-192e81d6d7054551422bb88bdf9f90a3
      - hpecp-sc-secret-33630169f143ac582f69d43ofa3e3669
      - hpecp-ext-auth-secret
      - model-mgmt-secret
...
```

Results

HPE Ezmeral Runtime Enterprise returns you to the **Notebooks** screen. The new pod that you just created appears in the **Running Applications** table.

When the **Status** of this pod changes to **ready**, then you can access the service endpoints within that pod using one of the following methods:

- Through the command line.
- From the **Notebook Servers** screen in the new HPE Ezmeral ML Ops UI. On the **Notebook Servers** screen, select the notebook name to access the service endpoint.
- From the **Notebook Endpoints** tab in the old HPE Ezmeral ML Ops UI in the **Access Points** column.



Kubernetes Service Name	Role	Details	KubeDirector Cluster	Services	Ports	Access Points	Service Type
nb-3-controller-7zc5p-0	controller	KubeDirectorApp: ID: jupyter-notebook Name: Jupyter Notebook with ML toolkits	nb-3	SSH	22	mip-bd-vm647.mip.storage.hpecorp.net:10042	NodePort
				Jupyter Notebook	8000	mip-bd-vm647.mip.storage.hpecorp.net:10044	
nb-no-secret-controller-jmctc-0	controller	KubeDirectorApp: ID: jupyter-notebook Name: Jupyter Notebook with ML toolkits	nb-no-secret	SSH	22	mip-bd-vm647.mip.storage.hpecorp.net:10034	NodePort
				Jupyter Notebook	8000	mip-bd-vm647.mip.storage.hpecorp.net:10035	
nbdtap-controller-cm2hs-0	controller	KubeDirectorApp: ID: jupyter-notebook Name: Jupyter Notebook with ML toolkits	nbdtap	SSH	22	mip-bd-vm647.mip.storage.hpecorp.net:10030	NodePort
				Jupyter Notebook	8000	mip-bd-vm647.mip.storage.hpecorp.net:10031	

Deleting Notebook Servers

This topic describes deleting notebook servers in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

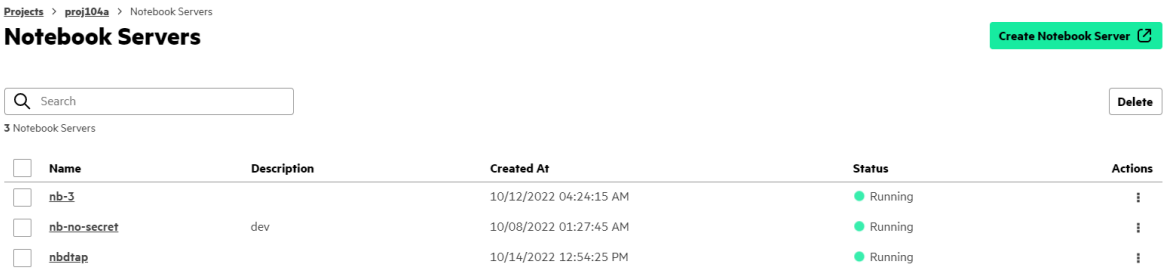
Prerequisites

Required access rights: Project Administrator or Project Member

Procedure

You can delete a notebook server in one of the following ways:

- Through the HPE Ezmeral ML Ops old UI:
  a. Select **Notebooks**. The **Notebooks** screen opens.
  b. Select the **trash can** action button next to the notebook server you want to delete.
- Through the HPE Ezmeral ML Ops new UI:
  a. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
  b. Select **View All** on the **Notebook Servers** panel. The **Notebook Servers** screen opens.



  c. Perform one of the following:
    - Select the check box next to the name of the notebook server that you want to delete. Select **Delete**.
    - Open the **Actions** menu next to the notebook server that you want to delete, and select **Delete**.

Experiments

The topics in this section describe using Experiments in HPE Ezmeral ML Ops.

Viewing Experiment Results

This topic describes how to view experiment results in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

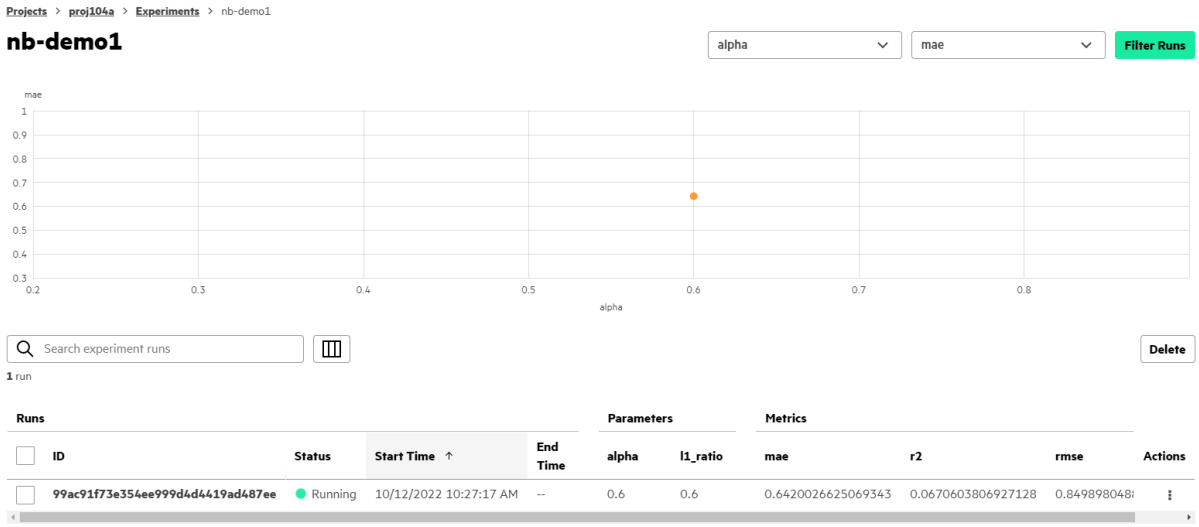
Required access rights: Project Administrator or Project Member

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Select **View All** on the **Experiments** panel. The **Experiments** screen opens.

3. Perform one of the following:
   - Open the **Actions** menu next to the experiment, and select **View Runs**.
   - Select the name of the experiment to view its runs.

   The **Experiment Runs** screen for the selected experiment opens.



4. **(Optional)**: You can display a chart to display the relationship between different parameters and metrics. Select the parameters or metrics that you want to compare from the **dropdown menus**.

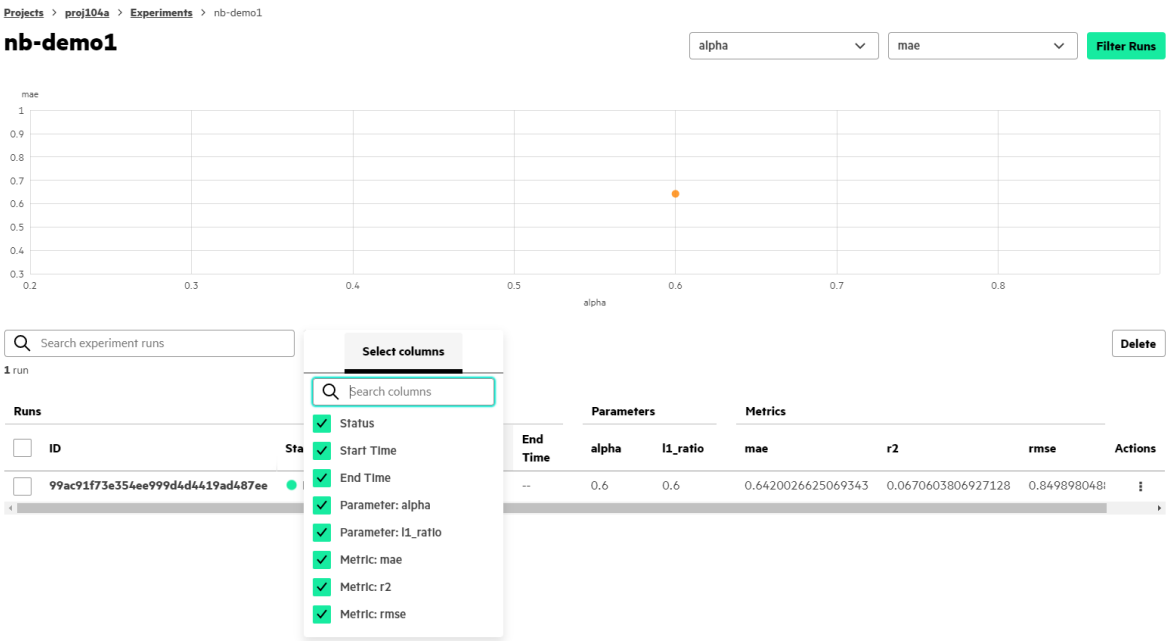
5. **(Optional)**: Select the **Filter Runs** button to do the following:
   - Compare specific experiment runs.
   - Filter experiment runs based on the displayed list of statuses. This list is dynamically generated based on the actual states of your experiment runs.

   After selecting filters, select **Apply** to filter results based on the selected criteria.

   To remove the filters, select **Reset**.

6. **(Optional)**: Select the **Columns** button to choose parameters, metrics, and tags to display as columns in the **Runs** table.

   The first three items in each section (Parameter, Metric, and Tag) are selected by default. If you modify the selected items, your new settings are saved and load the next time you open the **Experiment Runs** screen for the selected experiment.



Deleting Experiments

This topic describes deleting experiments in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

Required access rights: Project Administrator or Project Member

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Select **View All** on the **Experiments** panel. The **Experiments** screen opens.

3. Perform one of the following:
   - Select the check box next to the name of the experiment that you want to delete. Select **Delete**.
   - Open the **Actions** menu next to the experiment that you want to delete, and select **Delete**.

Deleting Experiment Runs

This topic describes deleting experiment runs in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

Required access rights: Project Administrator or Project Member

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. Select **View All** on the **Experiments** panel. The **Experiments** screen opens.

3. Perform one of the following:

   - Open the **Actions** menu next to the experiment, and select **View Runs**.
   - Select the name of the experiment to view its runs.

   The **Experiment Runs** screen for the selected experiment opens.

4. Perform one of the following:

   - Select the check box next to the name of the experiment run you want to delete. Select **Delete**.
   - Open the **Actions** menu next to the experiment run you want to delete, and select **Delete**.

Models

The topics in this section describe using Models in HPE Ezmeral ML Ops.

Registering Models

This topic describes registering models in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

Required access rights: Project Administrator or Project Member

About this task

Before registering models, you must run experiments. Each experiment run creates a model. You can compare metrics for experiment runs in the UI, as described in Viewing Experiment Results on page 172.

After deciding which model of an experiment run best suits your needs, you can register the model for later deployment. You must register the model in the model registry before the model can be deployed into production.

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

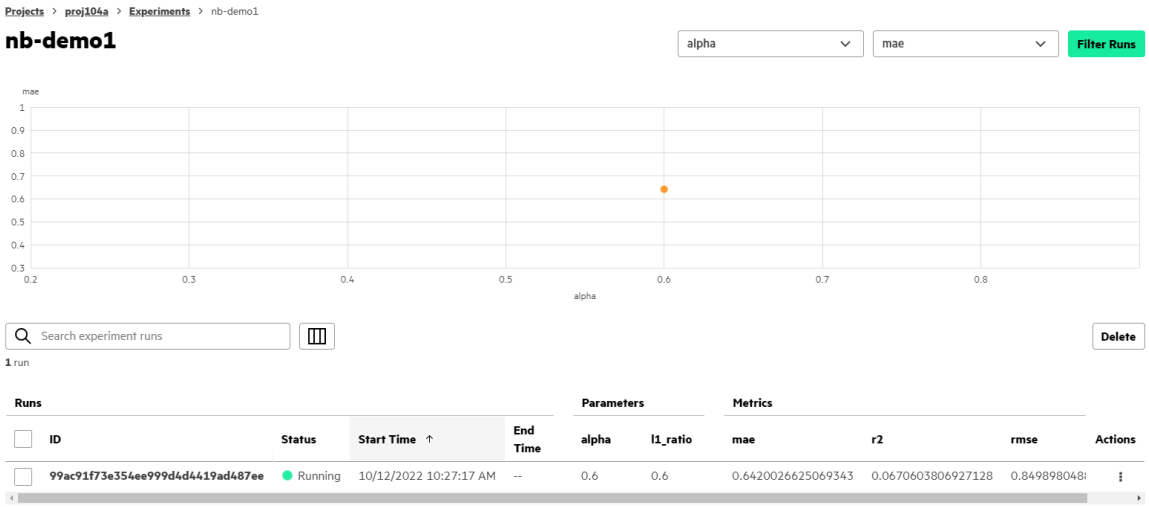
2. **Option 1**: To register a model from the **Experiments** screen, proceed as follows:

   a. Select **View All** on the **Experiments** panel. The **Experiments** screen opens.

   b. Perform one of the following:

      - Open the **Actions** menu next to the experiment, and select **View Runs**.
      - Select the name of the experiment to view its experiment runs.

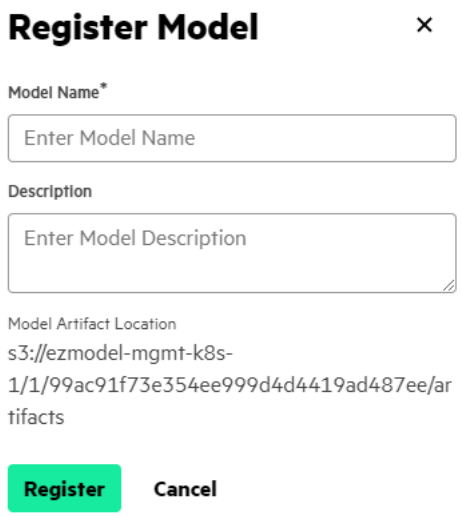
      The **Experiment Runs** screen for the selected experiment opens.



   c. Open the **Actions** menu next to the experiment run you want to register as your model, and select **Register**.

      NOTE: If the model is in a Killed or Failed state, the **Register** action does not appear in the **Actions** menu.

   d. The **Register Model** side-drawer opens:



- **Model Name**
- **Description**
- **Model Artifact Location**: The model artifact location is the location of the model in the object store cofigured for the model management service. When you register a model from the **Experiment Runs** screen, the information in this field is automatically generated.

Select **Register**.

3. **Option 2**: To register a model from the **Model Registry** screen, proceed as follows:

   a. Select **View All** on the **Model Registry** panel. The **Model Registry** screen opens.

   b. Select **Register Model**. The **Register Model** side-drawer opens:

   - **Model Name**
   - **Description**
   - **Model Artifact Location**: The model artifact location is the location of the model in the object store cofigured for the model management service. To retrieve the location, navigate to the object store and find the artifacts, or copy the location from the **Experiment Runs** screen.

   Select **Register**.

Editing Registered Models

This topic describes editing registered models in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

Required access rights: Project Administrator or Project Member

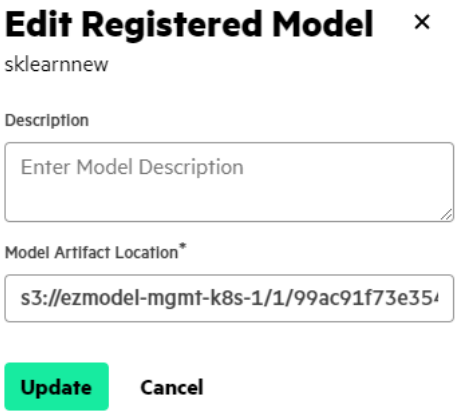
About this task

Editing a registered model does not impact any deployed instances of the registered model.

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Select **View All** on the **Model Registry** panel. The **Model Registry** screen opens.

3. Open the **Actions** menu next to the model that you want to edit, and select **Edit**. The **Edit Registered Model** side-drawer opens.



4. Update the model **Description** and **Model Artifact Location**.
5. When you are done, select **Update**.

Deleting Registered Models

This topic describes deleting deployed models in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

Required access rights: Project Administrator or Project Member

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Select **View All** on the **Model Registry** panel. The **Model Registry** screen opens.

3. Perform one of the following:
   - Select the check box next to the name of the model that you want to delete. Select **Delete**.
   - Open the **Actions** menu next to the model that you want to delete, and select **Delete**.

Deploying Models

This topic describes deploying models in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

Prerequisites

Required access rights: Project Administrator or Project Member

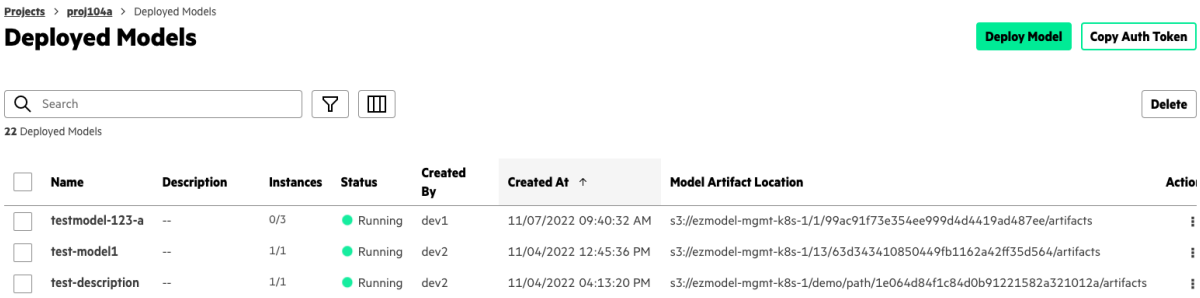
About this task

This task is part of the process to put a model into production. After the model has been developed and registered in the HPE Ezmeral Runtime Enterprise model registry, you deploy the model, which enables prediction calls to be sent to this model.

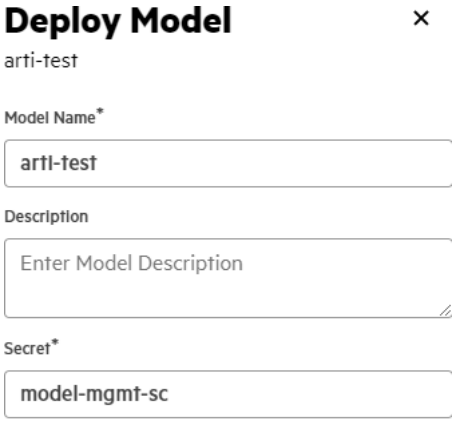
Procedure

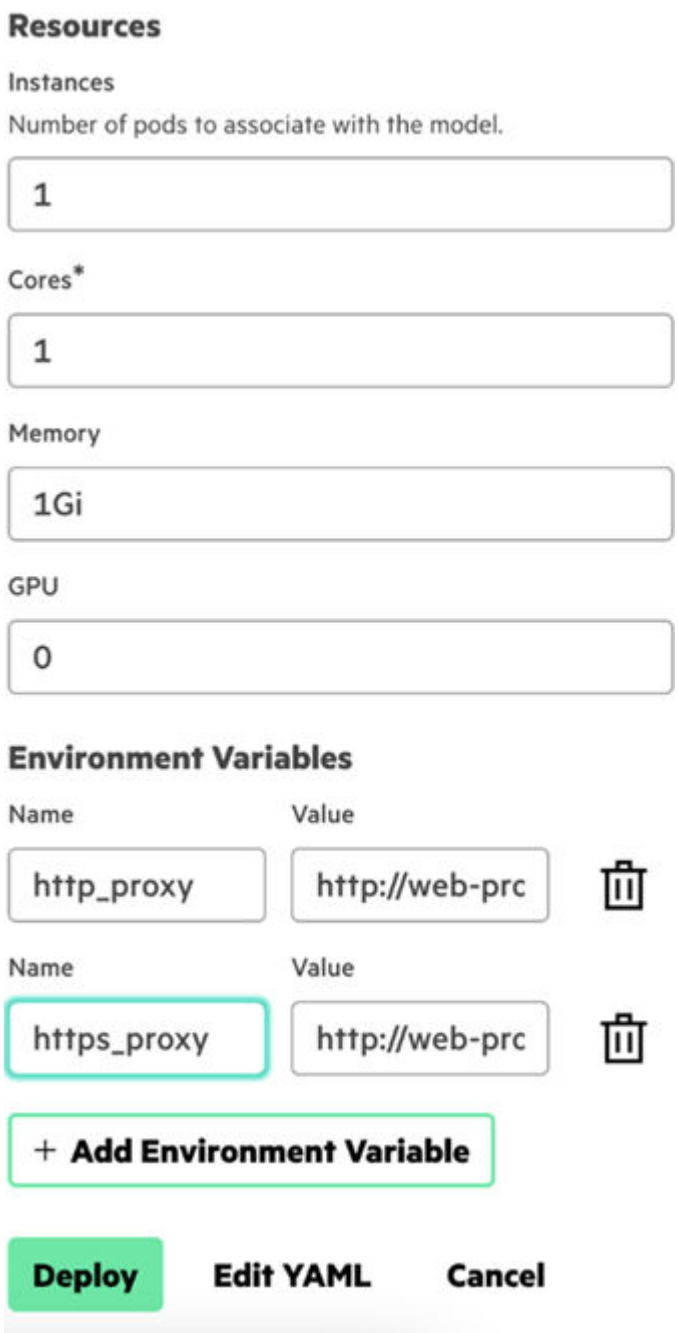
1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.
2. Select one of the following:
   - **View All** on the **Model Registry** panel. The **Model Registry** screen opens.

   - **View All** on the **Deployed Models** panel. The **Deployed Models** screen opens.



3. Select the model to deploy:
   - If you are deploying from the **Model Registry** screen, open the **Actions** menu next to the model you want to deploy, and select **Deploy**.
   - If you are deploying from the **Deployed Models** screen, select the check box next to the registered models you want to deploy, and then select **Deploy Model**.
4. The **Deploy Model** side-drawer opens:





- **Model Name**: By default, HPE Ezmeral ML Ops selects the same name as the registered model.

  You can deploy the same model multiple times under different names.

- **Description**

- **Secret**: HPE Ezmeral Runtime Enterprise automatically fills the **Secret** field with the Model Management secret. If no Model Management secret is found, you must enter a secret.

- **Resources**: Depending on the expected traffic for your model, you can create additional instances. Depending on the size and complexity of your model, you can select more cores and memory.

  - **Instances**

  - **Cores**

    - **Memory**

- **GPU**

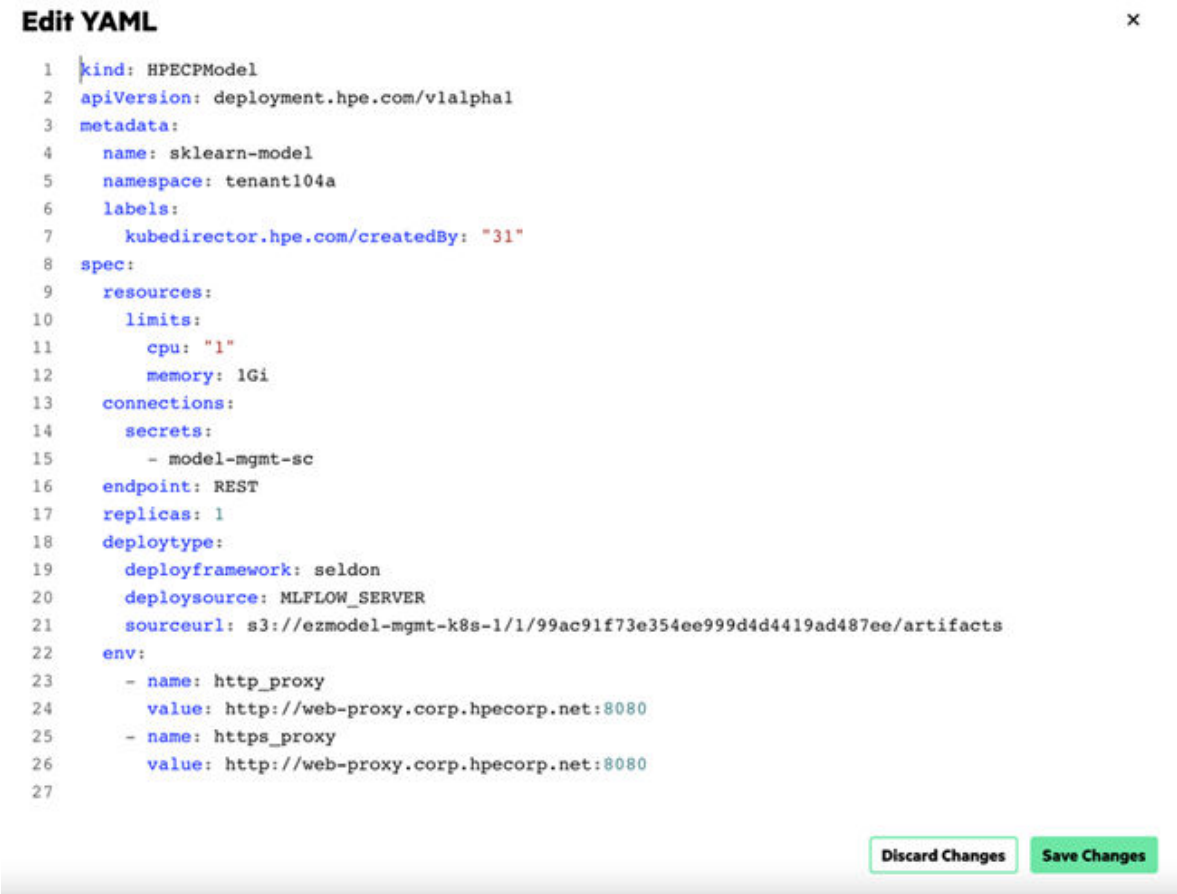
- **Environment Variables**:

    - **Name**

    - **Value**

    To add another environment variable, select **Add Environmental Variable**. To delete an environmental variable, select the **Trash Can** button next to the environmental variable you want to delete.

- **Edit YAML**: To change additional values not shown in the HPE Ezmeral Runtime Enterprise UI, click the **Edit YAML** button. After you close the **Edit YAML** window, the HPE Ezmeral Runtime Enterprise UI updates to reflect changes made to the YAML file.



5. Select **Deploy** to deploy the model. The model now appears on the **Deployed Models** screen.

Viewing Model Information

This topic describes viewing model information in HPE Ezmeral ML Ops, including model details, logs, events, and deployment events.

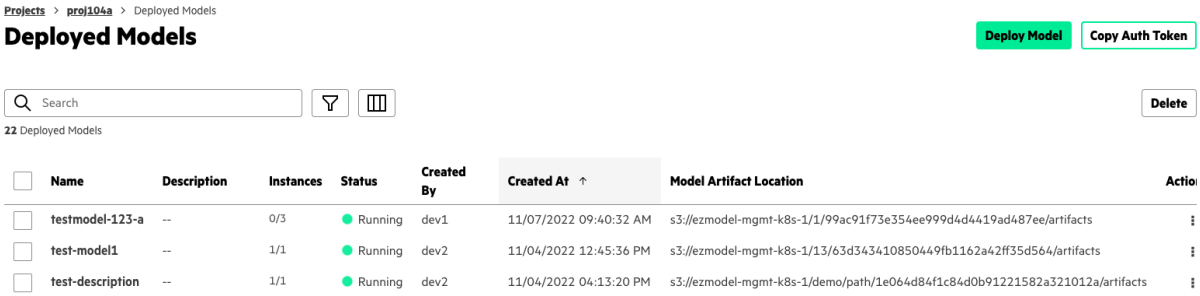
Prerequisites

Required access rights: Project Administrator or Project Member

Procedure

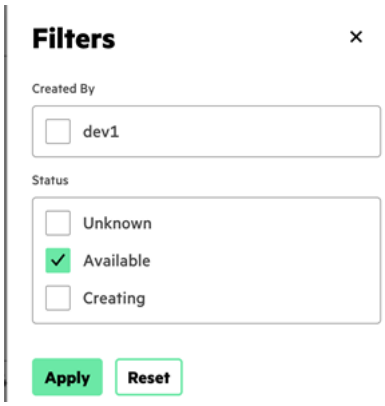
1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. Select **View All** on the **Deployed Models** panel. The **Deployed Models** screen opens.



3. From the **Deployed Models** screen, the following information is available.

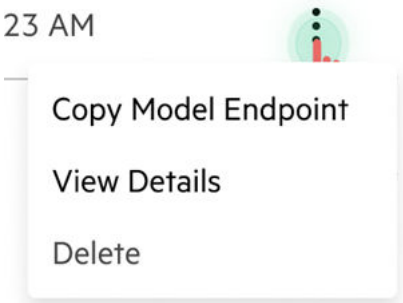
   - To filter models based on their deployment status and the user who created the model, select the **Status** column. The **Filters** side drawer opens:



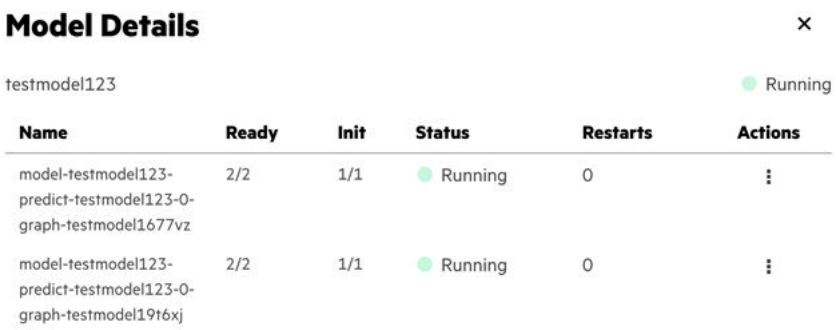
   - The **Model Details** side drawer displays all instances of a model and their container statuses.

     To open the **Model Details** side drawer:

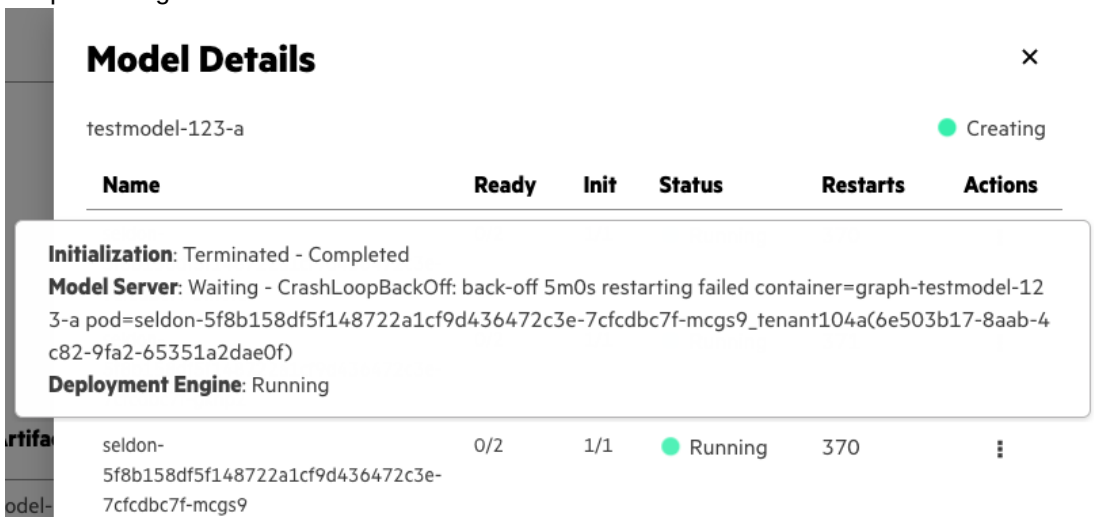
     a. Open the **action items** dropdown menu, and select **View Details**.



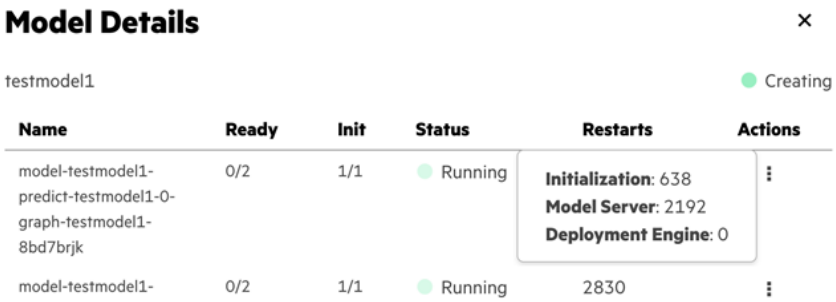
     b. The **Model Details** side drawer opens.



- To display detailed information about a deployed model pod, hover your pointing device over the pod listing:

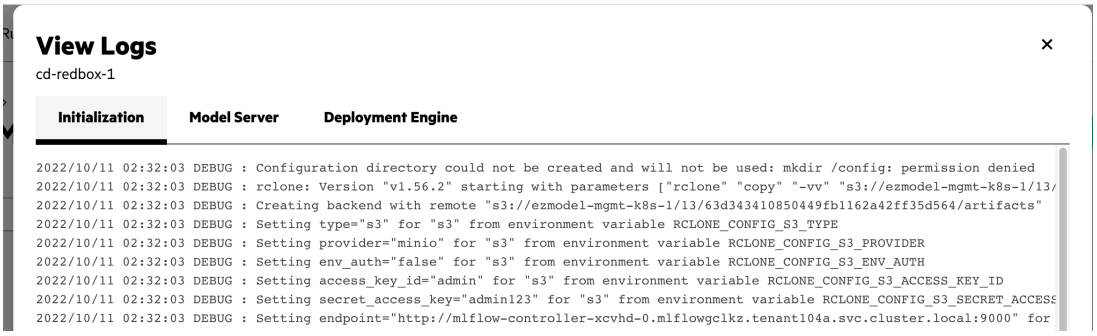


- To view detailed restart information for a pod, hover your pointing device over the **Restarts** column on a pod listing:

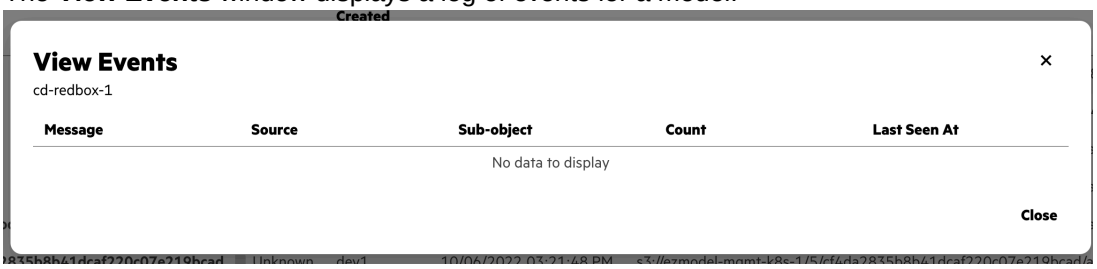


c. From the **Model Details** side drawer, you can open the **action items** dropdown to access the following information:

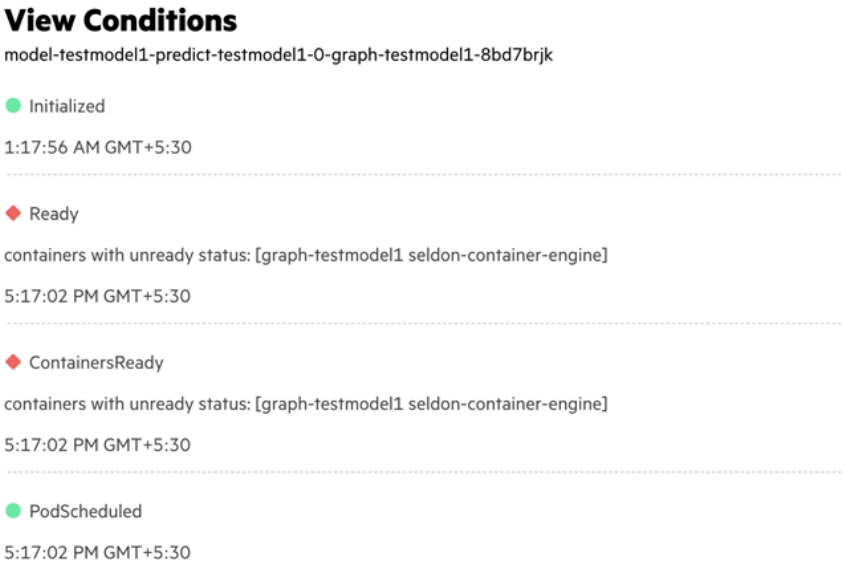
- The **View Logs** side drawer displays logs for a model.



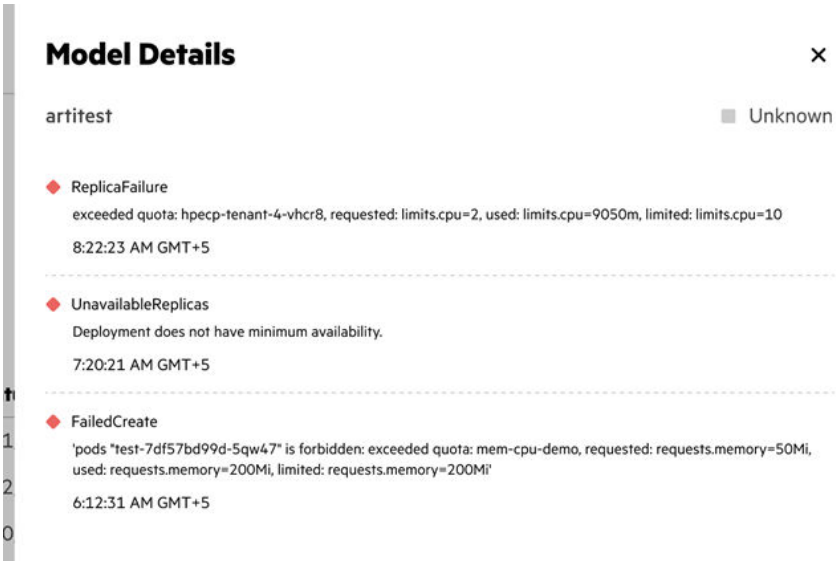
- The **View Events** window displays a log of events for a model.



- The **View Conditions** window displays a list of conditions of the deployed pods of a model.



- If a model does not start because of resource issues, you can view a table of deployment events for a model.



Making Prediction Calls With Deployed Models

This topic describes the requirements and a template for making a prediction call to the deployed model.

Prerequisites

Required access rights: Project Administrator or Project Member

About this task

This task is part of the process to put a model into production. After the model has been developed and registered in the HPE Ezmeral Runtime Enterprise model registry, you deploy the model, which enables prediction calls to be sent to this model.

Procedure

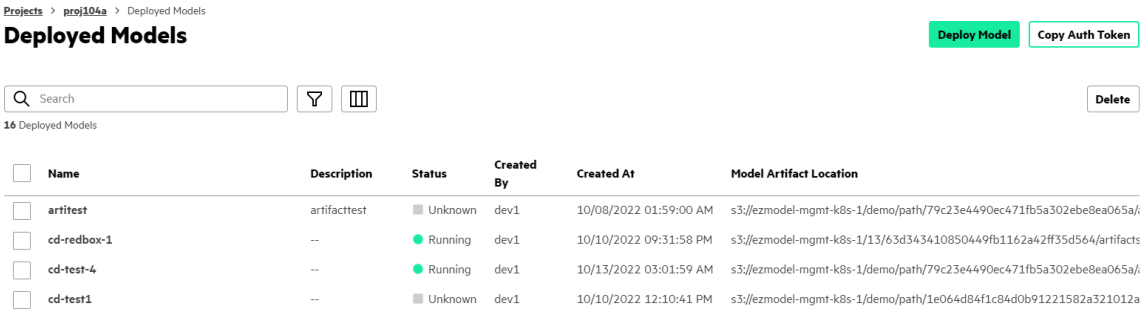
1. Get an auth token for the prediction call.

   To make a prediction on the deployed model, a user token is required. The procedure to get the token depends on whether you are using the grapical user interface (GUI) or the command line.

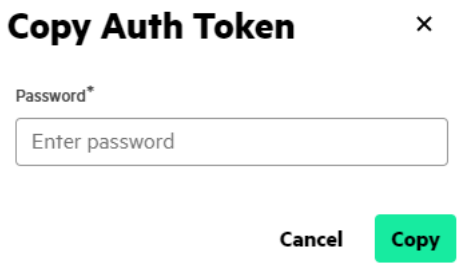
   Using the GUI:

   a) Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

   b) Select **View All** on the **Deployed Models** panel. The **Deployed Models** screen opens.



   c) Select **Copy Auth Token**. The **Copy Auth Token** menu appears.



d) Enter your password and select **Copy**.

Using the command line:

a) Record the URL of the Kubeflow Dashboard for use in a later step. To display the URL, in a tenant view, select the **Dashboard** tab.

b) Enter the following `kubectl` command:

```
kubectl get svc kftoken-svc --n prism-ns --o yaml
```

c) In the output, the annotation for `hpecp-internal-gateway/10001` provides the URL to use to obtain the auth todken. Record that URL.

d) To get the auth token, enter the following command, substituting your own values for <variable> items:

```
curl --location --request POST 'http://<auth-token-provider-url>/
token' --header 'Content-Type: application/json' --data-raw
'{  "kubeflow_dashboard": "<dashboard-url>",  "user": "<username>",
"password": "<password>"}'
```

The token expires after 24 hours by default. After the token expires, existing processes continue to run, but subsequent requests are returned with a 403 error, and you must obtain a new token.

2. Retrieve the model endpoint by selecting **Actions** > **Copy Model Endpoint** for a running model.

3. Make a prediction call.

   Using the model endpoint, an auth token, and data, make a prediction call on the deployed model.

   For example:

```
curl --cookie "authservice_session=xxxxxxxxxxxxxxxxxxxxxxxxxxx" \
-X POST -H 'Content-Type: application/json' -d '{"data": {"ndarray":
["This film has great actors"]}}' \
http://localhost:8003/seldon/seldon/movie/api/v1.0/predictions
```

   A general template for a prediction call using `curl` is as follows:

```
curl --cookie "authservice_session=<auth-token>" -X POST \
-H 'Content-Type: application/json' -d <data> <access-point>
```

   The `<data>` in this call is a JSON representation of an array or a dataframe. For example, the following represents an array:

```
-d '{"data": {"ndarray":[[39, 7, 1, 1, 1, 1, 4, 1, 2174, 0, 40, 9]]}}'
```

   When the data contains a names field, a dataframe is assumed. The type of predict call (dataframe or array) depends on the input requirement for the model artifact.

   For more information about the Seldon prediction call, see External Prediction API in the official Seldon Core documentation (link opens an external site in a new browser tab or window).

Deleting Deployed Models

This topic describes deleting deployed models in HPE Ezmeral Runtime Enterprise deployments that implement HPE Ezmeral ML Ops.

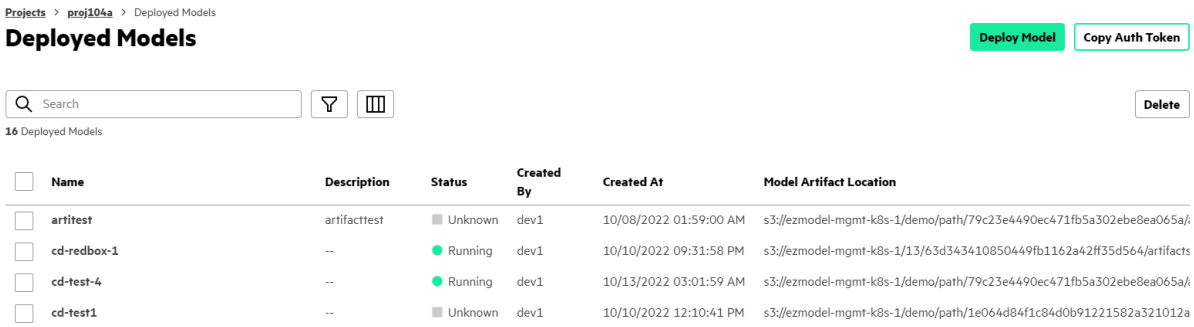
Prerequisites

Required access rights: Project Administrator or Project Member

Procedure

1. Navigate to the project in the new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. Select **View All** on the **Deployed Models** panel. The **Deployed Models** screen opens.



3. Perform one of the following:

   - Select the check box next to the name of the model that you want to delete. Select **Delete**.
   - Open the **Actions** menu next to the model that you want to delete, and select **Delete**.

Model Management APIs

This topic describes Model Management APIs in HPE Ezmeral ML Ops.

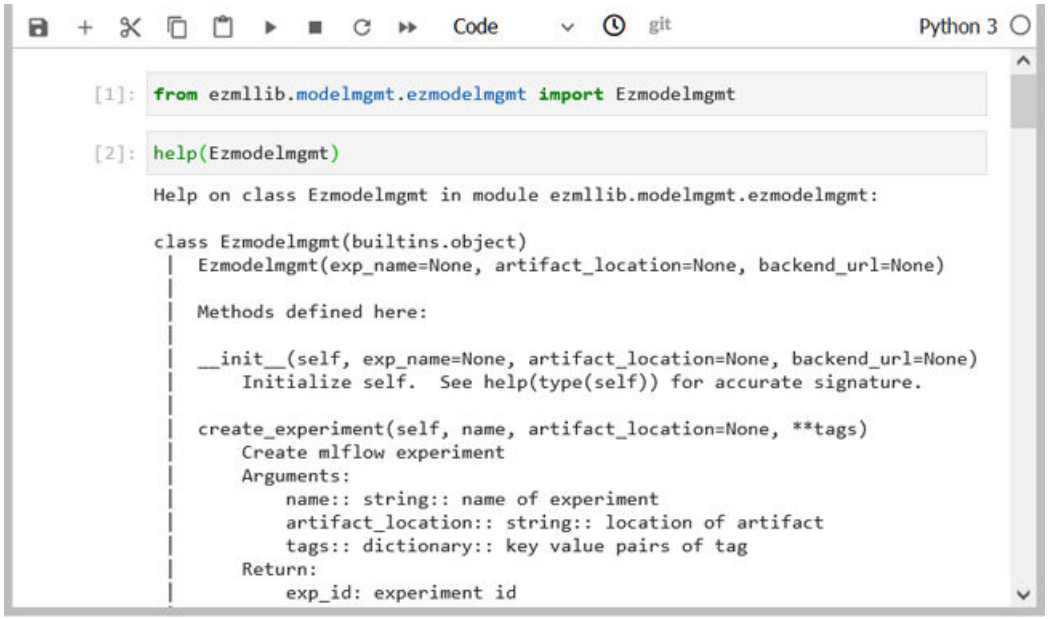
The model management APIs are a set of functions in a module of the `ezmllib` library:

```
ezmllib.modelmgmt.ezmodelmgmt.Ezmodelmgmt
```

To use the functions, enter the following command in a notebook cell. The notebook must be a Python 3 notebook.

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
```

For example:



You can use the following command to display the source code with color coding of the different syntax elements.

```
?? Ezmodelmgmt
```

For example:

```
[4]: ?? Ezmodelmgmt

Init signature:  Ezmodelmgmt(exp_name=None, artifact_location=None, backend
_url=None)
Docstring:      <no docstring>
Source:
class Ezmodelmgmt(object):


    def __init__(self, exp_name=None, artifact_location= None, backend_url=
None ):
        self.exp_name = exp_name
        self.header = utils.get_header()
        self.backend_url = backend_url if backend_url != None else utils.ge
t_modelmgmt_backend_url()
        #create exp or if exp exists set exp
        if not self.backend_url:
            print(f"Please mount a {MODEL_MGMT_SECRET_LABEL} secret to the
notebook, or explicitly provide a backend url with the backend_url kwarg")
            return

        if exp_name is not None and exp_name != "":
            exp_response = self.get_experiment_by_name(exp_name)
```

Notebook `ezmllib` Functions

HPE Ezmeral ML Ops on Kubernetes in HPE Ezmeral Runtime Enterprise provides a specialized function library to streamline various Machine Learning (ML) and Spark pipeline operations in Jupyter notebooks. This library, `ezmllib`, has several modules.

The `ezmllib` library is a packaged library that streamlines your notebook-based coding experience through built-in Python functions.

Utility Package Modules

The library contains the following modules:

`kubeconfig`	The `kubeconfig` modules contain functions to set up kubectl access from the notebook, to add users to an existing Kubeflow user secret, and to get information about the user and secret.
`kubeflow`	The `kubeflow` modules contain functions to configure a Kubeflow environment in a notebook session, to print Kubeflow job logs in a notebook, and to manage PyTorch and TensorFlow jobs.
`Ezmodelmgmt`	The `Ezmodelmgmt` modules contain functions related to model management. Functions in this module import and set environment variables, create a secret populated with S3 credentials, create model endpoints, register models, and register experiments.
`spark`	The `spark` modules contain functions to manage Spark jobs. For more information about using this module, see Spark on Kubernetes on page 243.
`storage`	The `storage` modules contain utilities for generating secrets and for uploading and downloading objects from the Amazon S3 object storage service.

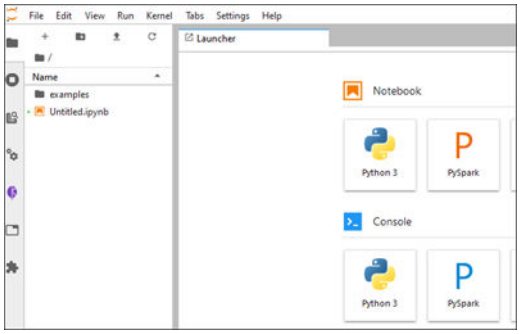
last-updated: May 06, 2024

`tensorflow`	The `tensorflow` modules contain functions to configure a TensorFlow environment in a notebook session.

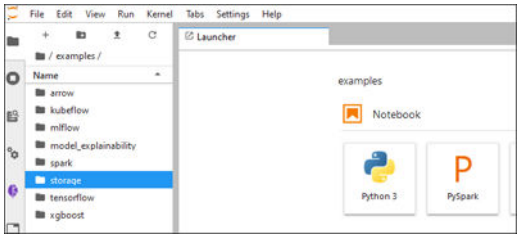
For reference information about the `ezmllib` library, see ezmllib on page 192

Finding Examples in the Notebook

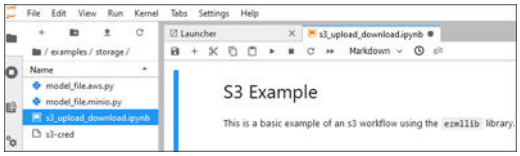
When you create a notebook in HPE Ezmeral Runtime Enterprise, the notebook includes several examples, located in the `examples` folder.



The examples are organized similarly to the `ezmllib` modules.



Explore the examples to see how `ezmllib` functions might be used.



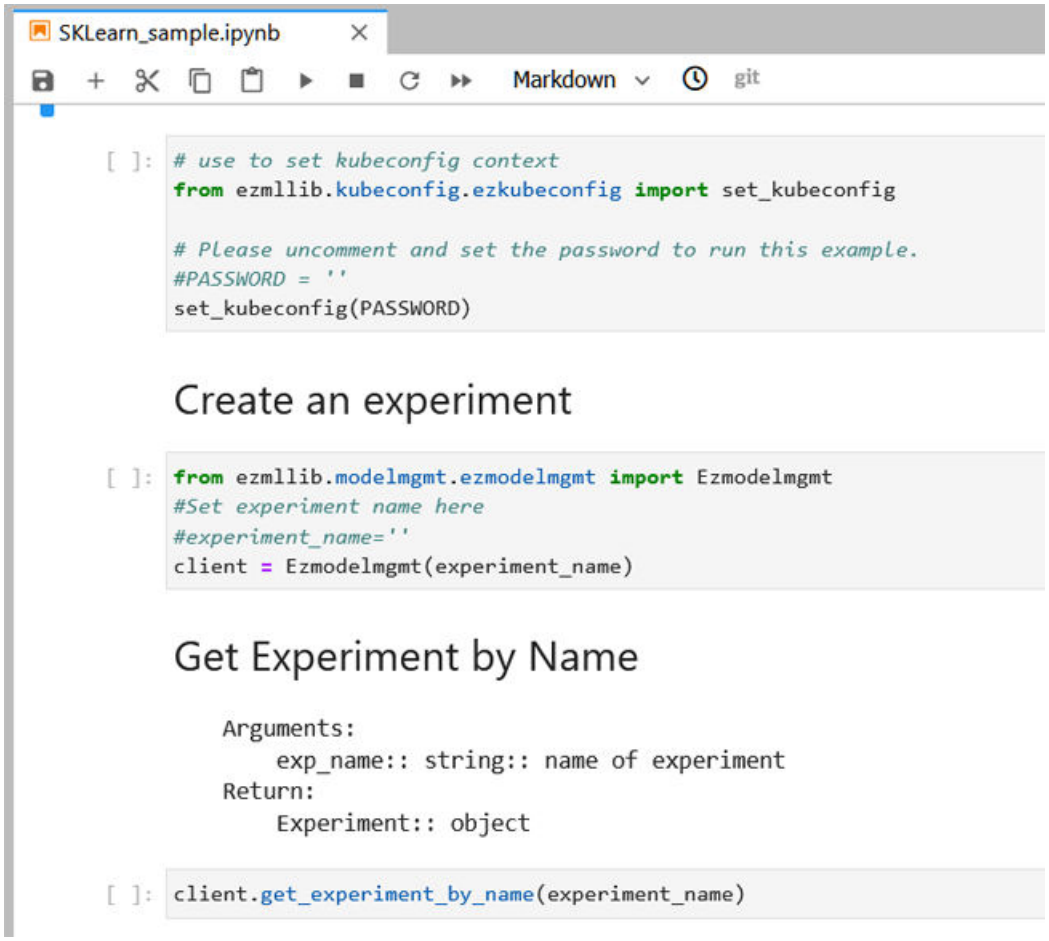
Equivalent Magic and `ezmllib` Functions

Some magic functions provided by Hewlett Packard Enterprise have an equivalent function in the `ezmllib` library.

Magics are deprecated. Where possible, use the `ezmllib` functions instead of the equivalent magics.

Example: Calling Functions In a Notebook

The following example shows importing a module and calling functions in a notebook.



ezmllib

This documentation describes the HPE Ezmeral ML Ops Notebook Python library:

`ezmllib` version 0.3

The `ezmllib` library streamlines your notebook-based coding experience through built-in Python functions. In some cases, these built-in functions are intended as replacements for the equivalent notebook magic functions supplied by Hewlett Packard Enterprise.

Kubeconfig

These modules contain functions to set up kubectl access from the notebook, to add users to an existing Kubeflow user secret, and to get information about the user and secret.

Example:

```
from ezmllib import kubeconfig
```

ezkubeconfig

This module contains functions to set up kubectl access from the notebook and to add users to an existing Kubeflow user secret.

Example:

```
from ezmllib.kubeconfig import ezkubeconfig
```

set_kubeconfig

This function sets up kubeconfig for current user. The function makes an API call to HPE Ezmeral Runtime Enterprise and gets the latest kubeconfig file.

Syntax:

```
set_kubeconfig(pwd)
```

Parameters:

- **pwd** : The HPE Ezmeral Runtime Enterprise password of the user.

Returns:

None

Example:

```
from ezmllib.kubeconfig import ezkubeconfig
ezkubeconfig.set_kubeconfig("password")
```

set_local_kubeconfig

This function sets up kubeconfig from a user-uploaded kubeconfig file. The file must be present in the following notebook directory: `/home/{user}/kubeconfig`

Syntax:

```
set_local_kubeconfig()
```

Returns:

None

Example:

```
from ezmllib.kubeconfig import ezkubeconfig
ezkubeconfig.set_local_kubeconfig()
```

Kubeflow

These modules contain functions to configure a Kubeflow environment in a notebook session, to print Kubeflow job logs in a notebook, and to manage PyTorch and TensorFlow jobs.

Example:

```
from ezmllib import kubeflow
```

ezkflog

This module provides an interface for accessing Kubeflow job logs from the notebook session.

Example:

```
from ezmllib.kubeflow import ezkflog
```

logs

This function prints out the specified Kubeflow job logs in the notebook.

Syntax:

```
logs(name, events=False, status=False, **kwargs)
```

Parameters:

- **name** : Kubeflow job name.
- **events**: [True,False]: Display Kubernetes job events for the Kubeflow job.
- **status**: [True,False]: Display Kubernetes status of the Kubeflow job.
- **`**kwargs`**:
  - **follow**: [True,False]: Stream the logs.
  - **since**: ["10m","30m","1h", ...]: time period to fetch logs from.
  - **previous**: [True,False]: When True, prints the logs for the previous instance of the container in the pod, if available. For example, you can specify True to get the logs of the container for a job that failed.

```
from ezmllib.kubeflow.ezkflog import logs
logs("test",events=True, status=True, follow=True, since="1h",
previous="False")
```

ezkfp

This module provides a Kubeflow session client for interfacing with Kubeflow from a notebook.

Example:

```
from ezmllib.kubeflow import ezkfp
```

`KfSession`

Kubeflow session class

Syntax:

```
KfSession()
```

Parameters:

The constructor takes the following `**kwargs`:

- **user**: (Optional) HPE Ezmeral Runtime Enterprise user.
- **password**: (Optional) HPE Ezmeral Runtime Enterprise password of the user.
- **url**: (Optional) URL of the Kubeflow API.
- **certs**: Path to CA certificate. Required if using an https enabled system, optional if using http.

If an optional parameter is not provided, it will be set through automation. For example, if the password is not provided, the function prompts the user for the password.

Returns: KfSession object

Example:

```
from ezmllib.kubeflow.ezkfp import KfSession
K = KfSession()
```

`kf_client`

This function creates returns an object for interacting with the Kubeflow Pipeline API. The function checks if the user has a `.kubeflow` directory that contains the file `kf.json`. The `kf.json` file contains the Kubeflow endpoint, the Kubeflow session, and certs location. If the user's `.kubeflow` directory exists, the function reads the session cookies from that directory, and returns the client object. If not, the function creates the directory and file, creates and stores session cookies and the Kubeflow endpoint in the file, and then returns the client object. **Syntax:**

```
kf_client(recreate=False)
```

Parameters:

- **recreate**: [True,False]: When `True`, the function overwrites the user's existing `.kubeflow` directory with the directory the function creates. Default value: `False`

Returns:

Returns a `kfp.Client` object for interacting with the Kubeflow Pipeline API

Example:

```
from ezmllib.kubeflow.ezkfp import KfSession
K = KfSession()
client = K.kf_client()
```

MLflow

These modules contain functions related to using MLflow for model management. Functions in this module import and set environment variables, create a secret populated with S3 credentials, create model endpoints, register models, and register experiments.

ezmlflow

This module contains the MLflow automation functions.

Example:

```
from ezmllib.mlflow import ezmlflow
```

`load_mlflow`

This function imports and sets environment variables for MLflow. The function then creates a secret that is populated with the s3 credentials and endpoint for use by Prism. Execute this function prior to using MLflow in a KD Notebook. The mlflow secret must be attached to the notebook application to load successfully.

Syntax:

```
load_mlflow()
```

Returns: None

Example:

```
from ezmllib.mlflow import load_mlflow
load_mlflow()
```

`register_model`

This function registers a model for deployment and creates a Kubernetes ConfigMap with the details. The function requires the model artifact path and the name to be registered.

Syntax:

```
register_model(modelname, modelpath, description_url="")
```

Parameters:

- **modelname**: Name of model to be registered.
- **modelpath**: Path to model artifact.
- **description_url**: (Optional) String containing a description of the model.

Returns:

None

Example:

```
from ezmllib.mlflow import register_model
register_model("test-model","s3://mlflow/example/path/to/artifacts/
model","Example model")
```

set_exp

This function registers the specified experiment in the MLflow tracking service.

Syntax:

```
set_exp(exp_name)
```

Parameters:

- **exp_name**: Name of the experiment to be registered.

Returns:

None

Example:

```
from ezmllib.mlflow import set_exp
set_exp("demo experiment")
```

logs

This function prints out MLflow job logs in the notebook.

Syntax:

```
logs(job_id, training_engine_name)
```

Parameters:

- **job_id**: ID of MLflow job.
- **training_engine_name**: Training engine cluster name.

Returns:

Logs from the specified MLflow job

Example:

```
from ezmllib.mlflow import logs
logs("<ID>", "training-engine-instance")
```

Model

These modules automate the management of model prediction, registry, and deployment.

Example:

```
from ezmllib import model
```

ezkdmodel

This module contains functions related to HPE Ezmeral ML Ops model management, including registering and deploying models and making predictions. The model registry and deployment functions provide an alternative to doing these tasks through the HPE Ezmeral Runtime Enterprise GUI.

Example:

```
from ezmllib.model import ezkdmodel
```

`predict`

This function makes model predictions and returns a string of data.

Syntax:

```
predict(modelName, modelVersion, data, deployment_service)
```

Parameters:

- **deployment_service**: Name of model deployment service, created under the "Model Serving" tab.
- **modelName**: Name of the model.
- **modelVersion**: Model version number in integer format.
- **data**: Inference data in dictionary format.

Returns:

A response object that contains the model prediction results

Example:

```
from ezmllib.model import predict
predict("ml-inferencing", "test-model", 1, inference-data-dict)
```

`register`

This function registers the model components. A model must be registered before it can be served.

Syntax:

```
register(model_registry_name, model_path, scoring_path, model_version,
model_description="")
```

Parameters:

- **model_registry_name**: the model registry name created to register the model_path, scoring_path, and model_version.

- **model_path**: The model URI. For example: `repo://your/model/path`
- **scoring_path**: The model prediction/scoring script URI. For example: `repo://your/scoring/path`
- **model_version**: The version of the model to serve, in integer format.
- **model_description** (kwarg): Description of the model for display in model registry.

Returns:

None

Example:

```
from ezmllib.model import register
register("test-model","repo://your/model/path"," repo://your/scoring/
path",1,model_description="Example model")
```

`deploy`

This function deploys a model and provides options for controlling resources allocated to the model.

Syntax:

```
deploy(deployment_service,cm_array,sc_array=[],
       dtapenabled="false",
       lb_cpu=LB_CPU,
       lb_memory=LB_MEMORY,
       lb_gpu=LB_GPU,
       lb_cpu_lmt=LB_CPU_LMT,
       lb_memory_lmt=LB_MEMORY_LMT,
       lb_gpu_lmt=LB_GPU_LMT,
       rs_cpu=RS_CPU,
       rs_memory=RS_MEMORY,
       rs_gpu=RS_GPU,
       rs_cpu_lmt=RS_CPU_LMT,
       rs_memory_lmt=RS_MEMORY_LMT,
       rs_gpu_lmt=RS_GPU_LMT,
       description="""`)
```

Parameters:

- **deployment_service**: Name of deployment service or inference cluster to deploy the model with. Enclose the name in quotes.
- **cm_array**: An array containing the list of ConfigMap names to attach to model inference app.
  - Default value: **[]**
- **sc_array** (kwarg): List of secret names to attach to model inference app.
  - Default value: **[]**
- **dtapenabled**: [`"true"`,`"false"`]: When `"true"`, enables DataTap connection for the model inference app.
  - Default value: `"false"`
- **lb_cpu** (Optional): Requested CPU for the inference load balancer. Enclose the value in quotes.
  - Default value: `"2"`
- **lb_memory** (Optional): Requested Memory for the inference load balancer. Enclose the value in quotes.

  - Default value: `"4Gi"`
- **lb_gpu** (Optional): Requested GPU for the inference load balancer. Enclose the value in quotes.
  - Default value: `"0"`
- **lb_cpu_lmt** (Optional): Requested CPU limit for the inference load balancer. Enclose the value in quotes.
  - Default value: `"2"`
- **lb_memory_lmt** (Optional): Requested Memory limit for the inference load balancer. Enclose the value in quotes.
  - Default value: `"4Gi"`
- **lb_gpu_lmt** (Optional): Requested GPU limit for the inference load balancer. Enclose the value in quotes.
  - Default value: `"0"`
- **rs_cpu** (Optional): Requested CPU for the inference REST Server. Enclose the value in quotes.
  - Default value: `"2"`
- **rs_memory** (Optional): Requested Memory for the inference REST Server. Enclose the value in quotes.
  - Default value: `"4Gi"`
- **rs_gpu** (Optional): Requested GPU for the inference REST Server. Enclose the value in quotes.
  - Default value: `"0"`
- **rs_cpu_lmt** (Optional): Requested CPU limit for the inference REST Server. Enclose the value in quotes.
  - Default value: `"2"`
- **rs_memory_lmt** (Optional): Requested Memory limit for the inference REST Server. Enclose the value in quotes.
  - Default value: `"4Gi"`
- **rs_gpu_lmt** (Optional): Requested GPU limit for the inference REST Server. Enclose the value in quotes.
  - Default value: `"0"`
- **description** (kwarg): Description of deployed model, enclosed in quotes. For example: `description="test deployment"`

Returns: None

Example:

```
from ezmllib.model import deploy
cm_array=[]
deploy("ml-inferencing",cm_array,description="test deployment")
```

`register_and_deploy`

This function registers then deploys a model in a KubeDirector inference application, and provides options for limiting resource consumption.

Syntax:

```
register_and_deploy(model_registry_name, model_path, scoring_path,
                    model_version,
                    deployment_service,
                    cm_array,
                    sc_array=[],
                    dtapenabled="false",
                    lb_cpu=LB_CPU,
                    lb_memory=LB_MEMORY,
                    lb_gpu=LB_GPU,
                    lb_cpu_lmt=LB_CPU_LMT,
                    lb_memory_lmt=LB_MEMORY_LMT,
                    lb_gpu_lmt=LB_GPU_LMT,
                    rs_cpu=RS_CPU,
                    rs_memory=RS_MEMORY,
                    rs_gpu=RS_GPU,
                    rs_cpu_lmt=RS_CPU_LMT,
                    rs_memory_lmt=RS_MEMORY_LMT,
                    rs_gpu_lmt=RS_GPU_LMT,
                    description="",
                    model_description="")
```

Parameters:

- **model_registry_name**: The model registry name created to register the model_path, scoring_path, and model_version. Enclose the name in quotes.
- **model_path**: The model URI, enclosed in quotes. For example: `"repo://your/model/path"`
- **scoring_path**: The model prediction/scoring script URI, enclosed in quotes. For example: `"repo://your/scoring/path"`
- **model_version**: The version of the model to serve, in integer format.
- **deployment_service**: Name of deployment service or inference cluster to deploy the model with. Enclose the name in quotes.
- **cm_array**: An array containing the list of ConfigMap names to attach to model inference app.
  - Default value: **[]**
- **sc_array** (kwarg): List of secret names to attach to model inference app.
  - Default value: **[]**
- **dtapenabled**: [`"true"`,`"false"`]: When `"true"`, enables DataTap connection for the model inference app.
  - Default value: `"false"`
- **lb_cpu** (Optional): Requested CPU for the inference load balancer. Enclose the value in quotes.
  - Default value: `"2"`
- **lb_memory** (Optional): Requested Memory for the inference load balancer. Enclose the value in quotes.
  - Default value: `"4Gi"`
- **lb_gpu** (Optional): Requested GPU for the inference load balancer. Enclose the value in quotes.
  - Default value: `"0"`

- **lb_cpu_lmt** (Optional): Requested CPU limit for the inference load balancer. Enclose the value in quotes.
  - Default value: `"2"`
- **lb_memory_lmt** (Optional): Requested Memory limit for the inference load balancer. Enclose the value in quotes.
  - Default value: `"4Gi"`
- **lb_gpu_lmt** (Optional): Requested GPU limit for the inference load balancer. Enclose the value in quotes.
  - Default value: `"0"`
- **rs_cpu** (Optional): Requested CPU for the inference REST Server. Enclose the value in quotes.
  - Default value: `"2"`
- **rs_memory** (Optional): Requested Memory for the inference REST Server. Enclose the value in quotes.
  - Default value: `"4Gi"`
- **rs_gpu** (Optional): Requested GPU for the inference REST Server. Enclose the value in quotes.
  - Default value: `"0"`
- **rs_cpu_lmt** (Optional): Requested CPU limit for the inference REST Server. Enclose the value in quotes.
  - Default value: `"2"`
- **rs_memory_lmt** (Optional): Requested Memory limit for the inference REST Server. Enclose the value in quotes.
  - Default value: `"4Gi"`
- **rs_gpu_lmt** (Optional): Requested GPU limit for the inference REST Server. Enclose the value in quotes.
  - Default value: `"0"`
- **description** (kwarg): Description of deployed model, enclosed in quotes. For example: `description="test deployment"`
- **model_description** (kwarg): Description for the model for display in model registry, enclosed in quotes. For example: `model_description="Example model"'

Returns: None

Example:

```
from ezmllib.model import register_and_deploy
cm_array = []
register_and_deploy("test-model","repo://your/model/path",
                    "repo://your/scoring/path",1,
                    "ml-inferencing",cm_array,description="test
deployment",
                    model_description="Example model")
```

get_inference_app_details

This function gets inference app details of the specified the inference app.

Syntax:

```
get_inference_app_details(kd_inference_app_name)
```

Parameters:

- **kd_inference_app_name**: Name of the model inference service (name of app created by "Model Serving" tab).

Returns: Dictionary containing "Inference App State", "Message", and "Service URL" as keys with relevant details.

Example:

```
from ezmllib.model import get_inference_app_details
get_inference_app_details("test-inference-app")
```

update_registry

This function updates the existing model registry.

Syntax:

```
update_registry(context, modelname, modelpath=None, scoringpath=None)
```

Parameters:

- **context**: Name of current context as listed in kubeconfig. Enclose the name in quotes.
- **modelname**: Registered name of model. Enclose the value in quotes.
- **modelpath**: (Optional) URI for model artifacts. Enclose the value in quotes.
- **scoringpath**: (Optional) URI for model prediction or scoring script. Enclose the value in quotes.

Returns: None; Prints message indicating success or failure to update

Example:

```
from ezmllib.model import update_registry
update_registry("ECP-TEST-compute-mlops-dev1", "test-model")
```

Spark

These modules contain functions to manage Spark jobs.

ezspark

This module contains functions to manage Spark jobs.

Example:

```
from ezmllib.spark import ezspark
```

submit

This function submits Spark jobs with inputs, or with a provided yaml file.

Syntax:

```
submit(
    app_path=None,
    data_path=None,
    yaml_path=None,
    name=None,
    image_name='gcr.io/mapr-252711/spark-py-3.1.2:202111021109R',
    driver_cores=1,
    driver_memory='512m',
    driver_core_limit='1000m',
    executor_cores=1,
    executor_instances=2,
    executor_memory='512m',
    executor_core_limit='1000m',
    spark_version='3.1.2',
    python_version='3',
    app_type='Python',
    api_version='sparkoperator.hpe.com/v1beta2',
    kind='SparkApplication',
    namespace=None,
)
```

Parameters:

- **app_path**: (Optional) Path to application file for the Spark job. Enclose the path in quotes. For example: `app_path="local:///opt/mapr/spark/spark-2.4.7/examples/src/main/python/wordcount.py"` - Default value: `None`.
- **data_path**: (Optional) Path to data file for the Spark job. Enclose the path in quotes. For example: `data_path="dtap://TenantStorage/data/wordcount.txt"` - Default value: `None`.
- **yaml_path**: (Optional) Path to yaml file for the Spark job. Enclose the path in quotes. - Default value: `None`.
- **name**: (Optional) Name of the Spark job. Enclose the name in quotes. For example: `name="mysparkjob1"` - Default value: `None`.
- **image_name**: Spark image name.
  - Default value: `'gcr.io/mapr-252711/spark-py-3.1.2:202111021109R'`
- **driver_cores**: Number of CPU cores to allocate to the Spark driver.
  - Default value: `1`
- **driver_memory**: Allocated memory for the Spark driver. - Default value: `'512m'`
- **driver_core_limit**: Maximum allowed CPU cores for Spark driver.
  - Default value: `'1000m'`
- **executor_cores**: Allocated CPU cores for Spark executor.
  - Default value: `1`
- **executor_instances**: Number of Spark executor instances.
  - Default value: 2
- **executor_memory**: Memory request for the Spark executor.
  - Default value: `'512m'`

- **executor_core_limit**: Maximum CPU cores allotted to the Spark executor.
  - Default value: `'1000m'`
- **spark_version**: Version of Spark to use in the Spark job.
  - Default value: `'3.1.2'`
- **python_version**: Version of Python to use in the Spark job.
  - Default value: `'3'`
- **app_type**: Spark application type.
  - Default value: `'Python'`
- **api_version**: Spark API version to be used.
  - Default value: `'sparkoperator.hpe.com/v1beta2'`
- **kind**
  - Default value: `'SparkApplication'`
- **namespace**: (Optional) Namespace in which to execute the job.
  - Default value: `None`. If `None`, deploys to the notebook namespace.

Returns: None; Prints output of job configuration and status of its creation

Example:

```
from ezmllib.spark import submit
submit(app_path="local:///opt/mapr/spark/spark-3.1.2/examples/src/main/
python/wordcount.py",
       data_path="dtap://TenantStorage/data/wordcount.txt",
       name="test1"
      )
```

`delete`

This function deletes the Spark job specified by job name. If no name is provided, lists all Spark jobs and prompts user to list the Spark job names to be deleted, in space-delimited format. For example: `name1 name2 name3`

Syntax:

```
delete(name)
```

Parameters:

- **name**: String containing the job name.

Returns: None; Prints message indicating deletion status of the job

Example:

```
from ezmllib.spark import delete
delete("test1")
```

`logs`

This function displays log messages from the specified Spark job.

Syntax:

```
logs(name, events=False, **kwargs)
```

Parameters:

- **name**: Spark job name.
- **events**: [True,False]: When `True`, includes Kubernetes event logs.
- **`**kwargs`**:
  - **follow**: [True,False]: When `True`, streams the logs
  - **since** ["10m", "30m", "1hr", ...etc.]: Timeframe from which to fetch logs.
  - **tail**: ["10","100", ...etc.]: Provides the last N log messages.
  - **previous**: [True,False]: When `True`, prints the logs for the previous instance of the container in the pod, if available. For example, you can specify True to get the logs of the container for a job that failed.

Returns: None; Prints spark job logs as notebook cell output

Example:

```
from ezmllib.spark import logs
logs("test1")
```

Storage

These modules contain a utility for generating secrets and modules for uploading and downloading objects from the Amazon S3 object storage service.

Example:

```
from ezmllib import storage
```

ezs3

This module provides a client interface for easily moving files between a notebook session and s3 storage.

Example:

```
from ezmllib.storage import ezs3
```

`s3_util`

This is a class that provides s3 upload and download functions. This class automatically reads s3 storage credentials from the secret in Kubernetes environment. Users can provide these credentials manually through optional parameters in functions that accept `**kwargs`.

Syntax:

```
s3_util(src, dest)
```

Parameters:

- **src**: Source path. Specifies where file is located.
- **dest**: Destination path. Specifies where file is to be uploaded to or downloaded to.

Returns: `s3_util` object

Example:

```
from ezmllib.storage.ezs3 import s3_util
u = s3_util(local_path, aws_s3_path)
```

`upload`

This function uploads file to `self.dest` of the `s3_util` client.

Syntax:

```
upload()
```

Parameters:

To use different credentials than the credentials stored in the `s3_secret`, specify the following optional parameters to be passed as `**kwargs`:

- **aws_access_key_id**: (Optional) Remote storage username credential.
- **aws_secret_access_key**: (Optional) Remote storage password credential.

Returns: None

Example:

```
from ezmllib.storage.ezs3 import s3_util
u = s3_util(local_path, aws_s3_path)
u.upload()
```

`download`

This function downloads a file to: `self.**dest**`

Syntax:

```
download()
```

Parameters: To use different credentials than the credentials stored in the `s3_secret`, specify the following optional parameters to be passed as `**kwargs`:

- **aws_access_key_id**: (Optional) Remote storage username credential.
- **aws_secret_access_key**: (Optional) Remote storage password credential.

Returns: None

Example:

```
from ezmllib.storage.ezs3 import s3_util
u = s3_util(aws_s3_path, local_path)
u.download()
```

`create_s3_secret`

Given the provided credentials, this function creates a secret in the user's namespace to store the remote storage access key and ID. The `s3_util` reads the security parameters from the secret to facilitate the file transfer.

Syntax:

```
create_s3_secret(s3_id, s3_key)
```

Parameters:

- **s3_id**: Username for the remote storage.
- **s3_key**: Password for the remote storage.

Returns: None

Example:

```
from ezmllib.storage.ezs3 import create_s3_secret
AWS_S3_ACCESS_KEY_ID = "user"
AWS_S3_SECRET_ACCESS_KEY = "pass"
create_s3_secret(AWS_S3_ACCESS_KEY_ID,AWS_S3_SECRET_ACCESS_KEY)
```

TensorFlow

These modules contain functions to configure a TensorFlow environment in a notebook session.

Example:

```
from ezmllib import tensorflow
```

eztensorflow

This module provides a convenient interface for TensorFlow in a notebook environment.

Example:

```
from ezmllib.tensorflow import eztensorflow
```

`config`

This function configures the TensorFlow environment for this notebook session. The function configures TensorFlow to use GPU hardware, if available, and to specify whether to use TensorFlow version 2.

Syntax:

```
config(tf, v2=False)
```

Parameters:

- **tf**: TensorFlow library object.
- **v2**: [True,False]: Specifies whether to use TensorFlow version 2. Default value: `False`

Returns: A TensorFlow library object for use in the notebook

Example:

```
import tensorflow as tf
import ezmllib
tf = ezmllib.tensorflow.config(tf, v2=True)
```

modelmgmt

These modules interface with the model management service from within a notebook instance

Example:

```
from ezmllib import modelmgmt
```

ezmodelmgmt

Ezmodelmgmt provides an client for easily managing model experiments, runs, and their artifacts.

Example:

```
from ezmllib.modelmgmt import ezmodelmgmt
```

`Ezmodelmgmt`

This is a class that provides model management functions for experiments, runs, and artifacts. This class automatically reads s3 storage credentials and model management details from the model management secret in Kubernetes environment. This secret is required for Ezmodelmgmt to function.

Syntax:

```
Ezmodelmgmt(exp_name=None, artifact_location= None, backend_url=None)
```

Parameters:

- **exp_name**: Name of the experiment the client will manage. Client will notify if name is already used
- **artifact_location**: (Optional) Path to store all artifacts to for experiment. Default used if not provided
- **backend_url**: (Optional) URL of model management backend service for client to connect to

Returns: `Ezmodelmgmt` object

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
#Set experiment name here
#experiment_name=''
client = Ezmodelmgmt(experiment_name)
```

`create_experiment`

This function updates the client to use a new experiment created by the model management APIs

Syntax:

```
create_experiment(name, artifact_location=None, **tags)
```

Parameters:

- **name**: String containing name of experiment to be created
- **artifact_location**: (Optional) Path to store all artifacts to for experiment. Default used if not provided
- **tags**: (Optional) Dictionary of key-value pairs to tag the experiment with

Returns: Experiment ID; Prints API response

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name_1)
exp_id = client.create_experiment(experiment_name_2)
```

start_run

Starts/creates mlflow run for the experiment set in the client object

Syntax:

```
start_run(tags)
```

Parameters:

- **tags**: (Optional) Dictionary of key-value pairs to tag the run with

Returns: Run ID; Prints API response

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
run_id = client.start_run()
```

log_param

Log a parameter for a run

Syntax:

```
log_param(key, value, run_id = None)
```

Parameters:

- **key**: String with key name for parameter
- **value**: Float with parameter value
- **run_id**: (Optional) String indicating which run to log the parameter to. Uses existing run if not provided, or creates one.

Returns: None; Prints API response

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.log_param("learning_rate",0.5)
```

log_metric

Log a metric for a run

Syntax:

```
log_metric(key, value, run_id = None)
```

Parameters:

- **key**: String with the key name for the metric
- **value**: Float with value of metric
- **run_id**: (Optional) String indicating which run to log the metric to. Uses existing run if not provided, or creates one.

Returns: None; Prints API response

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.log_metric("root_mean_squared_error", 0.85)
```

log_run_tag

Log a tag for a run

Syntax:

```
log_run_tag(key, value, run_id = None)
```

Parameters:

- **key**: String with the key name for tag
- **value**: String with value of tag
- **run_id**: (Optional) String indicating which run to log the tag to. Uses existing run if not provided, or creates one.

Returns: None; Prints API response

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.log_run_tag("key", "value")
```

get_run

Get run details by run_id

Syntax:

```
get_run(run_id = None)
```

Parameters:

- **run_id**: (Optional) String specifying run by run_id. Uses existing run if not provided, or creates one.

Returns: API Response Object

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.get_run()#gets current run details
client.get_run("e8fda241508f4658a88762a3c37f311c") # gets details of
specified run if user has access
```

get_runs

Get all run details for a set of experiments by their experiment ID

Syntax:

```
get_run(exp_ids=[])
```

Parameters:

- **exp_ids**: (Optional) List of Strings specifying experiment by exp_id. Uses client experiment ID if not provided.

Returns: API Response Object

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.get_runs()#gets current experiment details
client.get_runs(["1","2","3"]) # gets details of specified experiments if
user has access
```

`delete_run`

Soft delete of experiment run by run id

Syntax:

```
delete_run(run_id = None)
```

Parameters:

- **run_id**: (Optional) String specifying run by run_id. Uses current active run if not provided

Returns: API Response Object

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.delete_run("e8fda241508f4658a88762a3c37f311c") #delete specific run
client.delete_run() # delete active run
```

`restore_run`

Restore a deleted experiment run by run id

Syntax:

```
delete_run(run_id = None)
```

Parameters:

- **run_id**: (Optional) String specifying run by run_id. Uses most recent cached run if not provided

Returns: API Response Object

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.restore_run("e8fda241508f4658a88762a3c37f311c") #restore specific run
client.restore_run() # restore active run
```

`list_experiment`

List all experiments of the notebook user

Syntax:

```
list_experiment()
```

Parameters:

- N/A

Returns: API Response Object

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.list_experiment()
```

get_experiment

Get experiment details for a specific experiment by ID

Syntax:

```
get_experiment(exp_id = None)
```

Parameters:

- **exp_id**: (Optional) String specifying experiment by exp_id. Uses client's experiment if not specified

Returns: API Response Object

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.get_experiment() # gets the experiment defined by "experiment_name"
client.get_experiment("42") # gets the experiment with exp ID 42
```

set_exp_tag

Tag an experiment with a key-value pair

Syntax:

```
log_run_tag(key, value, exp_id = None)
```

Parameters:

- **key**: String with the key name for tag
- **value**: String with value of tag
- **exp_id**: (Optional) String specifying experiment by exp_id. Uses client's experiment if not specified

Returns: None; Prints API response

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.set_exp_tag("key", "value")
```

log_artifacts

Log a directory of files for a specific run

Syntax:

```
log_artifact(folder_location, run_id=None ):
```

Parameters:

- **folder_location**: String with full path of folder to be logged, e.g. `/home/<user>/test_results`
- **run_id**: (Optional) String specifying run by run_id. Uses most recent cached run if not provided

Returns: None; Prints message on failure

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.log_artifacts("/home/user1/test_results")
```

`log_artifact`

Log a file for a specific run

Syntax:

```
log_artifact(file_location, run_id=None ):
```

Parameters:

- **file location**: String with full path of file to be logged, e.g. `/home/<user>/data.csv`
- **run_id**: (Optional) String specifying run by run_id. Uses most recent cached run if not provided

Returns: None; Prints message on failure

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.log_artifact("/home/user1/data/data.csv")
```

`log_model`

Log a model object and its specification to a specific run

Syntax:

```
log_model(model=None,flavor=None,run_id=None,registered_model_name=None):
```

Parameters:

- **model**: Model object to be logged
- **flavor**: (Optional) Framework used to create model. If not provided, will dynamically detect
  - Supported flavors:
    - sklearn
    - keras
    - xgboost
    - pytorch
    - h2o
    - spark
    - tensorflow

Note: Custom models defined with TF or PyTorch primitives *must* specify flavor. Client will prompt user to if it is not provided as an argument. Ex: `client.log_model(model=custom_model, flavor='pytorch'...)`

- **run_id**: (Optional) String specifying run by run_id. Uses most recent cached run if not provided
- **registered_model_name**: (Optional) Name to register the model with

Returns: None; Prints message on failure

Example:

```
from ezmllib.modelmgmt.ezmodelmgmt import Ezmodelmgmt
client = Ezmodelmgmt(experiment_name)
client.log_artifact("/home/user1/data/data.csv")
```

Notebook Magic Functions

Jupyter notebook magic functions, also known as magics, are special commands that provide notebook functions that might not be easy for you to program using Python. HPE Ezmeral ML Ops on Kubernetes in HPE Ezmeral Runtime Enterprise supports line magics and cell magics.

Jupyter notebook **Magic functions**, also known as **magic commands** or **magics**, are commands that you can execute within a code cell. Magics are not Python code. They are shortcuts that extend a notebook's capabilities. Magic commands start with the `%` character.

HPE Ezmeral ML Ops on HPE Ezmeral Runtime Enterprise supports built-in magic functions and the custom magics that are described in this topic. HPE Ezmeral ML Ops on Kubernetes in HPE Ezmeral Runtime Enterprise supports line magics and cell magics.

NOTE:

Magic functions are deprecated. Support for magics will be discontinued in a future release.

The `ezmllib` library is a packaged library that streamlines your notebook-based coding experience through built-in Python functions. This library has built-in functions are intended as replacements for the equivalent notebook magic commands.

Where possible, use the `ezmllib` functions instead of the equivalent magics. See Notebook ezmllib Functions on page 190.

Line magic commands do not require a cell body and start with a single `%` character. Cell magic commands start with `%%` and require additional lines of input (a cell body).

Some magic commands require a password. After you have supplied the password for one of these commands in the notebook, the password you supply is automatically applied to other commands in the notebook that require a password, so you do not have use the `--pwd` argument in subsequent commands. You can override the saved password, if needed, by specifying the `--pwd` argument in the command.

Listing Available Magics

The `%lsmagic` command lists all the magic commands available to this notebook. The commands are grouped by line magics and then by cell magics.

```
[1]: %lsmagic

[1]: ▾ root:
       ▸ line:
       ▾ cell:
           js: "DisplayMagics"
           javascript: "DisplayMagics"
           latex: "DisplayMagics"
```

Getting Help

To display help about a magic command, enter the command followed by a ? (question mark). For example:

```
%%kubeRefresh?
```

%kubeRefresh

Execute this magic if a user-specific `kubeconfig` secret was attached to the Notebook cluster before launching. This magic should be executed before entering `kubectl` commands from the notebook cell.

The syntax is:

```
%kubeRefresh [--local_kubeconfig {true|false}] [--pwd PWD]
```

Executing the magic without the `--pwd` argument generates an interactive request for the user password, unless you have already supplied the password in a previous command in this notebook.

If you want to use a local `kubeconfig` file instead of the default `kubeconfig` file, upload the file to the `/home/{user}/kubeconfig` directory and then use the `--local_kubeconfig true` argument.

The magic then obtains the appropriate `kubeconfig` file, and you may begin executing `kubectl` commands.

For example:

```
[2]: %kubeRefresh --local_kubeconfig true
     Kubeconfig file not found. Please add config file under directory /home/dev1/kubeconfig/

[3]: %kubeRefresh --local_kubeconfig true
     Kubeconfig is available to use
```

In HPE Ezmeral Runtime Enterprise 5.4.0 and later, this magic has an equivalent function in the `ezmllib` library. See Notebook ezmllib Functions on page 190.

%setLivy

You can use the `%setLivy` magic to connect to a different Livy session.

Syntax:

```
%setLivy --url URL [--pwd PWD]
```

For example:

```
%setLivy --url http://mycorp.net:10029
```

The `--url` argument specifies the Livy endpoint to which you want to connect.

Executing the magic without the `--pwd` argument generates an interactive request for the user password, unless you have already supplied the password in a previous command in this notebook. If the Livy session needs authentication, enter the password. If not, press enter.

Tutorials for HPE Ezmeral ML Ops on Kubernetes

For Kubeflow tutorials, see Kubeflow Tutorials on page 218.

For general Kubernetes tutorials and examples, see General Kubernetes Tutorials on page 372.

Tutorial: KubeDirector Training and Serving
Starting with HPE Ezmeral Runtime Enterprise 5.5.0, the KubeDirector training and deployment applications are deprecated, and are unavailable for creation through the HPE Ezmeral ML Ops UI. However, you can still create and use KubeDirector training and deployment applications with YAML files. This tutorial provides instructions on deploying Kubedirector Training and Kubedirector Serving applications with YAML files in HPE Ezmeral ML Ops.

If you have not done so already: Before beginning this tutorial, download the , which contains sample files for all of the included KubeDirector tutorials.

Instantiate a KubeDirector Training Application

Noteworthy fields are as follows:

- `Name`: Name of the training instance.
- `Namepsace`: Name of the Tenant namespace.
- `Spec.connection.secrets`: Tenant secrets that are included. To fetch and update the secret values, use the following command:

  ```
  kubectl -n <namespace> get secrets
  ```

- `Requests/limits`: You can edit this value for CPU, memory, or GPU as needed.
- `Roles.podLabels`: To enable DataTap for all node roles, set:

  ```
  hpecp.hpe.com/dtap: "hadoop2"
  ```

1. After you have populated the YAML based on your cluster configuration, apply the YAML towards your tenant namespace:

   ```
   Kubectl -n <tenant_namespace> apply -f training-instance.yaml
   ```

2. For each training instance, there are two pods. One pod services the APIs, and the other acts as a load balancer. Ensure both pods are in a running state. Once successful, the training instance now appears while creating a notebook cluster.

Instantiate a Kubedirector Model Serving Application

Noteworthy fields are as follows:

- `Name`: Name of the deployment instance.
- `Namepsace`: Name of the Tenant namespace.

- `Spec.connection.secrets`: Tenant secrets that are included. To fetch and update the secret values, use the following command:

  ```
  kubectl -n <namespace> get secrets
  ```

- `Requests/limits`: You can edit this value for CPU, memory, or GPU as needed.
- `Roles.podLabels`: To enable DataTap for all node roles, set:

  ```
  hpecp.hpe.com/dtap: "hadoop2"
  ```

1. After you have populated the YAML based on your cluster configuration, apply the YAML towards your tenant namespace:

   ```
   Kubectl -n <tenant_namespace> apply -f deployment-instance.yaml
   ```

2. For each training instance, there are two pods. One pod services the APIs, and the other acts as a load balancer. Ensure both pods are in a running state. Once successful, the training instance now appears while creating a notebook cluster.

Include the Training Clusters in the Kubedirector Notebook

You can include Training clusters in the Kubedirector notebook to run Training-related magic functions.

1. Log in to the HPE Ezmeral Runtime Enterprise web UI.
2. Select **Dashboard**.
3. Select the HPE Ezmeral ML Ops tenant.
4. Select the notebook and select **launch**.
5. Edit the YAML file and add the training cluster:

```
apiVersion: "kubedirector.hpe.com/v1beta1"
kind: "KubeDirectorCluster"
metadata:
  name: "jupyter-notebook-instance"
  namespace: "aiml1"
  labels:
    description: ""
spec:
  app: "jupyter-notebook"
  appCatalog: "local"
  connections:
    clusters:
      - my-training-engine-instance
      - my-mlflow-instance
    secrets:
      - hpecp-kc-secret-192e81d6d7054551422bb88bdf9f90a3
      - hpecp-sc-secret-33630169f143ac582f69d43ofa3e3669
      - hpecp-ext-auth-secret
      - mlflow-sc
...
```

More information

Tutorial: Transition from KubeDirector to Kubeflow Training on page 218
This tutorial provides a use case to help transition from KubeDirector training and deployment to Kubeflow equivalents.

Kubeflow Tutorials

This section contains Kubeflow tutorials and examples for Kubeflow on HPE Ezmeral Runtime Enterprise

NOTE: Beginning with HPE Ezmeral Runtime Enterprise 5.5.1, Kubeflow notebooks are available. However, Hewlett Packard Enterprise recommends that you use full-featured KubeDirector notebooks instead. See Creating Notebook Servers on page 169.

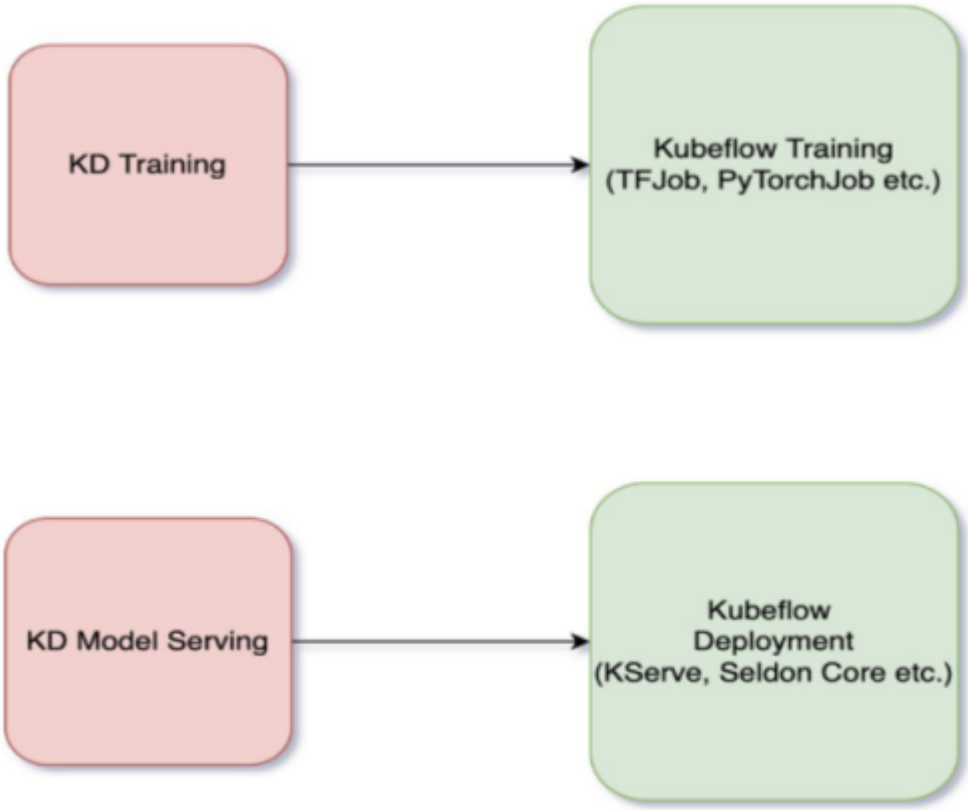
For HPE Ezmeral ML Ops tutorials, see Tutorials for HPE Ezmeral ML Ops on Kubernetes on page 216.

For general Kubernetes tutorials and examples, see General Kubernetes Tutorials on page 372.

Tutorial: Transition from KubeDirector to Kubeflow Training
This tutorial provides a use case to help transition from KubeDirector training and deployment to Kubeflow equivalents.

Prerequisites:

- This tutorial assumes you have an existing KubeDirector notebook cluster up and running.
- Before beginning this tutorial, download the , which contains sample files for all the included KubeDirector tutorials.



Tutorial 1: Transition From KubeDirector Training to TFJob

1. **Provision the KubeDirector Training cluster:**

   a. Run the `training.yaml` file included under `templates`:

   ```
   kubectl apply -f training.yaml -n <tenant>
   ```

   b. Check the provisioning status of the cluster:

   ```
   kubectl get pods -n <tenant> | grep train1
   ```

   c. Run the notebook example `training_sample.ipynb`. This notebook example runs the sample Tensorflow script using the KubeDirector Training cluster.

2. **Run the Tensorflow job:**

   Next, run the same training script using Kubeflow TFJob.

   You can run the scripts for this step using the `tutorial.ipynb` notebook included in both the `tensorflow/KServe` and `tensorflow/Seldon` folders. Select the folder corresponding to the type of inferencing that you want to run.

   The steps in the notebook are explained in detail below.

   a. Create an image that includes the required scripts and relevant datasets from the sample zip file. This image acts as the basis of the TFJob utility. Ensure that the required training and dataset files are available in your local machine.

b. You must have access to a Docker daemon to build and push the created image to a compatible docker registry. To install Docker, see this page in the official Docker documentation (link opens an external site in a new tab or window).

   Ensure you have access to a Docker registry which is accessible from the HPE Ezmeral Runtime Enterprise cluster.

c. Run the scripts using the `tutorial.ipynb` notebook. This notebook is included in both the `KServe` and `Seldon` folders. Select the folder corresponding to the type of inferencing you want to run.

   The steps included in the notebook are as follows:

   1. Create a basic Docker file following the template in Dockerfile. Make sure to include the datasets and the scripts to the image provided in the sample.

   2. After the image is ready, build and push the image to the registry:

      ```
      docker build -t <docker_image_name_with_tag>
      ```

      ```
      docker push <docker_image_name_with_tag>
      ```

      The pushed image now serves as the base image during the training phase.

3. Before beginning training, create a PVC for the saved model:

   a. Open and apply the PVC YAML available as part of the training folder:

      ```
      kubectl apply -f tfjob-pvc.yaml
      ```

   b. Verify that the PVC is created and in a bound state:

      ```
      kubectl get pvc
      ```

4. Apply the TFJob CR YAML to run the training:

   - If you are using **Kserve** inferencing:

     ```
     kubectl -n <namespace> apply -f tfjob_kserve.yaml
     ```

   - If you are using **Seldon** inferencing:

     ```
     kubectl -n <namespace> apply -f tfjob_seldon.yaml
     ```

5. A TFJob is created and a pod is provisioned to run the training. The output of the training is a file that exists in the associated PVC:

   ```
   kubectl get pods -n <namespace> | grep tfjob
   ```

   When the pod enters a complete state, the model building is complete. You can now deploy the generated model with KServe or Seldon. See:

   - Tutorial 3: Inferencing with KServe on page 222
   - Tutorial 4: Inferencing with Seldon Core on page 222

Tutorial 2: Transition From KubeDirector Training to PyTorchJob

This tutorial uses the notebook under `examples/mlflow/PyTorch_sample.ipynb` as an example. Sample scripts for this tutorial are located in the `tutorials/pytorch` folder in the sample zip file.

1. Upload the notebook `PyTorch_sample.ipynb` to your KubeDirector notebook cluster. Familiarize yourself with the script. Then proceed with the following steps to run the same script as a part of Kubeflow PyTorchJob.

2. Create an image that includes the required scripts and relevant datasets from the sample zip file. This image acts as the basis of the PyTorchJob utility. Ensure that the required training and dataset files are available in your local machine.

3. You must have access to a Docker daemon to build and push the created image to a compatible docker registry. To install Docker, see this page in the official Docker documentation (link opens an external site in a new tab or window).

   Ensure you have access to a Docker registry which is accessible from the HPE Ezmeral Runtime Enterprise cluster.

4. Create a basic Docker file following the template in Dockerfile. Make sure to include the required datasets and the scripts to the image.

5. After the image is ready, build and push the image to the registry:

   ```
   docker build -t <docker_image_name_with_tag>
   ```

   ```
   docker push <docker_image_name_with_tag>
   ```

6. The pushed image now serves as the base image during our training phase. Before we start training, ensure storage for the saved model. Create a PVC:

   a. Open and apply the PVC YAML available as part of the training folder:

   ```
   kubectl apply -f pytorch-pvc.yaml
   ```

   b. Verify that the PVC is created and in a bound state:

   ```
   kubectl get pvc
   ```

7. Apply the PyTorch CR YAML to run the training:

   ```
   kubectl -n <namespace> apply -f pytorch.yaml
   ```

8. A PyTorchJob is created and a pod is provisioned to run the training. The output of the training is a model file that exists in the associated PVC:

   ```
   kubectl get pods -n <namespace> | grep  tfjob1-sample-worker-0
   ```

   When the pod enters a complete state, the model building is complete. You can now deploy the generated model with Seldon Core. See: Tutorial 4: Inferencing with Seldon Core on page 222.

Tutorial 3: Inferencing with KServe

1. Obtain the `KServe/inference_kserve.yaml` file from the `tensorflow` directory in the sample zip file.

2. Apply `KServe/inference_kserve.yaml` to the tenant namespace:

```
kubectl apply -f KServe/inference_kserve.yaml -n <namespace>
```

3. Ensure that the pods are up and running. You can track the status of the serving deployment with the following commands:

```
kubectl get inferenceservices
```

```
kubectl get pods | grep tfjob-serving
```

4. After the pods are up and running, send a request to the model.

   Sample requests are available under `tensorflow/kserve/requests_kserve.py`.

   a. In the Jupyter Notebook terminal, install the following Python dependencies:

```
pip install requests lxml --user
```

   b. From the Jupyter notebook, launch `kserving-request.py` as follows:

```
python kfserving-request.py http://
<kserve-service>-default.<tenant-name>.svc.cluster.local:80
```

   The output appears similar to the following:

```
200
{u'predictions': [[0.841960549]]}
```

Tutorial 4: Inferencing with Seldon Core

1. Obtain the `inference_seldon.yaml` file from the `tensorflow` directory in the sample zip file.

2. Apply `inference_seldon.yaml` to the tenant namespace:

```
kubectl apply -f inference_seldon.yaml -n <namespace>
```

3. Ensure that the pods are up and running. You can track the status of the serving deployment with the following commands:

```
kubectl get sdep
```

```
kubectl get pods | grep tfserving
```

4. After the pods are up and running, send a request to the model.

   Sample requests are available under `tensorflow/seldon/requests_seldon.py`.

a. In the Jupyter Notebook terminal, install the following Python dependencies:

```
pip install requests lxml --user
```

b. From the Jupyter notebook, launch `seldon-request.py` as follows:

```
python seldon-request.py http://
<seldon-service>.<tenant-name>.svc.cluster.local:8000
```

The output appears similar to the following:

```
200
{u'predictions': [[0.841960549]]}
```

More information

Tutorial: KubeDirector Training and Serving on page 216
Starting with HPE Ezmeral Runtime Enterprise 5.5.0, the KubeDirector training and deployment applications are deprecated, and are unavailable for creation through the HPE Ezmeral ML Ops UI. However, you can still create and use KubeDirector training and deployment applications with YAML files. This tutorial provides instructions on deploying Kubedirector Training and Kubedirector Serving applications with YAML files in HPE Ezmeral ML Ops.

Tutorial: Training with TensorFlow (Financial Series)

Prerequisites:

- An Internet connection is required to download the dependencies needed for this tutorial. This tutorial is not available for Air Gapped environments.
- **If you have not done so already:** Before beginning this tutorial download the , which contains sample files for all of the included Kubeflow tutorials.

The following tutorial is based on the example at https://github.com/mapr/kubeflow-examples/tree/master/financial_time_series.

Step 1: Mount the Volume for Storing a Model

1. Log in to the KubeDirector notebook as an LDAP user.
2. Obtain `pvc-tf-training-fin-series.yaml` from the zip file mentioned above.
3. Upload `pvc-tf-training-fin-series.yaml` to the KubeDirector notebook for the Persistent Volume Claim (PVC).
4. Open the web terminal in the HPE Ezmeral Runtime Enterprise UI, or from the terminal within the KubeDirector notebook.

   NOTE: By default, you cannot execute `kubectl` commands in a newly created KubeDirector notebook. To enable `kubectl` in a notebook, select one of the following methods:

   - **Through the HPE Ezmeral Runtime Enterprise UI:**
     a. In the HPE Ezmeral Runtime Enterprise UI, navigate to the **Tenant** section and initialize a web terminal with the corresponding button.
     b. Start a new Terminal session inside the KubeDirector notebook. Check that the files inside your KubeDirector notebook have the appropriate file permissions that allow you to work with them.

c. Move all files you want to work with to the following path:

```
/bd-fs-mnt/TenantShare
```

d. You can now access the files inside the web terminal with `kubectl`.

- **From inside the KubeDirector notebook:**

  a. To authorize your user inside the KubeDirector notebook, execute the following Jupyter code cell:

  ```
  from ezmllib.kubeconfig.ezkubeconfig import set_kubeconfig
  set_kubeconfig()
  ```

  b. A prompt appears below the code cell you executed. Enter your user password in the prompt.

  c. `kubectl` is now enabled for your KubeDirector notebook. Start a Terminal session in the KubeDirector notebook to work with `kubectl`.

5. Apply the `.yaml` file to create the PVC:

```
kubectl apply -f pvc-tf-training-fin-series.yaml
```

6. Verify that the PVC was created and is in the `bound` state:

```
kubectl get pvc
```

The results should look like this:

```
NAME         STATUS   VOLUME
CAPACITY   ACCESS MODES   STORAGECLASS  AGE
pvctf        Bound    mapr-pv-edeb3067-0332-44cf-88d8-a44be8c39f7c
10Gi        RWX           default     21m
```

Step 2: Exploration Phase

To complete the exploration phase:

1. Log in to the KubeDirector notebook.

2. Perform the following:

   a. Upload `FinancialTimeSerieswithFinanceData.ipynb`.

b. If your environment is behind a proxy, open the uploaded notebook and perform the following workaround:

Add a cell above the first step, and insert the following specifying your proxies:

```
%env https_proxy=YOUR_PROXY
%env http_proxy=YOUR_PROXY
%env no_proxy=YOUR_PROXY

%env HTTPS_PROXY=YOUR_PROXY
%env HTTP_PROXY=YOUR_PROXY
%env NO_PROXY=YOUR_PROXY
```

c. Walk throug the notebook step by step to better understand the problem and suggested solutions.

Step 3: Training Phase

To complete the training phase:

1. Upload and apply `financial-series-tfjob.yaml`.
2. Verify the TensorFlow job is created successfully:

```
kubectl get tfjobs
NAME          STATE       AGE
trainingjob   Created   2m47s
```

3. Verify that pods are created, running, and then completed:

```
kubectl get pods | grep trainingjob
trainingjob-ps-0            0/1     Completed   0     5m39s
trainingjob-worker-0        0/1     Completed   0     5m39s
```

4. Check the logs to walk through the training process description:

```
kubectl logs trainingjob-ps-0
```

The output should appear as follows:

```
…
INFO:tensorflow:SavedModel written to: b'model/1/saved_model.pb'
INFO:tensorflow:SavedModel written to: b'model/1/saved_model.pb'
INFO:root:copy files to /data/model/1
5000 0.5607639
10000 0.5755208
15000 0.5946181
20000 0.6145833
25000 0.6302083
30000 0.6449653
Precision =  0.9142857142857143
Recall =  0.2222222222222222
F1 Score =  0.35754189944134074
Accuracy =  0.6006944444444444
```

Step 4: Clean Up the Namespace

To clean up the namespace:

1. Delete both the pods and the job with the following command:

```
kubectl delete tfjob trainingjob
```

2. Delete the PVC:

```
kubectl delete -f pvc-tf-training-fin-series.yaml
```

Tutorial: Serving a TensorFlow Model with KServing (Financial Series)

Prerequisites:

- An Internet connection is required to download the dependencies needed for this tutorial. This tutorial is not available for Air Gapped environments.
- **If you have not done so already:** Before beginning this tutorial, download the , which contains sample files for all of the included Kubeflow tutorials.

Serving TensorFlow Model with KServing

1. In a KubeDirector Jupyter notebook, create or upload the serving `.yaml` file `financial-series-serving.yaml`.
2. Open the web terminal in the HPE Ezmeral Runtime Enterprise UI, or from the terminal within the KubeDirector notebook.

   NOTE: By default, you cannot execute `kubectl` commands in a newly created KubeDirector notebook. To enable `kubectl` in a notebook, select one of the following methods:

   - **Through the HPE Ezmeral Runtime Enterprise UI:**
     a. In the HPE Ezmeral Runtime Enterprise UI, navigate to the **Tenant** section and initialize a web terminal with the corresponding button.
     b. Start a new Terminal session inside the KubeDirector notebook. Check that the files inside your KubeDirector notebook have the appropriate file permissions that allow you to work with them.
     c. Move all files you want to work with to the following path:

     ```
     /bd-fs-mnt/TenantShare
     ```

     d. You can now access the files inside the web terminal with `kubectl`.
   - **From inside the KubeDirector notebook:**
     a. To authorize your user inside the KubeDirector notebook, execute the following Jupyter code cell:

     ```
     from ezmllib.kubeconfig.ezkubeconfig import set_kubeconfig
     set_kubeconfig()
     ```

     b. A prompt appears below the code cell you executed. Enter your user password in the prompt.

c. `kubectl` is now enabled for your KubeDirector notebook. Start a Terminal session in the KubeDirector notebook to work with `kubectl`.

3. In the terminal, apply the file:

```
$ kubectl apply -f financial-series-serving.yaml
```

4. Check that inference service, revision, and relative pod are created:

```
$ kubectl get pods | grep finance-sample
NAME
READY   STATUS    RESTARTS   AGE
finance-sample-predictor-default-d45t4-deployment-784457c4wjtl2
3/3     Running   0          63s
```

```
$ kubectl get ksvc
NAME
URL
LATESTCREATED             LATESTREADY
READY   REASON
finance-sample-predictor-default   http://
finance-sample-predictor-default.nkili.example.com
finance-sample-predictor-default-9lrnm
finance-sample-predictor-default-9lrnm   True
```

```
$ kubectl get revision
NAME                                      CONFIG
NAME                        K8S SERVICE NAME
GENERATION   READY   REASON
finance-sample-predictor-default-9lrnm
finance-sample-predictor-default
finance-sample-predictor-default-9lrnm   1       True
```

```
$ kubectl get inferenceservices
NAME
URL                                                                  READY
DEFAULT TRAFFIC   CANARY TRAFFIC   AGE
finance-sample   http://finance-sample.<profile-name>.example.com/v1/
models/finance-sample   True    100                     4h52m
```

5. Check that virtual service is created:

```
$ kubectl get virtualservices | grep finance-sample
finance-sample                          [kubeflow-gateway.kubeflow
knative-serving/cluster-local-gateway]
[finance-sample.nkili.example.com
finance-sample.nkili.svc.cluster.local]
                      24m
finance-sample-predictor-default        [knative-serving/
cluster-local-gateway kubeflow/kubeflow-gateway]
[finance-sample-predictor-default.nkili
finance-sample-predictor-default.nkili.example.com
finance-sample-predictor-default.nkili.svc
finance-sample-predictor-default.nkili.svc.cluster.local]   24m
finance-sample-predictor-default-mesh
[mesh]
[finance-sample-predictor-default.nkili
finance-sample-predictor-default.nkili.svc
finance-sample-predictor-default.nkili.svc.cluster.local]
                                      24m
```

6. Create or upload the file `kserving-request.py`.

7. Install the following Python dependencies:

```
$ pip install requests lxml --user
```

8. Launch `kserving-request.py` with the following options:

```
$ python kserving-request.py http://
finance-sample-predictor-default.$NAMESPACE.svc.cluster.local:80
```

For example:

```
$ python kserving-request.py http://
finance-sample-predictor-default.test1.svc.cluster.local:80
```

The output should appear similar to the following:

```
200
{'predictions': [{'model-version': '1', 'prediction': 0}]}
```

Cleaning the Namespace After Running the Sample

NOTE: Make sure `finance-sample` is the only used name for resources in your cluster.

1. Delete pods:

```
kubectl delete pod -n $(kubectl get pods -n <USED_NAMESPACE> | grep
'finance-sample-predictor' | awk '{ print $1, $2 }' < echo)
```

The output should appear similar to the following:

```
kubectl delete pod -n $(kubectl get pods -n kfdf-tenant | grep
'finance-sample-predictor' | awk '{ print $1, $2 }' < echo)

pod "finance-sample-predictor-default-00001-deployment-7bbfbd88shd88"
deleted
```

2. Delete `revision`:

```
kubectl delete revision -n $(kubectl get revision -A | grep
'finance-sample-predictor' | awk '{ print $1, $2 }')
```

The output should appear as follows:

```
kubectl delete revision -n $(kubectl get revision -A | grep
'finance-sample-predictor' | awk '{ print $1, $2 }')

revision.serving.knative.dev "finance-sample-predictor-default-00001"
deleted
```

3. Delete `kservice`:

```
kubectl delete ksvc -n $(kubectl get ksvc -A | grep 'finance-sample' |
awk '{ print $1, $2 }')
```

The output should appear as follows:

```
kubectl delete ksvc -n $(kubectl get ksvc -A | grep 'finance-sample' |
awk '{ print $1, $2 }')
service.serving.knative.dev "finance-sample-predictor-default" deleted
```

4. Delete `inferenceservice`:

```
kubectl delete inferenceservice -n $(kubectl get inferenceservice -A |
grep 'finance-sample' | awk '{ print $1, $2 }')
```

The output should appear as follows:

```
kubectl delete inferenceservice -n $(kubectl get inferenceservice -A |
grep 'finance-sample' | awk '{ print $1, $2 }')
inferenceservice.serving.kserve.io "finance-sample" deleted
```

Tutorial: Training a PyTorch Model (Pytorch MNIST)

If you have not done so already: Before beginning this tutorial, download the , which contains sample files for all of the included Kubeflow tutorials.

To complete this tutorial:

1. Log in to the KubeDirector notebook as an LDAP user.

2. Create or upload the `.yaml` file for the PyTorch job: `pytorch-mnist-ddp-cpu.yaml`.

3. Open the web terminal in the HPE Ezmeral Runtime Enterprise UI, or from the terminal within the KubeDirector notebook.

   NOTE: By default, you cannot execute `kubectl` commands in a newly created KubeDirector notebook. To enable `kubectl` in a notebook, select one of the following methods:

   - **Through the HPE Ezmeral Runtime Enterprise UI:**

     a. In the HPE Ezmeral Runtime Enterprise UI, navigate to the **Tenant** section and initialize a web terminal with the corresponding button.

     b. Start a new Terminal session inside the KubeDirector notebook. Check that the files inside your KubeDirector notebook have the appropriate file permissions that allow you to work with them.

     c. Move all files you want to work with to the following path:

     ```
     /bd-fs-mnt/TenantShare
     ```

     d. You can now access the files inside the web terminal with `kubectl`.

   - **From inside the KubeDirector notebook:**

     a. To authorize your user inside the KubeDirector notebook, execute the following Jupyter code cell:

     ```
     from ezmllib.kubeconfig.ezkubeconfig import set_kubeconfig
     set_kubeconfig()
     ```

     b. A prompt appears below the code cell you executed. Enter your user password in the prompt.

     c. `kubectl` is now enabled for your KubeDirector notebook. Start a Terminal session in the KubeDirector notebook to work with `kubectl`.

4. Create the PyTorch job:

   ```
   kubectl apply -f pytorch-mnist-ddp-cpu.yaml
   ```

   IMPORTANT: To complete this tutorial in an Air Gapped environment, you must perform the following:

   a. Push the `bluedata/pytorch:mnist-ddp-cpu` image to your Air Gap registry.

   b. Add the prefix of your Air Gap registry before the image name within the `.yaml` file. For example:

   ```
   <air-gap-registry>/bluedata/pytorch:mnist-ddp-cpu
   ```

5. Verify the PyTorch job is created:

```
$ kubectl get pytorchjobs
NAME                    STATE       AGE
pytorch-mnist-ddp-cpu   Created     3s
```

6. Verify the relative pods are created:

```
$ kubectl get pods -l job-name=pytorch-mnist-ddp-cpu
NAME                              READY   STATUS              RESTARTS
AGE
pytorch-mnist-ddp-cpu-master-0   0/1     ContainerCreating   0
6s
pytorch-mnist-ddp-cpu-worker-0   0/1     Init:0/1            0
6s
pytorch-mnist-ddp-cpu-worker-1   0/1     Init:0/1            0
6s
pytorch-mnist-ddp-cpu-worker-2   0/1     Init:0/1            0
5s
```

7. Verify the status for the PyTorch job pods. Wait until all pods have status `Completed`:

```
kubectl get pods -l job-name=pytorch-mnist-ddp-cpu
```

8. Insepct the logs to observe PyTorch training progress:

```
PODNAME=$(kubectl get pods -l
job-name=pytorch-mnist-ddp-cpu,replica-type=master,replica-index=0 -o
name) \
kubectl logs -f ${PODNAME};
```

You can also check the status of the PyTorch job with the `describe` command:

```
kubectl describe pytorchjob pytorch-mnist-ddp-cpu
...
//message: PyTorchJob pytorch-mnist-ddp-cpu is successfully completed.
...
```

9. Clean up after the job run:

```
kubectl delete -f pytorch-mnist-ddp-cpu.yaml
...
//message:
persistentvolumeclaim "pvcpy" deleted
pytorchjob.kubeflow.org "pytorch-mnist-ddp-cpu" deleted
...
```

Tutorial: Katib Hyperparameter Tuning

Example 1: TensorFlow

To complete this tutorial:

1. If you have not done so already, download the , which contains sample files for all of the included Kubeflow tutorials.

2. Deploy the example file:

```
kubectl apply -f tensorflow-example.yaml
```

3. Open the Kubeflow UI and nagivate to **Home** > **View Katib experiments**.
4. Click the experiment name, and then observe the running trials.
5. Check the experiment status:

```
kubectl get experiment
```

6. Check the experiment trials:

```
kubectl get trial
```

Example 2: Random Algorithm

This example may take some time to finish, depending on the resources allocated.

The following hyperparameters can be tuned:

- `--lr` - learning rate
- `--num-layers` - Number of layers in the neural networks
- `--optimizer`

To launch an experiment using the random algorithm example:

1. If you have not done so already, download the file, which contains sample files for all of the included Kubeflow tutorials.
2. Deploy the example file:

```
kubectl apply -f random-example.yaml
```

This example embeds the hyperparameters as arguments. You can embed hyperparameters in another way (e.g. by using environment variables) by using the template defined in the `TrialTemplate.GoTemplate.RawTemplate` section of the `yaml` file. The template uses the Go template format (link opens an external website in a new browser tab/window).

This example randomly generates the following hyperparameters:

- `--lr` - Learning rate (type: `double`).
- `--num-layers` - Number of layers in the neural network (type: `integer`).
- `--optimizer` - Optimizer (type: `categorical`).

Check the experiment status:

```
kubectl describe experiment random-example
```

Example 3: PyTorch

This example may take some time to finish, depending on the resources allocated.

1. If you have not done so already, download the file, which contains sample files for all of the included Kubeflow tutorials
2. Deploy the example file:

```
kubectl apply -f pytorch-example.yaml
```

3. Open the Kubeflow UI and navigate to **Home** > **View Katib experiments**.
4. Click the experiment name, and then observe the trials running.
5. Check the experiment status:

```
kubectl get experiment
```

6. Use the following command to check trials of the experiment:

```
kubectl get trial
```

Clean Up

Delete the examples with the following commands:

- Random algorithm example:

```
kubectl delete -f random-example.yaml
```

- Tensorflow example:

```
kubectl delete -f tensorflow-example.yaml
```

- PyTorch example:

```
kubectl delete -f pytorchjob-example.yaml
```

Sample Katib Commands

To check experiment results via the `kubectl` CLI.

- List experiments:

```
kubectl get experiment
```

```
NAME                STATUS      AGE
random-experiment   Succeeded   25m
```

- Check experiment result

```
kubectl get experiment random-example -o yaml
```

- List trials

```
kubectl get trials

NAME                          STATUS      AGE
random-experiment-24lgqghm    Succeeded   26m
```

- Check trial detail

```
kubectl get trials random-experiment-24lgqghm -o yaml
```

To check the status using the interface:

1. Go to the Kubeflow home page.
2. Click the **View Katib experiments** button.
3. Click the name of the experiment.
4. Observe the built experiment graph after all the trials have **Succeeded**.

Tutorial: ML Metadata

If you have not done so already: Before beginning this tutorial, download the , which contains sample files for all of the included Kubeflow tutorials.

To complete this tutorial:

1. Connect to the JupyterLab notebook server and upload the `demo-ml.ipynb` notebook.
2. Run the notebook step by step and obersve the result on the **Artifacts** page in the Kubeflow UI (workspace: `unknown`, name: `MNIST-v1`).

For more information about the ML Metadata component, see: https://github.com/google/ml-metadata/blob/master/g3doc/get_started.md.

Tutorial: Sample Pipeline in the Pipelines Interface

Look at this example from the Kubeflow documentation (link opens an external website in a new browser tab/window).

Running a Basic Pipeline

1. Open the Kubeflow dashboard (see Accessing the Kubeflow Dashboard), then access the **Pipelines** page.
2. Click the sample name **[Tutorial] DSL - Control Structures**.
3. Click **Create experiment**, then follow the on-screen prompts.
4. Create a run by clicking the **Start** button.
5. Select the name of the run on the **Experiments** dashboard.
6. Explore the graph and other aspects of your run by selecting the graph components and other interface elements.

Running a Pipeline in Jupyter Notebook

1. Create or open the KubeDirector notebook.

2. Upload the `lightweight_component.ipynb` notebook to the Jupyter Notebook.
3. Open the uploaded file and execute each cell in the Notebook until it is finished.
4. In the last step, follow the links to open the experiment and run it in the Pipelines interface.

Cleaning the Namespace After Running the Sample

After completing the tutorials, you can perform cleanup steps to remove the pods from your namespace. These cleanup steps are applicable to both the Running a Basic Pipeline on page 234 tutorial and the Running a Pipeline in Jupyter Notebook on page 234 tutorial.

The following is an example of Kubernetes pods in the `tenant` namespace after successful completion of the Running a Basic Pipeline on page 234 tutorial:

```
```
NAME   READY
STATUS   RESTARTS   AGE
conditional-execution-pipeline-with-exit-handler-tb6p4-1576391149   0/2
Completed   0   15m
conditional-execution-pipeline-with-exit-handler-tb6p4-1865731256   0/2
Error   0   15m
conditional-execution-pipeline-with-exit-handler-tb6p4-2761648130   0/2
Completed   0   14m
conditional-execution-pipeline-with-exit-handler-tb6p4-2796593516   0/2
Completed   0   15m
conditional-execution-pipeline-with-exit-handler-tb6p4-39186315   0/2
Completed   0   16m
```
```

NOTE: The `Error` state for one of the pipeline's pods is expected behavior, which occurs after looking through the pipeline's graph and components.

To remove pods from the namespace:

1. Start a new terminal session in the KubeDirector notebook.
2. Open the web terminal in the HPE Ezmeral Runtime Enterprise UI, or from the terminal within the KubeDirector notebook.

   NOTE: By default, you cannot execute `kubectl` commands in a newly created KubeDirector notebook. To enable `kubectl` in a notebook, select one of the following methods:

   - **Through the HPE Ezmeral Runtime Enterprise UI:**
     a. In the HPE Ezmeral Runtime Enterprise UI, navigate to the **Tenant** section and initialize a web terminal with the corresponding button.
     b. Start a new Terminal session inside the KubeDirector notebook. Check that the files inside your KubeDirector notebook have the appropriate file permissions that allow you to work with them.
     c. Move all files you want to work with to the following path:

        ```
        /bd-fs-mnt/TenantShare
        ```

     d. You can now access the files inside the web terminal with `kubectl`.
   - **From inside the KubeDirector notebook:**

a. To authorize your user inside the KubeDirector notebook, execute the following Jupyter code cell:

```
from ezmllib.kubeconfig.ezkubeconfig import set_kubeconfig
set_kubeconfig()
```

b. A prompt appears below the code cell you executed. Enter your user password in the prompt.

c. `kubectl` is now enabled for your KubeDirector notebook. Start a Terminal session in the KubeDirector notebook to work with `kubectl`.

3. Run the following `kubectl` commands to remove the pipeline's resources:

   - ```
     kubectl get wf
     ```

   - ```
     kubectl delete wf <wf_name>
     ```

     For example:

     ```
     kubectl delete wf
     conditional-execution-pipeline-with-exit-handler-k5v2w
     ```

4. The result of the pipeline run remains on the **Runs** page of the **Kubeflow dashboard**. If you want to remove the results from the list on the **Runs** page, you can archive the results instead.

Tutorial: Kale Extension in Kubedirector Notebook

The Kale extension for HPE Ezmeral Runtime Enterprise enables the automation of Jupyter Notebook deployments to Kubeflow Pipelines.

Kale enables you to deploy local or cloud-based Jupyter Notebooks to Kubeflow Pipelines, automatically converting the Notebook to a valid Kubeflow Pipelines deployment. Kale also resolves data dependencies and manages the lifecycle of the pipeline.

Kale Tutorial

If you have not done so already: Before beginning this tutorial, download the , which contains sample files for all the included Kubeflow tutorials.

To complete this tutorial:

1. Log in to the Kubedirector notebook.

2. Before enabling the Kale extension, execute the steps from `examples/kubeflow/kale/README.ipynb` in the Kubedirector notebook. Proceed as follows:

a. Run the following cells:

```
from ezmllib.kubeconfig.ezkubeconfig import set_kubeconfig
set_kubeconfig()
```

```
from ezmllib.kubeflow.ezkfp import KfSession
K = KfSession()
client=K.kf_client()
```

```
# List the current kubeflow pipelines
print(client.list_pipelines())
```

b. If `print(client.list_pipelines())` returns a nonempty array of pipelines, you have successfully created your KFP client.

3. Upload `candies_sharing.ipynb` using the **Upload Files** button in the Kubedirector notebook.

4. Enable the Kale extension, as follows:

   a. Click the **Kale** button in the navigation pane. The **Kale Deployment Panel** opens.

   b. Select **Enable**.

   For more information on the **Kale Deployment Panel**, see The Kale Deployment Panel on page 238.

5. Open the `candies_sharing.ipynb` file.

6. Run all cells in the notebook using **Run** > **Run All Cells**.

7. At the bottom of the **Kale Deployment Panel**, click **Compile and Run**.

8. On the Kubeflow **Dashboard**, open the **Runs** page. Check the status of the pipeline run launched from the Kubedirector notebook.

9. Perform cleanup steps to remove the pod from your namespace, as described in Cleaning the Namespace After Running the Sample on page 237.

Cleaning the Namespace After Running the Sample

After completing the Kale Tutorial on page 236, you can perform cleanup steps to remove the pod from your namespace.

The following is an example of Kubernetes pods in the `tenant` namespace after successful completion of the tutorial:

```
```
NAME   READY   STATUS   RESTARTS
AGE
candies-sharing-bq9mb-1430592451   0/2   Completed   0
17h
candies-sharing-bq9mb-196233272   0/2   Completed   0
17h
candies-sharing-bq9mb-213010891   0/2   Completed   0
17h
candies-sharing-bq9mb-229788510   0/2   Completed   0
17h
candies-sharing-bq9mb-833381010   0/1   Completed   0
```
```

```
17h
```

To remove pods from the namespace:

1. Start a new terminal session in the KubeDirector notebook.
2. Run the following `kubectl` commands to remove the pipeline's resources:
   - ```
     kubectl get wf
     ```
     ```
     NAME                          STATUS      AGE
     candies-sharing-zshg5         Succeeded   3m44s
     ```
   - ```
     kubectl delete wf <wf_name>
     ```

     For example:

     ```
     kubectl delete wf candies-sharing-zshg5
     ```
3. The result of the pipeline run remains on the **Runs** page of the **Kubeflow dashboard**. If you want to remove the results from the list on the **Runs** page, you can archive the results instead.

The Kale Deployment Panel

The **Kale Deployment Panel** contains the following sections:

- **Pipeline Metadata**: Define the name of the experiment and pipeline, and provide a description.
- **Run**: Enable the Katib feature for this pipeline. Set up appropriate hyperparameters and other variables by clicking the **Enable Katib Job** button.
- Click **Advanced Settings**. In **Advanced Settings**, you can set a container image. This container image is used for all steps of the current pipeline.

NOTE: The Kale extension does not support the Rok snapshot feature at this time. Instead, you can manually create a volume or use an existing volume for this pipeline.

Editing a Cell

To edit a cell:

1. Click the **pencil** icon. Cell information for Kale opens.
2. Edit the information about this cell.
3. Click the **x** button. Cell information closes.

Tutorial: TensorBoard

Prerequisites:

- KubeDirector notebook with ML Ops toolkits and Persistent Volume attached
- An Internet connection is required to download the MNIST dataset for this tutorial. This tutorial is not available for Air Gapped environments.

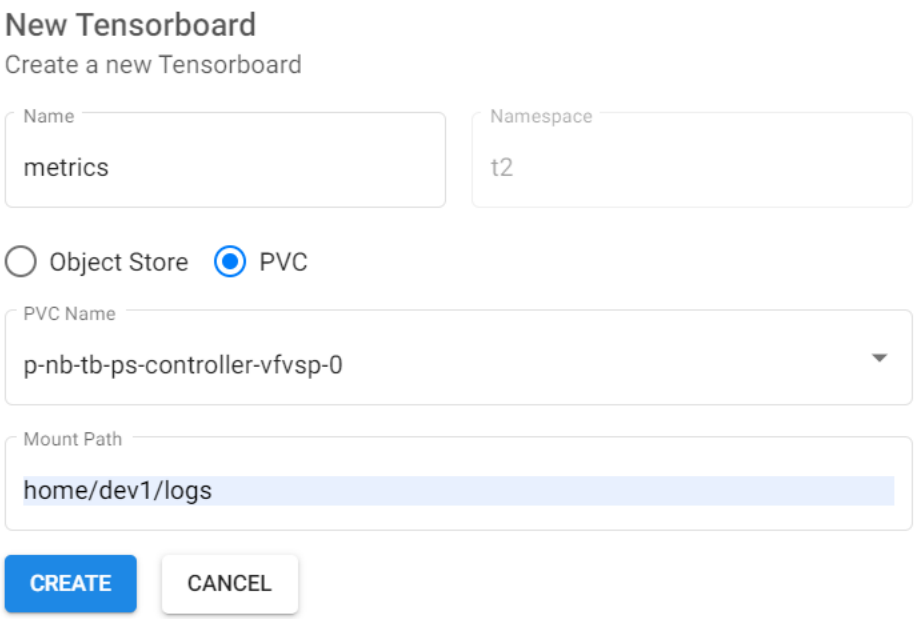
- **If you have not done so already:** Before beginning this tutorial, download the , which contains sample files for all of the included Kubeflow tutorials.

IMPORTANT: If you experience freezing while downloading the MNIST dataset, restart the notebook kernel.

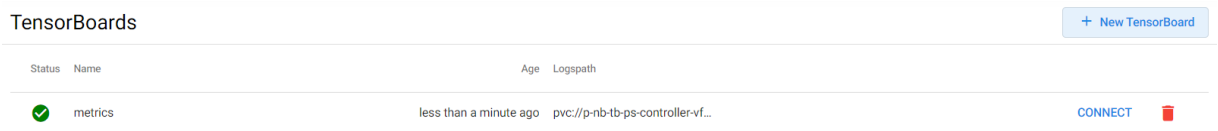
To complete the tutorial:

1. Log in to the KubeDirector notebook as an LDAP user.
2. Upload `tensorboard.ipynb` file using the respective button in the Jupyter notebook.
3. Open `tensorboard.ipynb` and run it using **Run** > **Run All Cells** in the menu panel.
4. Open the Kubeflow Dashboard, log in with the LDAP user credentials you used in step 1, and open the **Tensorboards** page.

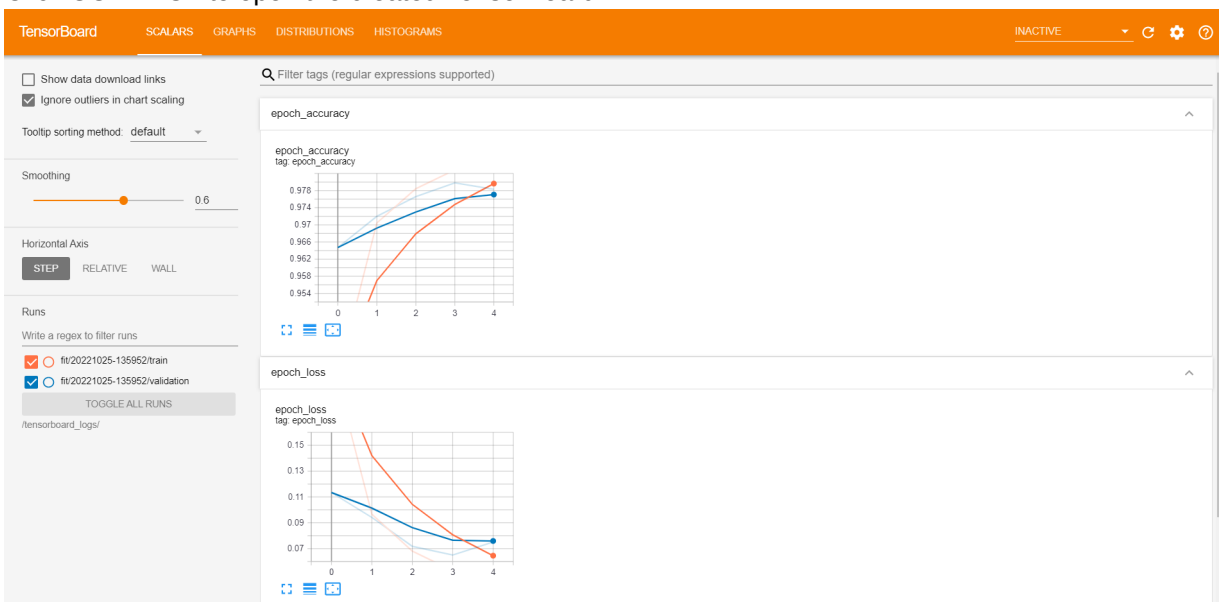
   Ensure that the tenant namespace is selected in the **Namespace** drop-down list.
5. Create a new TensorBoard:
   a. Select **+ NEW TENSORBOARD**. The **New Tensorboard** dialog box opens.



   b. In the **Name** field, enter a name for the TensorBoard. For example, *metrics*.
   c. Select the **PVC** check box.
   d. In the **PVC Name** drop-down list, select the KubeDirector notebook's persistent volume.
   e. In the **Mount Path** field, enter the path `home/<user>/logs`. For example, if the notebook user is `dev1`, the user mount path is `home/dev1/logs`.
   f. Select **CREATE**.
6. Wait until the TensorBoard becomes available.



7. Click **CONNECT** to open the created TensorBoard.



8. **(Optional)** You can delete the created TensorBoard with the **Delete** button from the **Tensorboards** page of the **Kubeflow Dashboard**.

Tutorial: Argo Workflows

If you have not done so already: Before beginning this tutorial, download the , which contains sample files for all of the included Kubeflow tutorials.

This article provides the following two examples:

- Simple Workflow
- Parallel Execution Workflow

Simple Workflow

To complete the simple workflow:

1. Upload `argo-hello-world.yaml` or `argo-hello-world_limit.yaml`:
   - If you are running in a profile namespace:

     ```
     kubectl apply -f argo-hello-world.yaml -n <namespace>
     ```

   - If you are running in a tenant namespace:

     ```
     kubectl apply -f argo-hello-world_limit.yaml -n <tenant_namespace>
     ```

2. Check if the workflow is created:

   ```
   kubectl get wf -n <namespace>
   ```

3. Verify the relative pod `hello-world` is created and running.

4. To remove the workflow, enter the following command:

```
kubectl delete wf hello-world -n <namespace>
```

Parallel Execution Workflow

To complete the parallel execution workflow:

1. Upload `argo-parallel-nested.yaml` or `argo-parallel-nested_limit.yaml` and apply the nested Argo workflow:
   - If you are using the user namespace:

     ```
     kubectl apply -f argo-parallel-nested.yaml -n <namespace>
     ```

   - If you are using the tenant namespace:

     ```
     kubectl apply -f argo-parallel-nested_limit.yaml -n <tenant_namespace>
     ```

2. Check that the pods are created per the template in the `.yaml` file.

3. The workflow is successfully completed:

```
parallelism-nested-dag-122631356
0/2     Completed          0          4m36s
parallelism-nested-dag-139408975
0/2     Completed          0          4m36s
parallelism-nested-dag-172964213
0/2     Completed          0          4m52s
parallelism-nested-dag-1858975680
0/2     Completed          0          4m19s
parallelism-nested-dag-1892530918
0/2     Completed          0          4m3s
parallelism-nested-dag-1909308537
0/2     Completed          0          4m3s
parallelism-nested-dag-2922067776
0/2     Completed          0          2m56s
parallelism-nested-dag-2938845395
0/2     Completed          0          2m56s
parallelism-nested-dag-2972400633
0/2     Completed          0          3m12s
parallelism-nested-dag-3470313907
0/2     Completed          0          4m19s
parallelism-nested-dag-3487091526
0/2     Completed          0          4m4s
parallelism-nested-dag-3503869145
0/2     Completed          0          4m4s
parallelism-nested-dag-458928436
0/2     Completed          0          3m30s
parallelism-nested-dag-475706055
0/2     Completed          0          3m30s
parallelism-nested-dag-492483674
0/2     Completed          0          3m46s
```

4. Remove the workflow with the following command:

```
kubectl delete wf parallelism-nested-dag -n <namespace>
```

Tutorial: GitHub Issue Summarization

Prerequisites:

- An Internet connection is required to download the dependencies needed for this tutorial. This tutorial is not available for Air Gapped environments.
- Before starting this tutorial, ensure that you have a Jupyter Notebook with ML toolkits enabled. The KubeDirector notebook should have PVC, as the trained model is stored in this notebook, and Seldon core reads this model from the notebook's PVC.

To complete this tutorial:

1. Log into the KubeDirector notebook as an LDAP user.
2. In the KubeDirector notebook, open the folder `examples/kubeflow/text-processing`. The working directory contains all the necessary files to work with the example.
3. Open the Training.ipynb notebook file and run all cells: `Run -> Run All Cells`.
4. Get the full path of the current home directory in the notebook. Edit the `seldon-issue-sum-deployment.yaml` file and replace `<home_dir>` with the full path of the current home directory.
5. Open the web terminal in the HPE Ezmeral Runtime Enterprise UI, or from the terminal within the KubeDirector notebook.

   NOTE: By default, you cannot execute `kubectl` commands in a newly created KubeDirector notebook. To enable `kubectl` in a notebook, select one of the following methods:

   - **Through the HPE Ezmeral Runtime Enterprise UI:**
     a. In the HPE Ezmeral Runtime Enterprise UI, navigate to the **Tenant** section and initialize a web terminal with the corresponding button.
     b. Start a new Terminal session inside the KubeDirector notebook. Check that the files inside your KubeDirector notebook have the appropriate file permissions that allow you to work with them.
     c. Move all files you want to work with to the following path:

     ```
     /bd-fs-mnt/TenantShare
     ```

     d. You can now access the files inside the web terminal with `kubectl`.
   - **From inside the KubeDirector notebook:**
     a. To authorize your user inside the KubeDirector notebook, execute the following Jupyter code cell:

     ```
     from ezmllib.kubeconfig.ezkubeconfig import set_kubeconfig
     set_kubeconfig()
     ```

     b. A prompt appears below the code cell you executed. Enter your user password in the prompt.

c. `kubectl` is now enabled for your KubeDirector notebook. Start a Terminal session in the KubeDirector notebook to work with `kubectl`.

6. Apply `seldon-issue-sum-deployment.yaml` with the following command:

```
kubectl apply -f seldon-issue-sum-deployment.yaml
```

7. Execute the following command to make a prediction:

```
python seldon-request.py http://
issue-summarization-example.<tenant-name>.svc.cluster.local:8000
```

Spark on Kubernetes

The topics in this section provide information about Apache Spark on Kubernetes in HPE Ezmeral Runtime Enterprise. (Not available with HPE Ezmeral Runtime Enterprise Essentials.)

Spark Overview

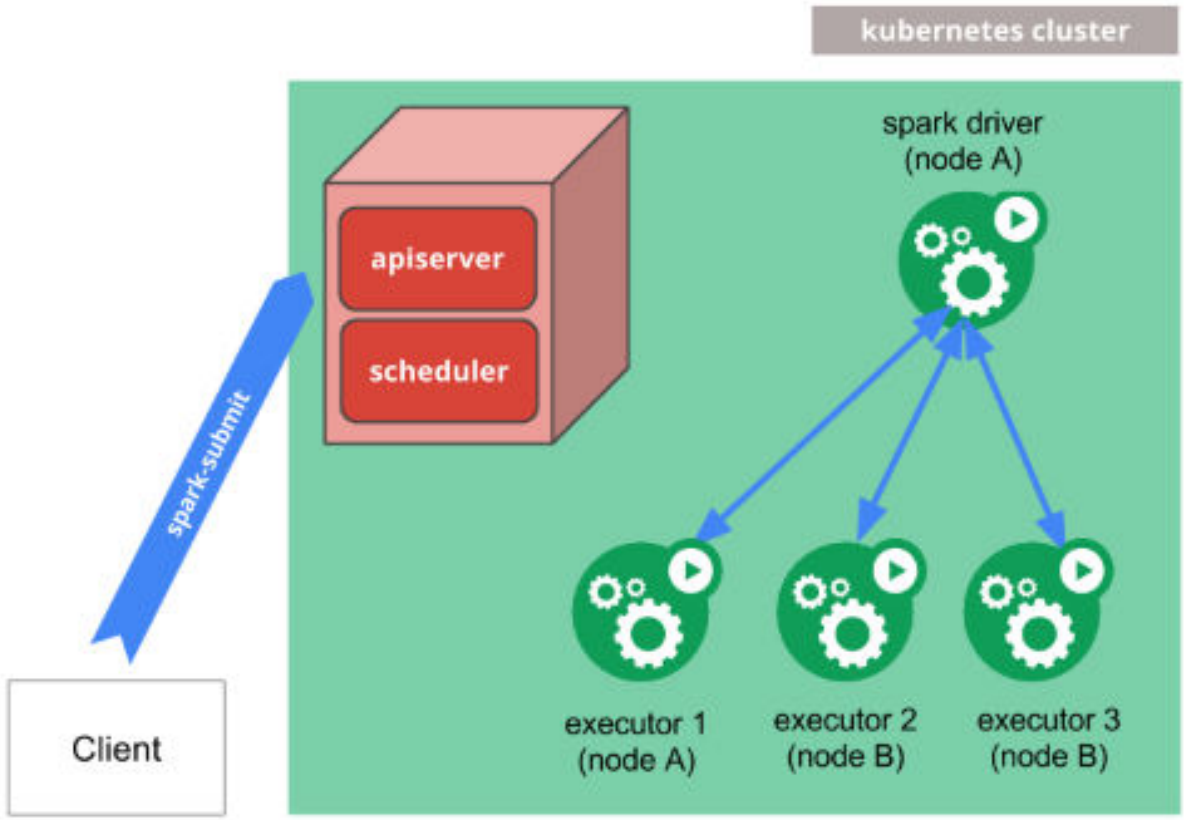
This topic provides a brief overview of Apache Spark on HPE Ezmeral Runtime Enterprise.

Spark is a unified analytics engine with high data processing speed that offers the high-level API in Java, Scala, Python, and R. Spark provides the in-memory computing and optimized query execution for the fast data processing.

You can run the Spark on Kubernetes managed clusters on HPE Ezmeral Runtime Enterprise. For more information about running Spark on Kubernetes, see Apache Spark on Kubernetes.

NOTE: Starting from HPE Ezmeral Runtime Enterprise 5.3, Spark Standalone is no longer supported.

When you submit a Spark application using the spark-submit to a Kubernetes cluster, you start a Spark driver within a Kubernetes pod. This driver creates the Spark executor pods within the Kubernetes cluster to execute the tasks.



Apache Spark on HPE Ezmeral Runtime Enterprise

- HPE Ezmeral Runtime Enterprise provides enterprise ready unified Spark which supports Apache Livy based RESTful interface.
- Spark 3.x.x supports RAPIDS Accelerator by Nividia to accelerate the processing for Spark by using the GPUs. See Nvidia Spark-RAPIDS Accelerator for Spark on page 254.
- Spark 3.x.x provides ACID transactions for Spark applications with Delta Lake. See Delta Lake with Apache Spark on page 296.
- Spark supports the following:
  - Global Hive Metastore: Starting from HPE Ezmeral Runtime Enterprise 5.3, you can access the Hive Metastore configured inside one Kubernetes cluster from Spark applications that is configured in another Kubernetes cluster. See Hive Metastore on page 309.
  - Spark History Server. See Spark History Server on page 297.
  - Spark Thrift Server. See Spark Thrift Server on page 305.
- You can run a Spark job on Kubernetes clusters in the HPE Ezmeral Runtime Enterprise in the following ways:
  - Using Spark Operator. See Spark Operator on page 264.
  - Using Livy to make REST calls. See Submitting Spark Application Using Livy on page 282.
  - Using spark scripts from spark-client pods. See Submitting Spark Applications Using spark-submit on page 295.

- Using Airflow to schedule Spark jobs. See Using Airflow to Schedule Spark Applications on page 314.

- You can run Spark jobs in the Data Fabric tenants or non Data Fabric tenants:

Data Fabric Tenants	Tenants created on HPE Ezmeral Data Fabric on Kubernetes on the HPE Ezmeral Runtime Enterprise or on HPE Ezmeral Data Fabric on Bare Metal outside of the HPE Ezmeral Runtime Enterprise. See HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579.
Non Data Fabric Tenants	Tenants created on an external storage that is not the HPE Ezmeral Data Fabric.

- To learn about new enhancements and changes for Spark on HPE Ezmeral Runtime Enterprise, see What's New in Version 5.6.x on page 99.

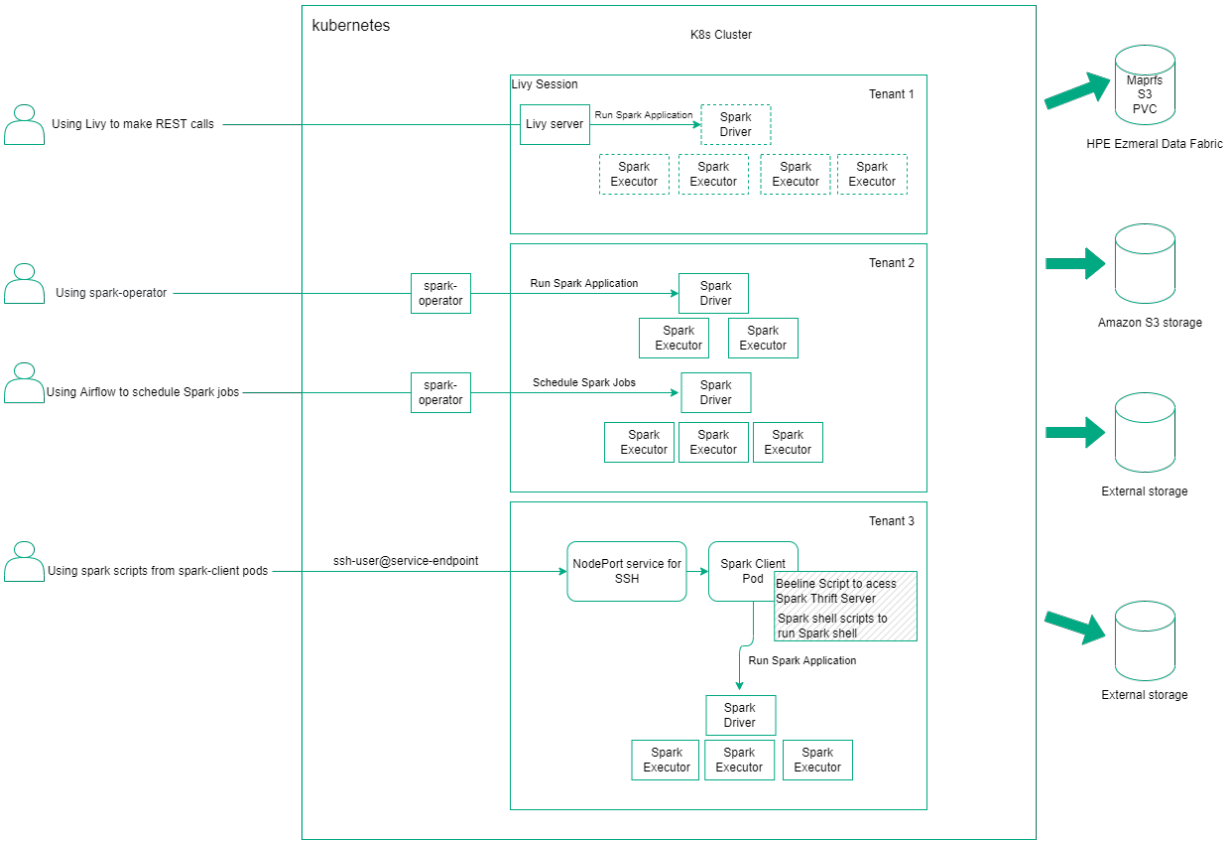


Figure 1: Overview of Running Spark Applications on HPE Ezmeral Runtime Enterprise

Spark Version Comparison Matrix

This matrix shows the different versions of Spark supported on HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise 5.6.x

HPE Ezmeral Runtime Enterprise supports open-source Spark images, and HPE distribution for Apache Spark 2.4.7 and Apache Spark 3.3.1. For details, see Spark Operator on page 264.

The following table compares the two different versions of Spark on HPE Ezmeral Runtime Enterprise.

Table

Capabilities	Spark 2.4.7	Spark 3.3.1
Enterprise readiness	Yes	Yes (Default and Recommended)
Installation	Using KubeDirector applications in HPE Ezmeral Runtime Enterprise GUI or manual deployment using Helm charts.	Using KubeDirector applications in HPE Ezmeral Runtime Enterprise GUI or manual deployment using Helm charts.
Delta Lake Support	No	Yes
Data Fabric (Filesystem, Database, Streams)	Yes	Yes
DTAP support for HDFS	Yes	Yes
Spark History Server	Yes	Yes
REST interface (Livy service)	Livy	Livy
Spark Thrift Server	Yes	Yes
Spark Client Pods	Yes	Yes
Enterprise Security	Yes (AD/LDAP, Data Fabric SASL, Kubernetes RBAC)	Yes (AD/LDAP, Data Fabric SASL, Kubernetes RBAC)
GPU (Nvidia RAPIDS)	No	Yes (RAPIDS 22.10.0)

Interoperability Matrix for Spark

This section provides information about support and interoperability for Spark and its components with HPE Ezmeral Runtime Enterprise.

The following table lists the versions of Spark and its components by HPE Ezmeral Runtime Enterprise release.

HPE Ezmeral Runtime EnterpriseVersions	Spark Operator Versions	Spark Applications Versions*	Livy Versions	Spark History Server Versions	Spark Thrift Server Versions	Hive Metastore Versions
5.6.x and higher	1.3.8.0-hpe	3.3.1	0.7.0	3.3.1	3.3.1	3.1.3
		2.4.7	0.7.0-2.4.7			
5.5.1	1.3.8.0-hpe	3.2.0	0.7.0	3.2.0	3.2.0	2.3.9
		2.4.7	0.7.0-2.4.7			
5.5.0	1.3.7.1-hpe	3.2.0	0.7.0	3.2.0	3.2.0	2.3.9
		2.4.7	0.7.0-2.4.7			
5.4.1	1.2.2.0-hpe	3.1.2	0.7.0	3.1.2	3.1.2	2.3.8
		2.4.7	0.5.0			
5.4.0	1.2.2.0-hpe	3.1.2	0.7.0	3.1.2	3.1.2	2.3.8
		2.4.7	0.5.0			

*Starting from HPE Ezmeral Runtime Enterprise 5.5.0, you can choose to use Spark images provided by HPE Ezmeral Runtime Enterprise or your own open-source Spark images. Livy does not support open-source Spark images on HPE Ezmeral Runtime Enterprise. See Spark Operator on page 264 for details.

Spark Prerequisites

This topic describes the prerequisites to run Spark Applications on Kubernetes clusters in HPE Ezmeral Runtime Enterprise.

- Install HPE Ezmeral Runtime Enterprise.
- Apply the license to run Apache Spark Applications. See Product Licensing on page 87 and License Tab on page 798.
- Log in to HPE Ezmeral Runtime Enterprise as a Platform Administrator. To learn more about the users and their roles in HPE Ezmeral Runtime Enterprise, see Users and Roles on page 130.
- Configure Kubernetes cluster with one Master host and at least two Worker hosts. To submit a simple Spark Application, a single worker host must have a combined total of at least 3 vCPU cores and at least 4GB of RAM (recommended). The amount of RAM required depends on the number of Spark jobs running concurrently.
- (Optional) Configure AD/LDAP server to use the `autoticketgenerator` feature. See Spark Security on page 251.
- HPE Ezmeral Runtime Enterprise only supports the Spark operator on Kubernetes clusters created within the deployment. Spark Operator is not supported on imported Kubernetes clusters.

Preparing the Spark Environment

This topic describes how to prepare the environment to run Spark Applications.

To prepare the environment:

1. Log in to HPE Ezmeral Runtime Enterprise web interface as a Kubernetes Administrator. See Launching and Logging In.
2. Create a New Kubernetes Cluster and ensure that this cluster meets or exceeds the Spark Prerequisites on page 247. Ensure you have selected the **Enable Spark Operator** option during Kubernetes cluster creation step. See Installing and Configuring Spark Operator on page 265.
3. Create a New Kubernetes Tenant and do not assign any quotas when creating this tenant.
4. If the new Kubernetes tenant will use external LDAP/AD authentication, then see Kubernetes Tenant External Authentication.
5. Assign a tenant to the Tenant Administrator and depending on authentication selected, see Assigning/Revoking User Roles (Local) or Assigning/Revoking User Roles (LDAP/AD).
6. Log out of the HPE Ezmeral Runtime Enterprise web interface.

Related tasks

Installing and Configuring Spark Operator on page 265
This section describes how to install and configure Spark Operator on HPE Ezmeral Runtime Enterprise.

Spark Support

This topic describes the Spark enhancements and limitations for HPE Ezmeral Runtime Enterprise.

No Support for Apache Spark 3.1.1 Operator

In HPE Ezmeral Runtime Enterprise 5.4, the unified version of Apache Spark 3.x.x (default and recommended) replaces the preview version of Apache Spark 3.1.1 introduced in HPE Ezmeral Runtime Enterprise 5.3.

Hewlett Packard Enterprise no longer supports Apache Spark 3.1.1 and recommends you to move all your Spark applications to Apache Spark 3.x.x.

Spark Limitations

- When you configure the Amazon S3 to store event logs, Spark History Server does not show the list of running applications when you click **Show incomplete applications**. Amazon S3 doesn't support the append functionality and web UI only displays the list of completed application runs.
- Starting from HPE Ezmeral Runtime Enterprise 5.6.0, Spark 3.3.x and later versions support enhanced S3 features introduced in Hadoop 3.x.
- Starting from HPE Ezmeral Runtime Enterprise 5.5.0, Spark Operator supports open-source Spark images. You can now build your Spark with Hadoop 3 profile or any other profile of your choice. See Spark Operator on page 264 for details.
- Starting from HPE Ezmeral Runtime Enterprise 5.4.0, Livy does not support ZooKeeper for session recovery. Configure `pvc` or `disabled` option for `sessionRecovery` in all data-fabric (default) tenants or none tenants.

For Spark issues, see Issues and Workarounds on page 15.

Configuring Memory for Spark Applications

This topic describes how to set memory options for Spark applications.

You can configure the driver and executor memory options for the Spark applications by using HPE Ezmeral Runtime Enterprise new UI (see Creating Spark Applications on page 255) or by manually setting the following properties on Spark application YAML file.

- `spark.driver.memory`: Amount of memory allocated for the driver.
- `spark.executor.memory`: Amount of memory allocated for each executor that runs the task.

However, there is an added memory overhead of 10% of the configured driver or executor memory, but at least 384 MB. The memory overhead is per executor and driver. Thus, the total driver or executor memory includes the driver or executor memory and overhead.

*Memory Overhead = 0.1 * Driver or Executor Memory (minimum of 384 MB)*

Total Driver or Executor Memory = Driver or Executor Memory + Memory Overhead

Configuring Memory Overhead

You can configure the memory overhead for driver and executor by using Spark Operator, Livy, and `spark-submit` script.

Spark Operator	Set the following configurations options in the Spark application YAML file. See Spark application YAML.
	`spark.driver.memoryOverhead`
	`spark.executor.memoryOverhead`
	If you are using the HPE Ezmeral Runtime Enterprise new UI, add these configuration options by clicking **Edit YAML** in **Review** step or **Edit YAML** from Actions menu on **Spark Applications** screen. See Managing Spark Applications on page 260.

Livy	**Using YAML:**

Add the following configuration options in `spark-defaults.conf` section in `extraConfigs` section of `values.yaml` file in a tenant namespace.

```
extraConfigs:
 spark-defaults.conf:|
  spark.driver.memoryOverhead
<value-for-overhead>
  spark.executor.memoryOverhead
<value-for-overhead>
```

Using Rest APIs:

Add the following configuration options to `conf` section when creating a Livy session.

```
{
"name": "My interactive session",
"executorMemory": "512m",
"conf":
  {"spark.executor.memoryOverhead":
"1g" }
}
```

`spark-submit` Script

Specify the overhead configuration options using `--conf` flag and dynamically load properties:

```
./bin/spark-submit --name
"<spark-app-name>" --master
<master-url> --conf
spark.driver.memoryOverhead=<value>
```

```
./bin/spark-submit --name
"<spark-app-name>" --master
<master-url> --conf
spark.executor.memoryOverhead=<value>
```

To learn more about driver or executor memory, memory overhead, and other properties, see Apache Spark 2.x.x and Apache Spark 3.x.x application properties.

Spark Images

This topic lists the images that must be available to install and run Spark Operator, Apache Livy, Spark History Server, Spark Thrift Server, and Hive Metastore. These images enables you to run the Spark applications in an air-gapped environment.

Images for HPE Ezmeral Runtime Enterprise 5.6.1

Images for Spark 3.3.1

The following images are required in order to install and run Spark and Spark based services:

Spark Operator Images

```
gcr.io/mapr-252711/
spark-operator-1.3.8:1.3.8.2-1.3.8.2-h
pe
```

Spark Applications and Livy Session Images

```
gcr.io/mapr-252711/
spark-3.3.1:202301161621R
```

```
gcr.io/mapr-252711/
spark-py-3.3.1:202301161621R
gcr.io/mapr-252711/
spark-r-3.3.1:202301161621R
gcr.io/mapr-252711/
spark-gpu-3.3.1:202301161621R
```

Spark Based Services Images

```
gcr.io/mapr-252711/
spark-hs-3.3.1:202301161621R
gcr.io/mapr-252711/
spark-ts-3.3.1:202301161621R
gcr.io/mapr-252711/
livy-0.7.0:202301161621R
gcr.io/mapr-252711/
hivemeta-3.1.3:202301161621R
```

Images for Spark 2.4.7

The following images are required in order to install and run Spark and Spark based services:

Spark Operator Images

```
gcr.io/mapr-252711/
spark-operator-1.3.8:1.3.8.2-1.3.8.2-h
pe
```

Spark Applications and Livy Session Images

```
gcr.io/mapr-252711/
spark-2.4.7:202210110658R
gcr.io/mapr-252711/
spark-py-2.4.7:202210110658R
gcr.io/mapr-252711/
spark-r-2.4.7:202210110658R
```

Spark Based Services Images

```
gcr.io/mapr-252711/
spark-hs-2.4.7:202210110658R
gcr.io/mapr-252711/
spark-ts-2.4.7:202210110658R
gcr.io/mapr-252711/
livy-0.7.0-2.4.7:202301161621R
gcr.io/mapr-252711/
hivemeta-3.1.3:202301161621R
```

Images for HPE Ezmeral Runtime Enterprise 5.6.0

Images for Spark 3.3.1

The following images are required in order to install and run Spark and Spark based services:

Spark Operator Images

```
gcr.io/mapr-252711/
spark-operator-1.3.8:1.3.8.0-202211111
429
```

Spark Applications and Livy Session Images

```
gcr.io/mapr-252711/
spark-3.3.1:202212201209R
gcr.io/mapr-252711/
spark-py-3.3.1:202212201209R
```

```
gcr.io/mapr-252711/
spark-r-3.3.1:202212201209R
gcr.io/mapr-252711/
spark-gpu-3.3.1:202212201209R
```

Spark Based Services Images

```
gcr.io/mapr-252711/
spark-hs-3.3.1:202212201209R
gcr.io/mapr-252711/
spark-ts-3.3.1:202212201209R
gcr.io/mapr-252711/
livy-0.7.0:202212201209R
gcr.io/mapr-252711/
hivemeta-3.1.3:202212160615R
```

Images for Spark 2.4.7

The following images are required in order to install and run Spark and Spark based services:

Spark Operator Images

```
gcr.io/mapr-252711/
spark-operator-1.3.8:1.3.8.0-202211111
429
```

Spark Applications and Livy Session Images

```
gcr.io/mapr-252711/
spark-2.4.7:202210110658R
gcr.io/mapr-252711/
spark-py-2.4.7:202210110658R
gcr.io/mapr-252711/
spark-r-2.4.7:202210110658R
```

Spark Based Services Images

```
gcr.io/mapr-252711/
spark-hs-2.4.7:202212201209R
gcr.io/mapr-252711/
spark-ts-2.4.7:202212201209R
gcr.io/mapr-252711/
livy-0.7.0-2.4.7:202212201209R
gcr.io/mapr-252711/
hivemeta-3.1.3:202212160615R
```

Spark Security

This topic describes the Spark security concepts in HPE Ezmeral Runtime Enterprise.

Authentication for Spark on Kubernetes

Kubernetes authentication and authorization rules are applicable to Spark applications of kind `SparkApplication` or `ScheduledSparkApplication`.

For example: You can create, edit, delete, and submit the Spark applications according to RBAC configuration in a tenant namespace.

User Secrets

Spark application images are run as a root user. You must start Spark applications as a user who submits the Spark Application.

HPE Ezmeral Data Fabric is configured with Data Fabric SASL security. When you create the Spark applications in a Data Fabric which is HPE Ezmeral Data Fabric on Kubernetes tenant or in HPE Ezmeral Data Fabric on Bare Metal tenant, you must authenticate Spark driver pods against the HPE Ezmeral Data Fabric.

To start a Spark application as a user who submits the Spark application and to authenticate Spark driver pods against the Data Fabric, you must create a secret. A secret contains the user information like user id, user name, user's main group id and group name, and user's MapR ticket.

Creating User Secrets

You can create a user secret in three different ways:

Automatically creating secrets:

The `autoticketgenerator` webhook intercepts all the **Create Spark Application** requests.

The webhook automatically generates a ticket and secret when AD/LDAP integration is enabled on Data Fabric and Kubernetes cluster. This ticket has a default expiration time of 14 days.

You cannot change or renew the expiration time of ticket.

The generated secrets will be deleted when you delete the Spark Application.

Manually creating secrets with ticketcreator utility:

The Data Fabric tenants contain the tenantcli pod. You can manually create your user secrets using the `ticketcreator.sh` script in the tenantcli pod.

This ticket has a default expiration time of 14 days. You provide this ticket to the Spark applications using the secrets; thus, you cannot change or renew the expiration time of ticket.

Perform the following steps to use `ticketcreator.sh` script from `tenantcli` pod:

1. Run the following command to enter into `tenantcli` pod on tenant namespace.

```
kubectl exec -it tenantcli-0 -n
<namespace> -- bash
```

2. Run the `ticketcreator.sh` utility by using the following command:

```
 /opt/mapr/kubernetes/
ticketcreator.sh
```

3. Enter the following information on the prompt:
   a. The username and password of the user for whom to create the secret.
   b. The name of the user secret. The default name is randomized for security.

Add the secret name to `spark.mapr.user.secret` field on your Spark application `yaml` file.

Manually creating secret without ticketcreator utility:

Some Spark applications have a long runtime, for example, Spark streaming applications. In such cases,

you will lose the access to HPE Ezmeral Data Fabric services like HPE Ezmeral Data Fabric Filesystem in 14 days.

For the Spark applications which must run for a long time (greater than 14 days), you can create the ticket secrets with a longer expiration time using `-duration` option of the maprlogin utility. The maprlogin utility is available at Kubernetes cluster or at `tentantcli-0` pod and `admincli-0` pod at HPE Ezmeral Data Fabric on Kubernetes cluster. See Tickets and mapr Command Examples.

For example: If you have a ticket saved at `/home/user/maprticket` file, you can run the following command to manually create ticket secrets with a long expiration time:

```
kubectl -n <namespace> create secret
generic <secret-name> \
--from-file=CONTAINER_TICKET=/home/
user/maprticket \
--from-literal=MAPR_SPARK_USER="[usern
ame]" \
--from-literal=MAPR_SPARK_GROUP="[user
group]" \
--from-literal=MAPR_SPARK_UID="[uid]"
\
--from-literal=MAPR_SPARK_GID="[main_g
id]"
```

Add the secret name to `spark.mapr.user.secret` field on your Spark Application `yaml` file.

Updating Helm Charts for Spark Services

This topic describes how to update the Helm charts for Hive Metastore, Livy, Spark History Server, and Spark Thrift Server on HPE Ezmeral Runtime Enterprise.

Prerequisites

Install the Hive Metastore, Livy, Spark History Server, and Spark Thrift Server in your tenant namespace.

- Installing and Configuring Apache Livy on page 277
- Installing and Configuring Spark History Server on page 298
- Installing and Configuring Spark Thrift Server on page 305
- Installing and Configuring Hive Metastore on page 309

About this task

To update the Helm charts for Hive Metastore, Livy, Spark History Server, and Spark Thrift Server installed by using the GUI or manually by using the Helm charts, perform the following:

Procedure

1. List all the Helm charts in a tenant namespace.

```
helm list -n <tenant-namespace>
```

2. Select the Helm chart to update from the list and run:

```
helm upgrade <spark-services-helm--release-name> ./<path/to/
spark-services-helm-chart> -n <tenant-namespace>
```

Locate the Helm charts for different versions of HPE Ezmeral Runtime Enterprise at:

- Spark Helm Charts for HPE Ezmeral Runtime Enterprise 5.6.0
- Spark Helm Charts for HPE Ezmeral Runtime Enterprise 5.5.0
- Spark Helm Charts for HPE Ezmeral Runtime Enterprise 5.4.0 and 5.4.1

Results

Running the `helm upgrade` command updates the Helm chart for the Spark services.

Nvidia Spark-RAPIDS Accelerator for Spark

This topic describes Nvidia `spark-rapids` accelerator support for Spark.

Starting from HPE Ezmeral Runtime Enterprise 5.5.0, you can use RAPIDS Accelerator for Apache Spark by Nividia to accelerate the processing for Spark by using the GPUs.

For details on Spark support for requesting and scheduling GPUs for executors, see Custom Resource Scheduling and Configuration Overview.

The GPU image (spark-gpu-<spark-version>), for example: spark-gpu-3.2.0, has built-in RAPIDS plugin in HPE Ezmeral Runtime Enterprise.

To allocate GPUs and enable RAPIDS plugin, set `sparkConf` option. For configuration details, see Spark GPU Example.

NOTE: You must set `spark.rapids.force.caller.classloader` option to `false`.

For full list of configuration details for RAPIDS, see RAPIDS Configuration.

Submitting and Managing Spark Applications Using HPE Ezmeral Runtime Enterprise new UI

This section describes how to access HPE Ezmeral Runtime Enterprise new UI to create and monitor Spark applications.

Prerequisites

- Apply the license to run Apache Spark applications on the HPE Ezmeral Runtime Enterprise. See Product Licensing on page 87.

About this task

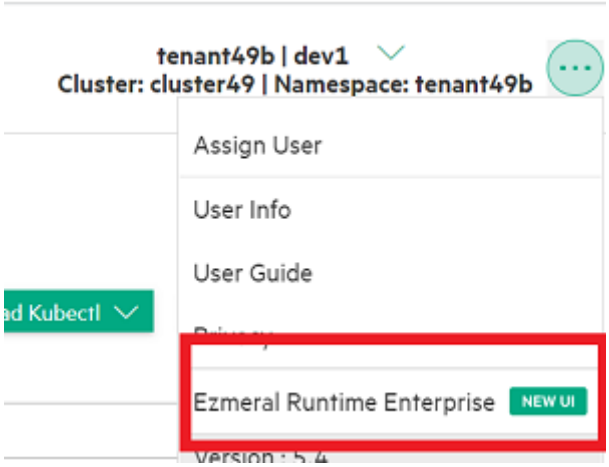
Starting from HPE Ezmeral Runtime Enterprise 5.4.0, you can create and submit Spark applications using the Spark Operator on the HPE Ezmeral Runtime Enterprise new UI. You can also view and monitor the resources and status of Spark applications using the GUI.

To access the GUI to manage Spark applications on HPE Ezmeral Runtime Enterprise, perform the following steps:

Procedure

1. Log in to HPE Ezmeral Runtime Enterprise as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member.

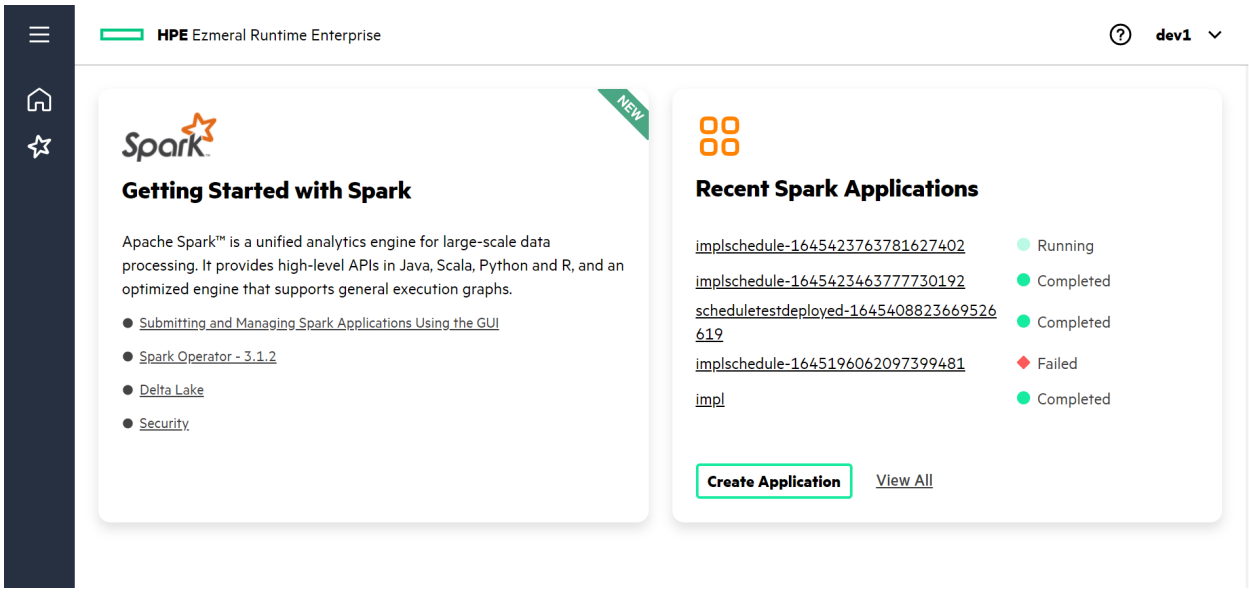
2. Click **horizontal three dots** menu on top banner of the screen.
3. Select **Ezmeral Runtime Enterprise**.



4. Log in to HPE Ezmeral Runtime Enterprise new UI as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member.

Results

You are now in the HPE Ezmeral Runtime Enterprise new UI and you can click **Create Application** to start creating Spark applications using the Spark Operator or click **View All** to view the list of previously created Spark applications.



Creating Spark Applications

This topic describes how to create Spark applications using the HPE Ezmeral Runtime Enterprise new UI.

Prerequisites

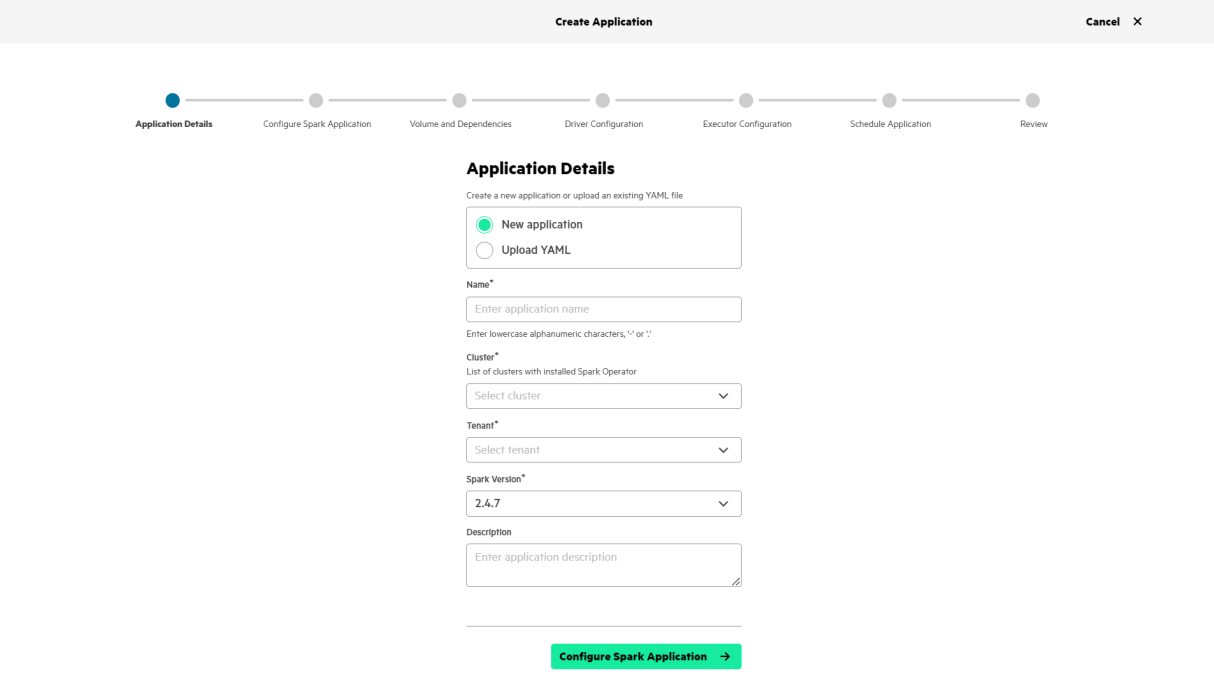
- Install Spark Operator on your Kubernetes cluster. See Spark Operator on page 264.

About this task

Create and submit Spark applications using the Spark Operator on HPE Ezmeral Runtime Enterprise new UI.

Procedure

1. To start creating Spark applications, click **Create Application** on the HPE Ezmeral Runtime Enterprise new UI screen or Spark Applications screen. Navigate through each step within the **Create Application** wizard:



a) **Application Details**: Create an application or upload a preconfigured YAML file. Set the following boxes:

YAML File	When you select the **Upload YAML**, you can upload a preconfigured YAML file from your local system. Click **Select File** to upload the YAML. The fields in the wizard are populated with the information from YAML.
Name	Enter the application name.
Cluster:	Select the cluster. The drop-down menu lists the clusters on which the Spark Operator is installed.
Tenant:	Select the tenant. Your Spark applications will run on this tenant.
Spark Version:	Select your preferred Spark version.
Description:	Enter the application description.

b) **Configure Spark Application**: Set the following boxes:

Type:	Select the application type from Java, Scala, Python, or R.
Image:	**Image** is auto filled based on the selected **Type** and **Spark Version**. There are different images for Spark, and different application types (Java

	or Scala, Python, and R). See Spark Images on page 249.
Image Pull Secret:	**Image Pull Secret** is preconfigured to the default value of `imagepull`.
Source:	Select the data source from **MapRFS**, **DataTap**, **S3**, and **Other**. **NOTE:** Open-source Spark images do not support **MapRFS**. Select **Other** as the data source to reference other locations of the application file. For example, to refer to a file inside the specific Spark image, use the `local` schema. To use **S3** as the data source, enter the S3 endpoint and (optional) Secret. To create a Secret containing the S3 credentials (user name and password), see Adding S3A Credentials Using a Kubernetes Secret on page 271.
Filename:	Enter location and file name of the application. For example: `s3a://apps/my_application.jar`
Class Name:	Enter main class of the application for Java or Scala applications.
Arguments:	Click **+ Add Argument** to add input parameters as required by the application.
Log Spark Events:	To enable logging of Spark events, check **Log Spark Events** check box. You can view the Spark events log by using Spark History Server. To disable the logging of Spark events, clear the check box. You must disable logging in the following scenarios: • When you have not installed Spark History Server in tenants. • When you are using open-source Spark images and have configured `maprfs` as the event log storage for Spark History Server.

c) **Volume and Dependencies:** Configure a volume and add dependencies in **Volume and Dependencies** step.

To configure a volume accessed by your application, toggle **Configure Volume**.

Set the following boxes:

Name	Enter volume name.
Type	Choose a volume type: • **ConfigMap**: Enter ConfigMap name in **ConfigMap Name** box. • **PersistentVolumeClaim**: Enter PersistentVolumeClaim name in **PersistentVolumeClaim Name** box.

To configure multiple volumes of different type, upload the preconfigured YAML file in the **Application Details** step or edit the YAML file in the **Review** step.

NOTE: When you upload a YAML file, the volume configurations are preserved and you can view it on **Review** step. If you choose to configure volume using the **Configure Volume**, it will override any previous volume configurations in the YAML file.

To add dependencies required to run your applications, click **Add Dependency**. Select a dependency type from excludePackages, files, jars, packages, pyfiles, or repositories, and enter the value of the dependency.

For example:

- Enter the package names as the values for the excludePackages dependency type.
- Enter the locations of file, for example, dtap://<path-to-file>, s3://<path-to file>, local:// <path-to-file> as the values for files, jars, pyfiles, or repositories.

d) **Driver Configuration:** Configure the number of cores, core limits, memory, and service account. The number of cores must be less than or equal to the core limit.

If a Platform Administrator configured a tenant with a CPU quota, you must set the core limit for the driver pods.

If a Platform Administrator configured a tenant with a memory quota, you must set the memory for the driver pods. See Configuring Memory for Spark Applications on page 248.

If you configured a volume in **Volume and Dependencies** step, you get an option to mount the volume in **Driver Configuration** step. To mount the volume, toggle **Configure Volume Mount**.

Name	Set with volume name from **Volume and Dependencies** step.
Path	Enter the mount path for the volume in the driver pod.

e) **Executor Configuration:** Configure the number of executors, number of cores, core limits, memory, and service account. The number of cores must be less than or equal to the core limit.

If a Platform Administrator configured a tenant with a CPU quota, you must set the core limit for the executor pods.

If a Platform Administrator configured a tenant with a memory quota, you must set the memory for the executor pods. See Configuring Memory for Spark Applications on page 248.

If you configured a volume in **Volume and Dependencies** step, you get an option to mount the volume in **Executor Configuration** step. To mount the volume, toggle **Configure Volume Mount**.

Name	Set with volume name from **Volume and Dependencies** step.
Path	Enter the mount path for the volume in the executor pods.

f) **Schedule Application:** To schedule a Spark application to run at a certain time, toggle **Schedule to Run**. You can configure the frequency intervals and set the concurrency policy, successful run history limit, and failed run history limit.

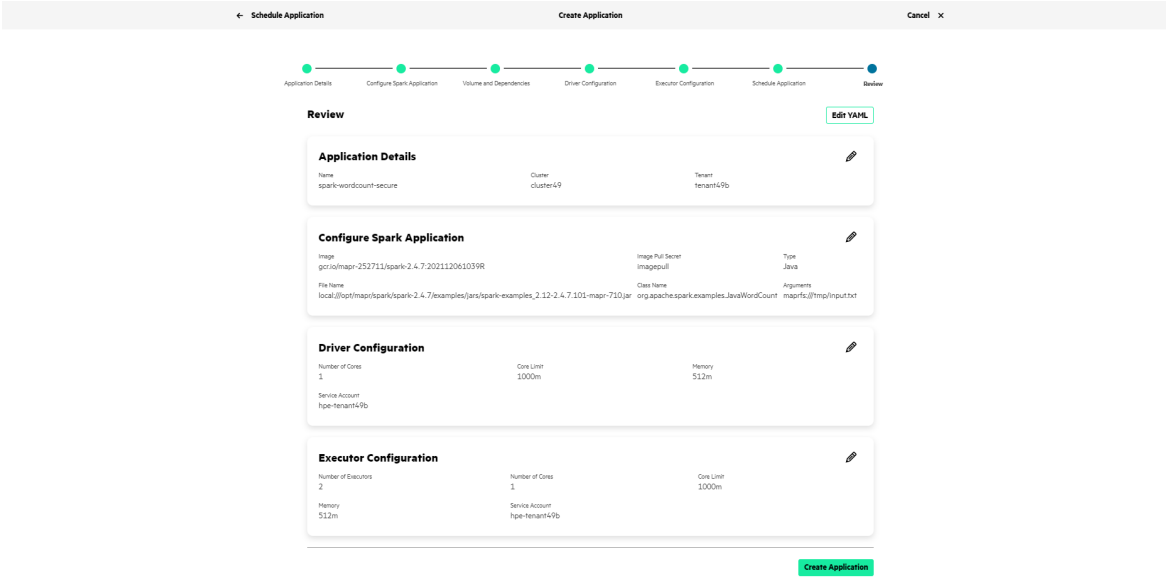
Set the **Frequency Interval** in two ways:

1. To choose from predefined intervals, select **Predefined Frequency Interval** and click **Update** to open a dialog with predefined intervals.

2. To set the frequency interval, select **Custom Frequency Interval**. The **Frequency Interval** accepts any of the following values:
   - CRON expression with
     - Field 1: minute (0–59)
     - Field 2: hour (0–23)
     - Field 3: day of the month (1–31)
     - Field 4: month (1–12, JAN - DEC)
     - Field 5: day of the week (0–6, SUN - SAT)
     - Example: `0 1 1 * *`, `02 02 ? * WED, THU`
   - Predefined macro
     - @yearly
     - @monthly
     - @weekly
     - @daily
     - @hourly
   - Interval using @every <duration>
     - Units: nanosecond (ns), microsecond (us, µs), millisecond (ms), second (s), minute (m), and hour (h).
     - Example: `@every 1h`, `@every 1h30m10s`

g) **Review:** Review the application details. Click the **pencil icon** in each section to navigate to the specific step to change the application configuration.

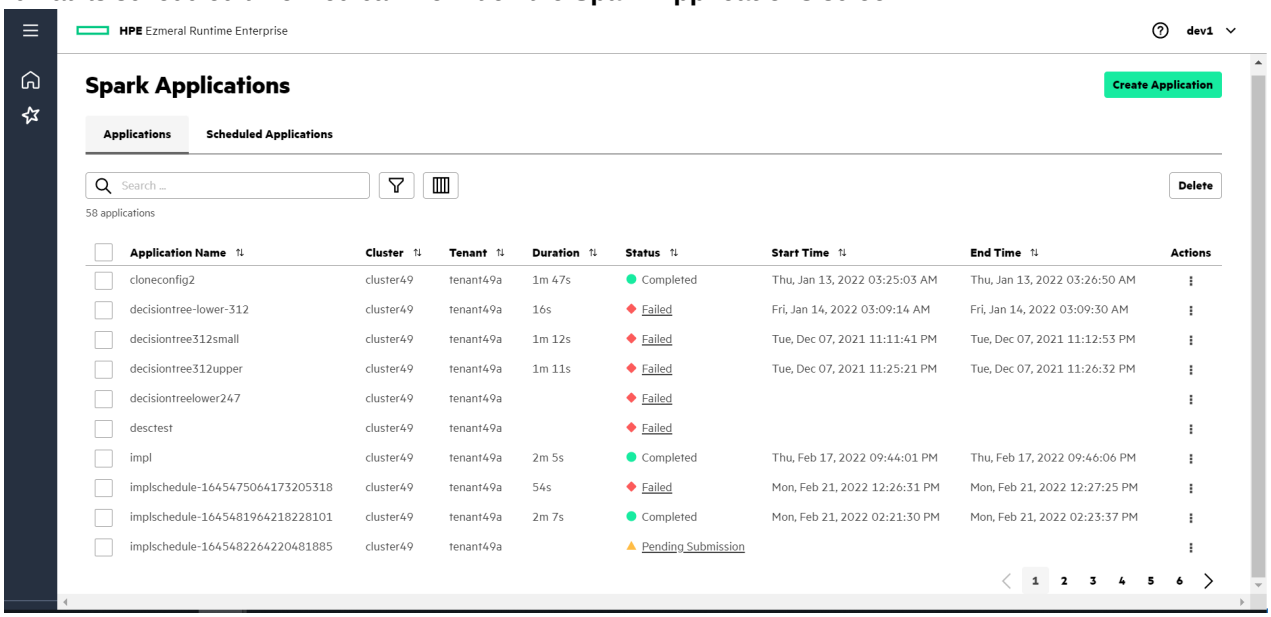
   To open an editor to change the application configuration using YAML in the GUI, click **Edit YAML**. You can use the editor to add the extra configuration options not available through the application wizard. To apply the changes, click **Save Changes**. To cancel the changes, click **Discard Changes**.



2. To submit the YAML to run on a selected tenant, click **Create Application** on the bottom right of the **Review** step.

Results

The GUI creates and immediately runs a Spark application or waits for a scheduled Spark application to run at its scheduled time. You can view it on the **Spark Applications** screen.



Managing Spark Applications

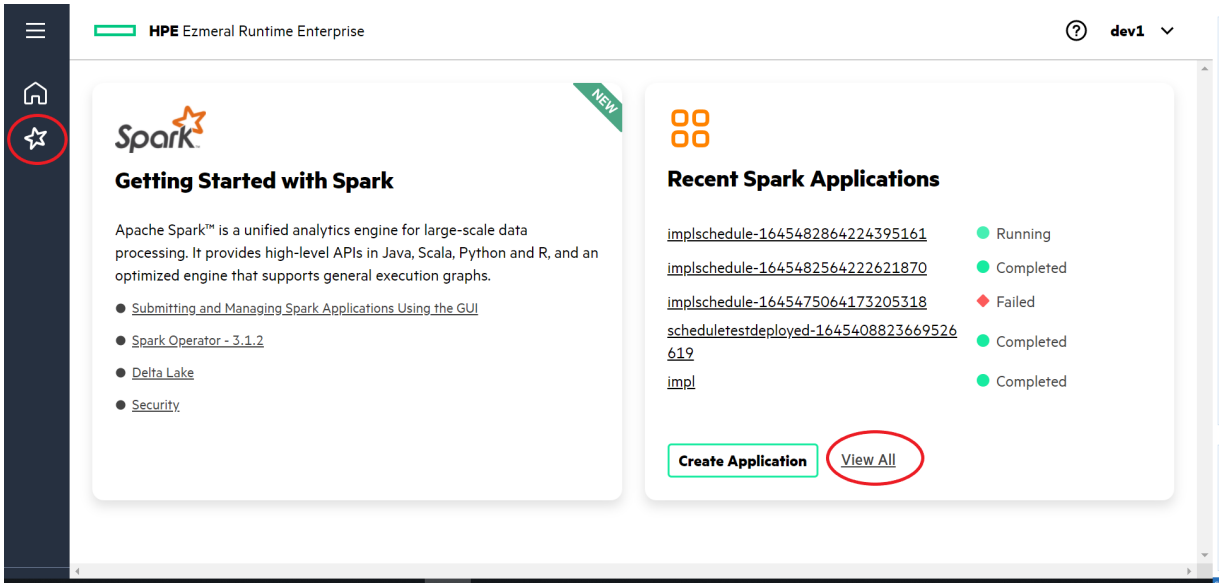
This topic describes how to view and manage Spark applications using HPE Ezmeral Runtime Enterprise new UI.

About this task

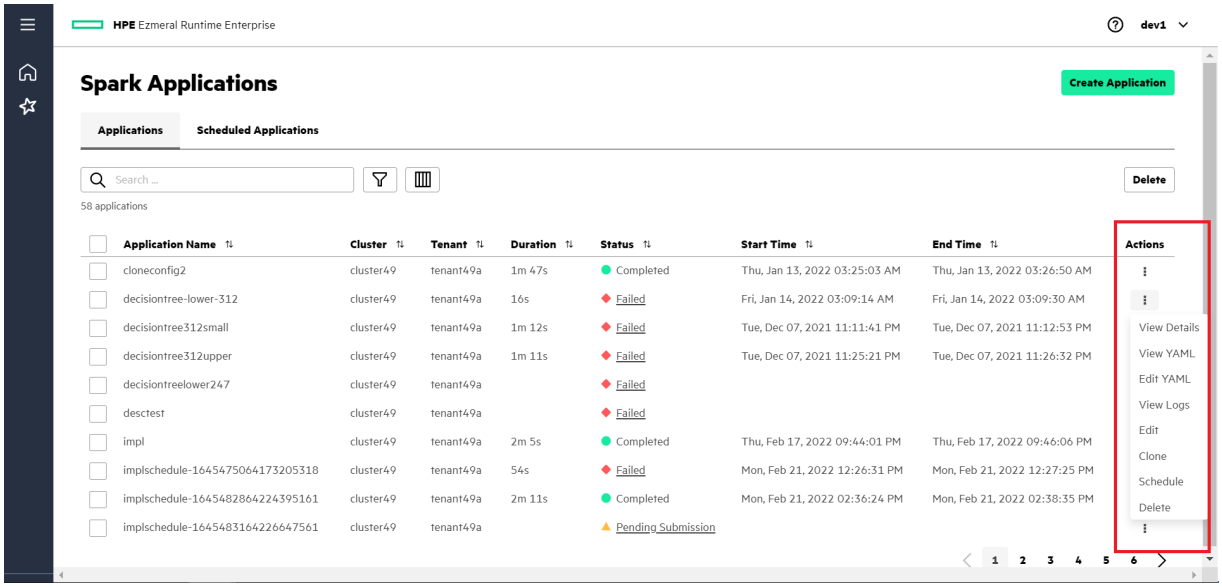
View and manage the status of all the Spark applications and scheduled Spark applications.

Procedure

1. Click the **Spark icon** on the left navigation bar or click **View All** in HPE Ezmeral Runtime Enterprise new UI home page.



2. To view actions you can perform on **Applications** and **Scheduled Applications** tab, click the **menu icon** in the **Actions** column.



View Details:

To view the details of an application, including CPU, memory usage, and events and logs of the pods, select **View Details**.

To access the Spark History Server and view and monitor the applications, click **Spark Web UI** in the top right of the **Application Detail** screen.

NOTE: To view the **Spark Web UI** link, ensure you have installed the Spark History Server in your tenant.

View YAML:

To view the YAML file and see the configuration details, select **View YAML**.

Edit YAML:

To open an editor to change the application configuration using a YAML in the GUI, click **Edit YAML**. You can use the editor to add the extra configuration options not available through application wizard. To apply the changes, click **Update Application**. To cancel the changes, click **Discard Changes**.

View Logs:

To view the Spark driver pod logs, select **View Logs**.

Edit:

To change application configurations and resubmit the application, select **Edit**.

You can use the editor to add the extra configuration options not available through application wizard.

You can update all the application parameters except name, cluster, tenant, Spark version, and type using **Edit**. Use **Clone** to update the parameters and create a new application.

You can update the schedule of the scheduled Spark application by using the **Edit**.

To open an editor to change the application configuration using YAML in the GUI, click **Edit YAML** in **Review** step. To apply the changes, click **Save Changes**. To cancel the changes, click **Discard Changes**.

To schedule the Spark application, select **Schedule** or select **Clone**.

NOTE: Using **Edit** to resubmit an application will remove pods and logs of the previous application run.

Clone:

To create a new Spark application with the similar configuration as an existing Spark application, select **Clone**. You can update any application parameters and submit it as a new application.

NOTE:

If you enter the same name as the current Spark application in the same tenant and configure the scheduling details on **Schedule Application** step, it will create a new scheduled Spark application.

Submitting an application in the same tenant with same name and application type as an existing application will remove pods and logs of the previous application run.

Schedule:

To schedule the application, click **Schedule**. You can view this application in the **Scheduled Applications** tab. To learn more about the **Schedule Application** step, see Creating Spark Applications on page 255.

Suspend:

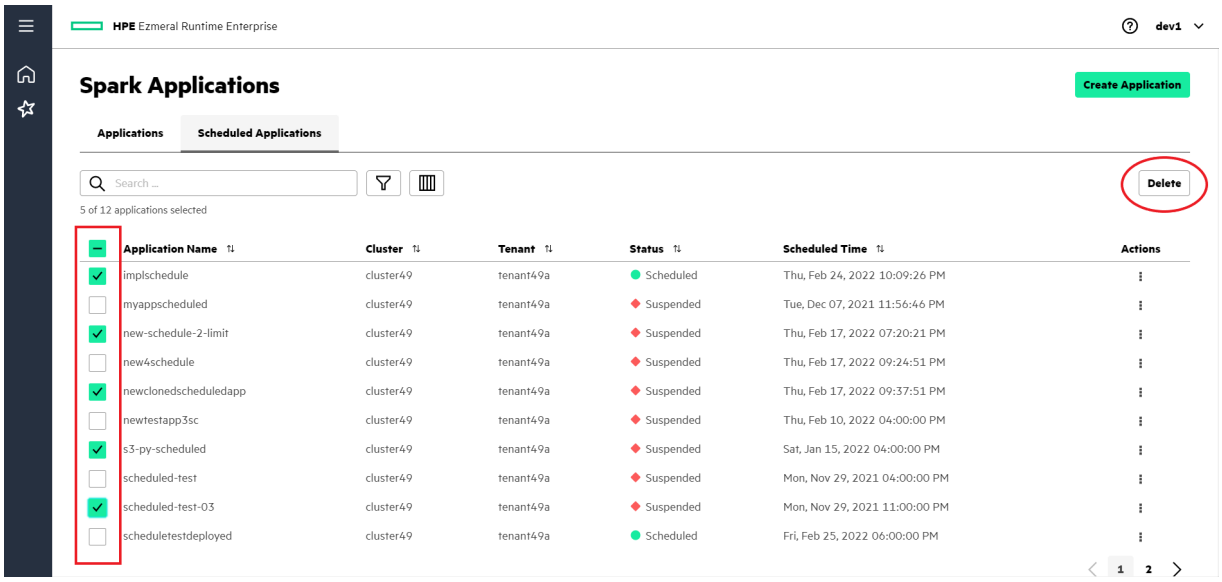
To stop the application from running at its scheduled time, select **Suspend** from the **Actions** menu in the **Scheduled Applications** tab.

Resume:

To restart the schedule of the suspended applications, select **Resume** from Actions menu on **Scheduled Applications** tab.

Delete: To delete the Spark application, select **Delete**.

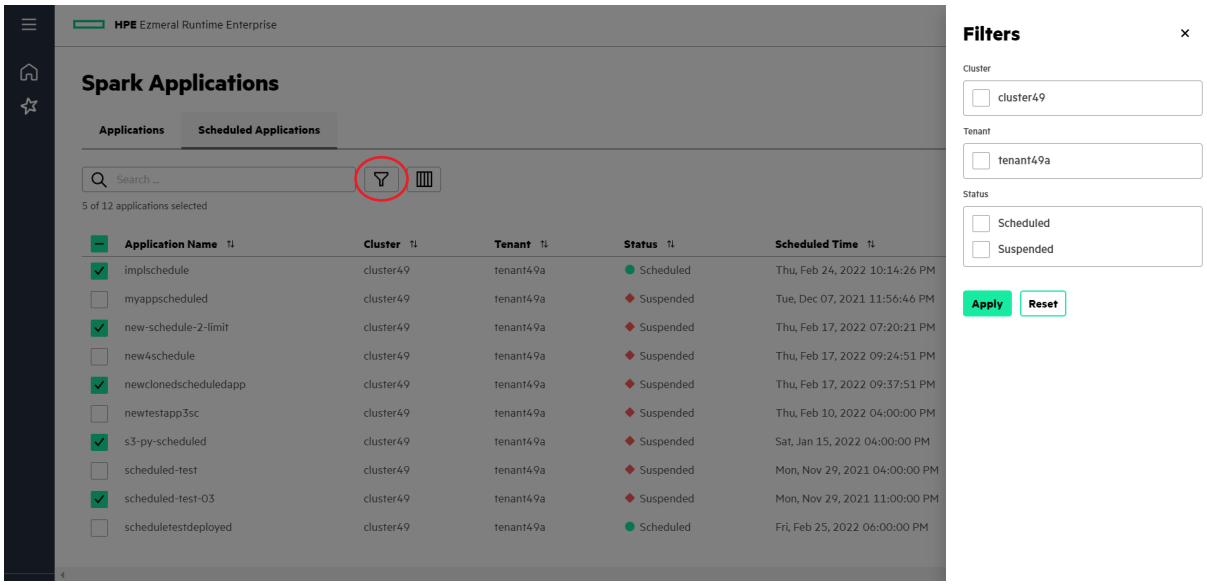
3. Delete multiple Spark applications at once:



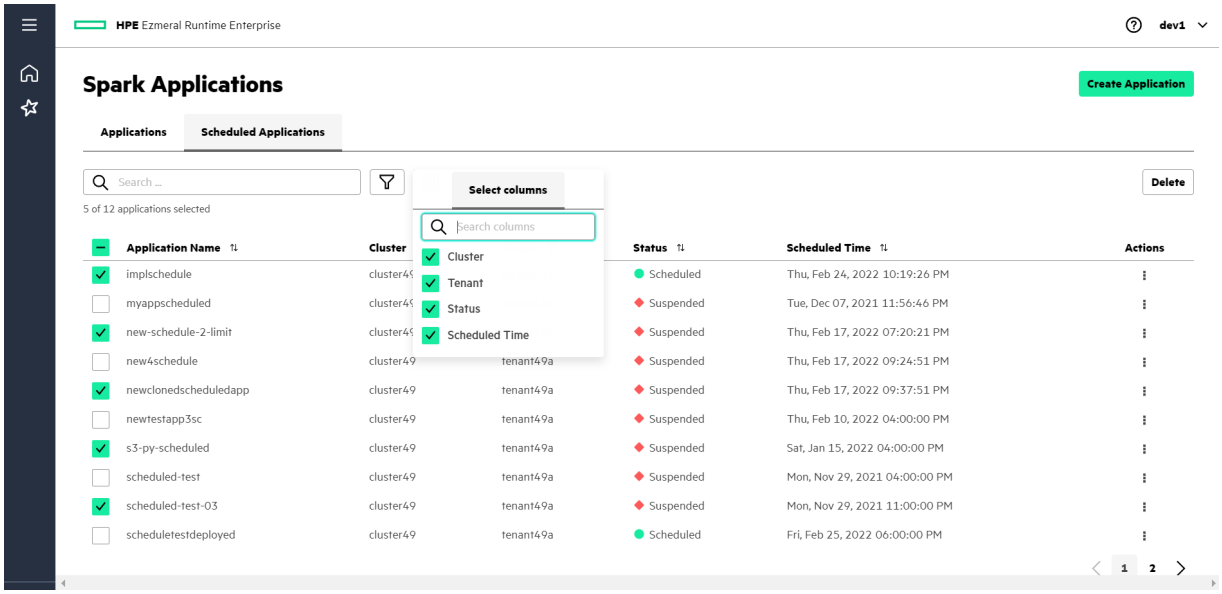
a) To select multiple applications, click the check box besides **Application Name** in the table.

b) Click **Delete** on the top right pane of the table.

4. To display the Spark applications according to the clusters, tenants, and status, click **Filter** icon.



5. To select the columns to display on your applications table, click **Columns** icon.



Spark Operator

This topic provides an overview of Spark Operator on HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise 5.4.0 and later supports multiversion Spark Operator. You can submit Spark Applications for different versions of Apache Spark using a single Spark Operator. When you submit the Spark Applications, Spark Operator creates a Kubernetes `spark-submit` job. The `spark-submit` job spawns the driver pod. A Spark driver pod launches a set of Spark executors that execute the job you want to run.

Starting from HPE Ezmeral Runtime Enterprise 5.6.0, Spark 3.3.x and later versions support enhanced S3 features introduced in Hadoop 3.x.

Starting from HPE Ezmeral Runtime Enterprise 5.5.0, you can choose to use Spark images provided by HPE Ezmeral Runtime Enterprise or your own open-source Spark images.

Spark Operator supports open-source Spark version compatible with the Kubernetes version supported on HPE Ezmeral Runtime Enterprise. With the support for open-source Spark, you can build your Spark with Hadoop 3 profile or any other profile of your choice.

You can integrate open-source Spark with Spark History Server by using PVC.

To use open-source Spark, build Spark and then build Spark images to run in HPE Ezmeral Runtime Enterprise. See Building Spark and Building Images.

However, open-source Spark does not support the following:

- Data Fabric filesystem, Data Fabric Streams, and any other Data Fabric sources and sinks which require Data Fabric client.
- Data Fabric specific security features (Data Fabric SASL).

NOTE: Livy does not support open-source Spark images on HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise supports all the features and parameters supported by open-source Spark on K8s documentation excluding the security feature. HPE Ezmeral Runtime Enterprise supports the following Spark security features:

- If you are a local user, set the `spark.mapr.user.secret` option on your Spark application `yaml` file.
- If you are AD/LDAP user, `spark.mapr.user.secret` option is automatically set using the ticketgenerator webhook.

- You must not change the user context. See using pod security context.

Related tasks

Submitting and Managing Spark Applications Using HPE Ezmeral Runtime Enterprise new UI on page 254
This section describes how to access HPE Ezmeral Runtime Enterprise new UI to create and monitor Spark applications.

Installing and Configuring Spark Operator

This section describes how to install and configure Spark Operator on HPE Ezmeral Runtime Enterprise.

Prerequisites

- Log in as a Kubernetes Cluster Administrator or Platform Administrator in HPE Ezmeral Runtime Enterprise.

About this task

In HPE Ezmeral Runtime Enterprise, you can install Spark Operator using GUI or manually using the Helm chart.

Learn more about supported Spark versions by Spark Operator at Interoperability Matrix for Spark on page 246.

Installing Spark Operator Using the GUI

About this task

Install Spark Operator during the Kubernetes Cluster creation step using the HPE Ezmeral Runtime Enterprise GUI. See Creating a New Kubernetes Cluster on page 463.

Procedure

1. Set up **Host Configurations**, **Cluster Configurations**, and **Authentication** for Kubernetes Cluster.
2. In **Application Configurations**, select **Enable Spark Operator**.
3. Click **Next**, review the summary of resources to be assigned to Kubernetes cluster.
4. To install the Kubernetes cluster, click **Submit**. This triggers the installation process for Spark Operator.

Results

The GUI installs the Spark Operator.

Starting from HPE Ezmeral Runtime Enterprise 5.4.0, selecting **Enable Spark Operator** option does not trigger the installation for Livy, Spark History Server, Spark Thrift Server, and Hive Metastore. You must install Livy, Spark History Server, Spark Thrift Server, and Hive Metastore using the GUI or manually using the helm charts.

Installing Spark Operator Using the Helm

Prerequisites

1. Install and configure Helm 3.
2. Install and configure kubectl.

About this task

Install the Spark Operator by using the Helm chart. See Apache Spark Operator Chart.

Procedure

To install the Spark Operator for Apache Spark in an existing namespace, run the following command:

```
helm install -f <path-to-values.yaml-file> <spark-operator-name> ./
<path-to-spark-operator-chart>/ \
--namespace <cluster-namespace> \
--set sparkJobNamespace=<cluster-namespace> \
--set webhook.namespaceSelector=hpe.com/tenant=<cluster-namespace> \
--set fullnameOverride=<spark-operator-name>
```

Autoticket generator webhook is not installed by default in the cluster namespace. To enable the installation of autoticket generator, set the following flag:

```
--set autotix.enable=true
```

Running the `helm install` installs Spark Operator in a cluster namespace.

Example

To install Spark Operator in the compute namespace, run the following command:

```
helm install -f spark-operator-chart/values.yaml spark-operator-compute ./
spark-operator-chart/ \
--namespace compute \
--set sparkJobNamespace=compute \
--set webhook.namespaceSelector=hpe.com/tenant=compute \
--set fullnameOverride=spark-operator-compute
```

Related tasks

Installing and Configuring Apache Livy on page 277
This section describes how to install and configure Apache Livy on HPE Ezmeral Runtime Enterprise.

Installing and Configuring Spark History Server on page 298
This section describes how to install and configure Spark History Server on HPE Ezmeral Runtime Enterprise.

Installing and Configuring Spark Thrift Server on page 305
This section describes how to install and configure Spark Thrift Server on HPE Ezmeral Runtime Enterprise.

Installing and Configuring Hive Metastore on page 309
This section describes how to install and configure Hive Metastore on HPE Ezmeral Runtime Enterprise.

More information

Interoperability Matrix for Spark on page 246
This section provides information about support and interoperability for Spark and its components with HPE Ezmeral Runtime Enterprise.

Setting Custom TrustStore

This topic describes how to set custom trustStore for SSL encryption using Spark Operator.

A Java trustStore is a repository to store the certificates from Certified Authorities (CA). CA verifies the certificate presented by the server in an SSL connection.

To set the custom trustStore, add the following configuration options to driver and executor options of `spec` section of the Spark application configuration:

```
driver:
  javaOptions: "-Djavax.net.ssl.trustStore=<path-to-custom-trustStore>"
  volumeMounts:
    - name: truststore
```

```
      mountPath: <path-to-custom-truststore>
executor:
  javaOptions: "-Djavax.net.ssl.trustStore=<path-to-custom-trustStore>"
  volumeMounts:
    - name: truststore
      mountPath: <path-to-custom-truststore>
```

The sample path to custom trustStore is `/opt/mapr/spark/spark-3.1.2/truststore`.

For example: To access the Amazon S3 buckets using SSL, you must add the following configuration options in the `spec` section of the Spark application configuration. The default Java trustStore `/etc/pki/java/cacerts` contains Amazon CA.

```
driver:
  javaOptions: "-Djavax.net.ssl.trustStore=/etc/pki/java/cacerts"
executor:
  javaOptions: "-Djavax.net.ssl.trustStore=/etc/pki/java/cacerts"
```

Submitting Spark Applications

This section describes how to submit the Spark applications using the Spark Operator on HPE Ezmeral Runtime Enterprise.

About this task

Spark resources are created in the tenant namespace managed by Kubernetes. When you submit the Spark applications, Spark Operator creates a Kubernetes `spark-submit` job. The `spark-submit` job spawns the driver pod and driver creates the executor pods. The driver pod remains in the **Completed** or **Error** state and executor pods terminate after the completion or failure of Spark applications. The driver pod does not consume any Kubernetes resources in the **Completed** state and now you can view the logs to see execution details and results.

To create and submit the Spark applications using the HPE Ezmeral Runtime Enterprise new UI, see Submitting and Managing Spark Applications Using HPE Ezmeral Runtime Enterprise new UI on page 254.

To manually create and submit the Spark applications using the Spark Operator in HPE Ezmeral Runtime Enterprise, perform the following steps:

1. Log in to HPE Ezmeral Runtime Enterprise as the Kubernetes Tenant Administrator or a Kubernetes Tenant Member. See Assigning/Revoking User Roles (Local) for local users or Assigning/Revoking User Roles (LDAP/AD) for LDAP/AD users.

2. If you are a local user or if you have not enabled LDAP/AD, you must use `ticketcreator.sh` script from tenantcli pod to create the ticket secrets. Add the secret name to `spark.mapr.user.secret` field on your Spark application YAML file (for example, `spark-wc.yaml`). See Spark Security on page 251.

3. Create a specification in `yaml` format to store all the necessary configurations required for the application.

   For example: Spark 3.3.1 Wordcount Example for HPE Ezmeral Runtime Enterprise 5.6.

   The Spark application specification is defined as kind `SparkApplication` or `ScheduledSparkApplication`, seeSpark application CRDs.

4. Upload the application file, for example application JAR, Python, or R files, in the FsMounts, DataTaps, or S3 location in the cluster.

5. To run the `kubectl` commands, access the Kubernetes Web Terminal on HPE Ezmeral Runtime Enterprise GUI or configure the `kubectl` on your local machine, see Using the HPE Kubectl Plugin on page 353. If you are running `kubectl` from your local machine, you can store the `yaml` file on your local machine.

6. Create a Spark application from YAML file by running the following `kubectl` command:

```
kubectl apply -f /<path-to-spark-job-yaml-file> -n <tenant_namespace>
```

Results

Spark Operator receives and submits the configured Spark applications to run on the Kubernetes cluster.

Example

To run the Spark application to count the words in a file using FsMounts as the file system storage, perform the following steps:

1. Log in to HPE Ezmeral Runtime Enterprise as the Kubernetes Tenant Administrator or a Kubernetes Tenant Member.

2. In the **FsMounts** screen, click the **TenantShare** link in the **Name** column of the table to open the **Data Source Browser** screen.

3. Create the `data` and `apps` subdirectory in the **TenantShare** filesystem mount.

4. Create a text file or download `wordcount.txt` example file from wordcount GitHub.

5. Upload the `wordcount.txt` file to the `data` subdirectory. Navigate to subdirectory in HPE Ezmeral Runtime Enterprise by `/hcp/tenant-<tenant_id>/fsmount/data/wordcount.txt`.

6. Download the `spark-wc.yaml` file. For example:

   - Spark 2.4.7 Wordcount Example for HPE Ezmeral Runtime Enterprise 5.6
   - Spark 3.3.1 Wordcount Example for HPE Ezmeral Runtime Enterprise 5.6

   To locate Spark examples for other versions of HPE Ezmeral Runtime Enterprise, navigate to the release branch of your choice at Spark on K8s GitHub location and find the examples in `examples` folder.

7. Update the namespace on YAML file to `<tenant-namespace>`, input filename to `spark-wordcount`, and add path to `wordcount.txt` to arguments field as `- maprfs:///hcp/tenant-<tenant_id>/fsmount/data/workdcount.txt`.

8. Upload the wordcount YAML at `/bd-fs-mnt/TenantShare/apps/` location in HPE Ezmeral Runtime Enterprise GUI as `spark-wc.yaml` file.

9. To run the `kubectl` commands, access the Kubernetes Web Terminal on HPE Ezmeral Runtime Enterprise GUI or configure the kubectl on your local machine, see Using the HPE Kubectl Plugin on page 353.

10. To run the Spark wordcount (wordcount.txt) example, execute:

```
kubectl apply -f /bd-fs-mnt/TenantShare/apps/spark-wc.yaml -n
<tenant_namespace>
```

You will get the following output:

```
sparkapplication.sparkoperator.hpe.com/spark-wordcount created
```

11. To check the pods running within the tenant namespace, run:

```
kubectl get pods -n <tenant-namespace>
```

You will get the following output:

```
NAME                                READY     STATUS       RESTARTS     AGE
hivemeta-9b4c8cfb5-hgbjf             1/1      Running         7         23h
spark-wordcount-driver              1/1      Running         0         13m
sparkhs-7bfb88bc4-m6b54              1/1      Running         6         23h
tenantcli-0                          0/1      Running         0         23h
```

After the job completion, the status will change to **Completed**.

12. To show the logs of the driver pod for the submitted Spark applications, run:

```
kubectl logs spark-wordcount-driver --follow -n <tenant-namespace>
```

Related tasks

Submitting and Managing Spark Applications Using HPE Ezmeral Runtime Enterprise new UI on page 254
This section describes how to access HPE Ezmeral Runtime Enterprise new UI to create and monitor Spark applications.

More information

Spark 3.3.1 DataTap Wordcount Example for HPE Ezmeral Runtime Enterprise 5.6

Spark 2.4.7 DataTap Wordcount Example for HPE Ezmeral Runtime Enterprise 5.6

Deleting and Resubmitting the Spark Applications

This section describes how to resubmit and delete the Spark applications using the Spark Operator on HPE Ezmeral Runtime Enterprise.

Prerequisites

Log in as Kubernetes Tenant Administrator.

About this task

To resubmit the same Spark application, delete the previously submitted Spark application and wait until Kubernetes cleans the driver pod. Failing to delete Spark application before resubmission will output the following message:

```
sparkapplication.sparkoperator.hpe.com/<spark-application-name> unchanged
```

Procedure

Run the following command to delete the Spark application:

```
kubectl delete -f /<path-to-spark-application-yaml-name>
```

To delete Spark applications using HPE Ezmeral Runtime Enterprise new UI, see Managing Spark Applications on page 260.

Sample Spark Applications

This topic describes how to locate the sample Spark Applications to run it using Spark Operator.

Some ready-to-use sample Spark Applications are built into the container image. These applications are located at `/opt/mapr/spark/spark-[version]/jars/spark-examples_[full-version].jar` and should be referenced using the `local` schema. You may also build your own applications and make them available in `/opt/mapr`.

If the Spark Application is located elsewhere, then modify the `mainApplicationFile` field to point to that storage and interface.

The Spark Operator supports the `http://`, `maprfs:///`, `dtap://`, and `S3://` interfaces.

Here are some use-cases for invoking a custom Spark job via the Spark Operator:

- Build an image, place the `.jar` in `/apps/`, and then reference it as follows:

  ```
  mainApplicationFile: "local:///apps/my_app.jar"
  ```

- If the application is stored and shared from HPE Ezmeral Data Fabric, then you can reference it by specifying `maprfs:///` or `S3://` in `mainApplicationFIle`. For example:

  ```
  maprfs:///path/to/my_app.jar
  S3://path/to/my_app.jar
  ```

- If the application is stored on the web server, then set `mainApplicationFIle` as follows:

  ```
  http://host:port/path/to/my_app.jar
  ```

Securely Passing Spark Configuration Values

This section describes how to pass the sensitive data to Spark configuration using the Kubernetes Secret.

About this task

You can pass the sensitive data which are part of the Spark configuration using the Kubernetes secret. The secret has a Key-Value format where the key is `spark-defaults.conf` file and the value is sensitive data.

Procedure

1. Create a Kubernetes Secret with the key as `spark-defaults.conf` and the value as sensitive data. See Creating a Secret.

2. Add `spark.mapr.extraconf.secret` option with value as Secret name on Spark application YAML.

Example

1. To securely pass the sensitive data, create a file with Spark configuration properties :

```
cat << EOF > spark-defaults.conf
spark.hadoop.fs.s3a.access.key EXAMPLE_ACCESS_KEY
spark.hadoop.fs.s3a.secret.key EXAMPLE_SECRET_KEY
EOF
```

2. Create a Secret from the file:

```
kubectl create secret generic
<k8s-secret-name> --from-file=spark-defaults.conf
```

3. Set the `spark.mapr.extraconf.secret` option with Secret name in Spark application YAML.

```
...
spec:
  sparkConf:
    spark.mapr.extraconf.secret: "<k8s-secret-name>"
...
```

Accessing Data on Amazon S3 Using Spark Operator

This topic describes how to access the data on Amazon S3 bucket using a Hadoop S3A Client.

Amazon Web Services (AWS) offers Amazon Simple Storage Service (Amazon S3). Amazon S3 provides the storage and retrieval of objects through a web service interface.

You can access the data stored on Amazon S3 bucket for your Spark job by using a Hadoop S3A Client. For the full list of Hadoop S3A Client configuration options, see Hadoop-AWS module: Integration with Amazon Web Services.

Adding S3A Credentials through YAML

Add the following configuration options on `sparkConf` section of SparkApplication and submit the spark jobs using Spark Operator.

```
spark.hadoop.fs.s3a.access.key <YOURACCESSKEY>
spark.hadoop.fs.s3a.secret.key <YOURSECRETKEY>
```

For example:

```
apiVersion: "sparkoperator.k8s.io/v1beta2"
kind: SparkApplication
metadata:
  name: spark-s3-example
spec:
  sparkConf:
    # ...
    spark.hadoop.fs.s3a.access.key: <YOURACCESSKEY>
    spark.hadoop.fs.s3a.secret.key: <YOURSECRETKEY>
```

Adding S3A Credentials Using a Kubernetes Secret

A Secret is an object that contains the sensitive data such as a password, a token, or a key. See Secrets.

You can access the data stored on Amazon S3 bucket for your Spark job by using the Kubernetes Secret.

Creating a Secret

Create the Kubernetes Secret with Base64-encoded values for AWS_ACCESS_KEY_ID (username) and AWS_SECRET_ACCESS_KEY (password).

For example: Run `kubectl apply -f` for the following YAML:

```
apiVersion: v1
kind: Secret
data:
  AWS_ACCESS_KEY_ID: <Base64-encoded
value; example: dXNlcg== >
  AWS_SECRET_ACCESS_KEY:
<Base64-encoded value;
example:cGFzc3dvcmQ= >
metadata:
  name: <K8s-secret-name-for-S3>
type: Opaque
```

Configuring Spark Applications with a Secret

You can configure Spark Applications with a Secret manually using a YAML or adding the Secret during **Configure Spark Applications** step using HPE Ezmeral Runtime Enterprise new UI.

Using YAML: Configure the `secretRef` property in `envFrom` section for driver and executor in Spark Applications. Set the `name` option with the Kubernetes Secret.

```
driver:
   coreLimit: "1000m"
   cores: 1
   labels:
     version: 2.4.7
   envFrom:
      - secretRef:
          name:
<K8s-secret-name-for-S3>
executor:
   cores: 1
   coreLimit: "1000m"
   instances: 2
   memory: "512m"
   envFrom:
      - secretRef:
          name:
<K8s-secret-name-for-S3>
```

Using HPE Ezmeral Runtime Enterprise new UI: Enter a Secret when you select a **Source** as **S3** during **Configure Spark Applications** step of creating Spark applications. This will automatically add `secretRef` option in the YAML. See Creating Spark Applications on page 255.

Additional Configuration Options in SSL Environment

To access the Amazon S3 buckets using SSL, in addition to the previous configurations, you must also add the following configuration options in the `sparkConf` section of the Spark application configuration.

```
spark.driver.extraJavaOptions:
"-Dcom.amazonaws.sdk.disableCertChecking=true"
```

```
spark.executor.extraJavaOptions:
"-Dcom.amazonaws.sdk.disableCertChecking=true"
```

If you are using the HPE Ezmeral Runtime Enterprise new UI, add these configuration options by clicking **Edit YAML** in **Review** step or **Edit YAML** from Actions menu on **Spark Applications** screen. See Managing Spark Applications on page 260.

Managing Spark Applications Dependencies
This topic describes how to pass the dependencies to Spark applications in HPE Ezmeral Runtime Enterprise.

You can manage custom Spark dependencies in three different ways:

- Build the dependencies in main application jar. For example: Use the maven-assembly-plugin. See maven-assembly-plugin.
- Create a PersistentVolume for dependencies and mount it into driver and executor pods. You can use `local` schema to reference those dependencies. For example: See PySpark-with-dependencies.yaml.
- Build the custom images on top of the Spark images provided by HPE Ezmeral Runtime Enterprise. Copy or install the dependencies in your custom images. You can use `local` schema to reference those dependencies.

Supported Schemas for Main Application File in Spark Applications:

The following schemas are supported for main application file in Spark applications:

- `local`
- `dtap`
- `s3a`
- `maprfs`

Supported Schemas for Passing Dependencies

The following schemas are supported for passing dependencies to Spark applications:

- `local`
- `s3a`(AWS)
- `dtap`

Unsupported Schemas for Passing Dependencies

The following schemas are not supported for passing dependencies to Spark applications:

- `maprfs`
- `s3a` (custom). To learn more, see Spark on Kubernetes Issues (5.4.0) on Issues and Workarounds on page 15.

Deleting Spark Operator
This topic describes how to delete Spark Operator using Helm.

You can delete Spark Operator on HPE Ezmeral Runtime Enterprise using Helm chart.

Run the following command to delete the Spark Operator using Helm:

```
helm delete <spark-operator-name> -n <namespace>
```

For example:

```
helm delete spark-operator-compute -n compute
```

NOTE: Running the `helm delete` command does not delete the Spark Applications CRDs.

Connecting to Spark Operator from a KubeDirector Notebook Applications

This topic describes how to submit Spark applications using the EZMLLib library on KubeDirector notebook application.

The EZMLLib library includes the `from ezmlib.spark import submit, delete, logs` API which sets the configurations of your Spark applications.

You can submit, delete, and check logs of the Spark applications using the API.

Submit Spark Applications

You can submit the Spark applications using two different `submit` command:

Using Python files

Run the following command:

```
submit(app_path="<path-to-python-application-file>",

data_path="<path-to-data-source-for-the-application>",
        name="<application-name>"
       )
```

For example:

```
submit(app_path="local:///opt/mapr/spark/spark-3.1.2/examples/src/main/python/wordcount.py",
       data_path="dtap://TenantStorage/data/wordcount.txt",
       name="test1"
       )

Spark configuration: {'app_path': 'local:///opt/mapr/spark/spark-3.1.2/examples/src/main/python/wordcount.py', 'data_path': 'dtap://TenantStorage/data/wordcount.txt', 'yaml_path': None, 'name': 'test1', 'image_name': 'gcr.io/mapr-252711/spark-py-3.1.2:20211102109R', 'driver_cores': 1, 'driver_memory': '512m', 'driver_core_limit': '1000m', 'executor_cores': 1, 'executor_instances': 2, 'executor_memory': '512m', 'executor_core_limit': '1000m', 'spark_version': '3.1.2', 'python_version': '3', 'app_type': 'Python', 'api_version': 'sparkoperator.hpe.com/v1beta2', 'kind': 'SparkApplication', 'namespace': None}

Spark application 'test1' created
```

Using YAML files

Run the following command:

```
submit(yaml_path=f"<path-to-yaml-file>")
```

For example:

```
[5]: submit(yaml_path=f"{home}/examples/spark/data/spark-wc.yaml")

     Spark configuration: {'app_path': None, 'data_path': None, 'yaml_path': '/home/dev1/examples/spark/data/spark-wc.yaml', 'name': None, 'image_name': 'gcr.io/mapr-252711/spark-py-3.1.2:20210924201IC', 'driver_cores': 1, 'driver_memory': '512m', 'driver_core_limit': '1000m', 'executor_cores': 1, 'executor_instances': 2, 'executor_memory': '512m', 'executor_core_limit': '1000m', 'spark_version': '3.1.2', 'python_version': '3', 'app_type': 'Python', 'api_version': 'sparkoperator.hpe.com/v1beta2', 'kind': 'SparkApplication', 'namespace': None}

     Spark application 'pyspark-wordcount-secure' created
```

Check Logs of the Spark Applications

After you submit the Spark applications, you can check the both events logs and regular logs using `logs`.

Check events logs

Run `logs("<application-name>", events=True)`.

Check regular logs

Run `logs("<application-name>")`.

Delete Spark Applications

You can delete the Spark applications using `delete`.

Delete multiple applications	Run `delete("<application_1>", "<application_2>")`.
List available applications and delete applications	Run `delete()` and enter the space-delimited Spark applications name.

For example:

```
delete("pyspark-wordcount-secure", "test1")

Spark applications [pyspark-wordcount-secure, test1] deleted

#delete()

Available spark job names: ['pyspark-wordcount-secure', 'test1']
Please type the space-delimited spark job names (e.g., name1 name2 name3):  pyspark-wordcount-secure
Spark application [pyspark-wordcount-secure] deleted
```

To learn more about using EZMLLib, see Notebook ezmllib Functions on page 190.

Livy Overview

This topic provides the overview for Apache Livy on HPE Ezmeral Runtime Enterprise.

Livy is an HTTP server that allows you to launch Spark applications and submit code statements using the REST API. Livy can be launched as a tenant service in HPE Ezmeral Runtime Enterprise. See Apache Livy Documentation.

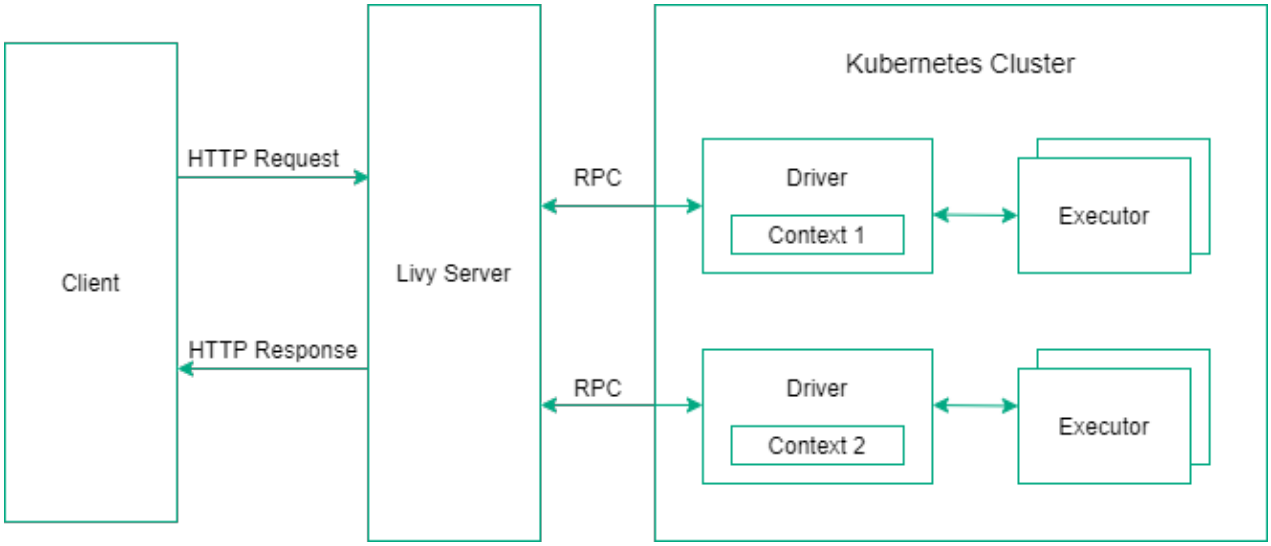


Figure 2: Using Apache Livy to Submit Spark Applications

Support for the Apache Livy Operator

HPE Ezmeral Runtime Enterprise integrates Apache Livy. You can only install one version of Livy in one tenant. You cannot simultaneously run Spark 2.x.x and Spark 3.x.x applications in the same tenant using Livy.

Learn more about supported Livy versions at Interoperability Matrix for Spark on page 246.

The new Livy features:

1. Authentication and Impersonation
2. Livy User Interface (UI)
3. Spark History Server

4. Livy Session Recovery
5. Support for Hive Metastore

NOTE: Livy does not support open-source Spark images on HPE Ezmeral Runtime Enterprise.

Apache Livy
The topics in this section provide information about Apache Livy features and limitations in HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise supports Apache Livy.

Learn more about supported Livy versions at Interoperability Matrix for Spark on page 246.

Apache Livy includes the following new features:

Authentication and Impersonation

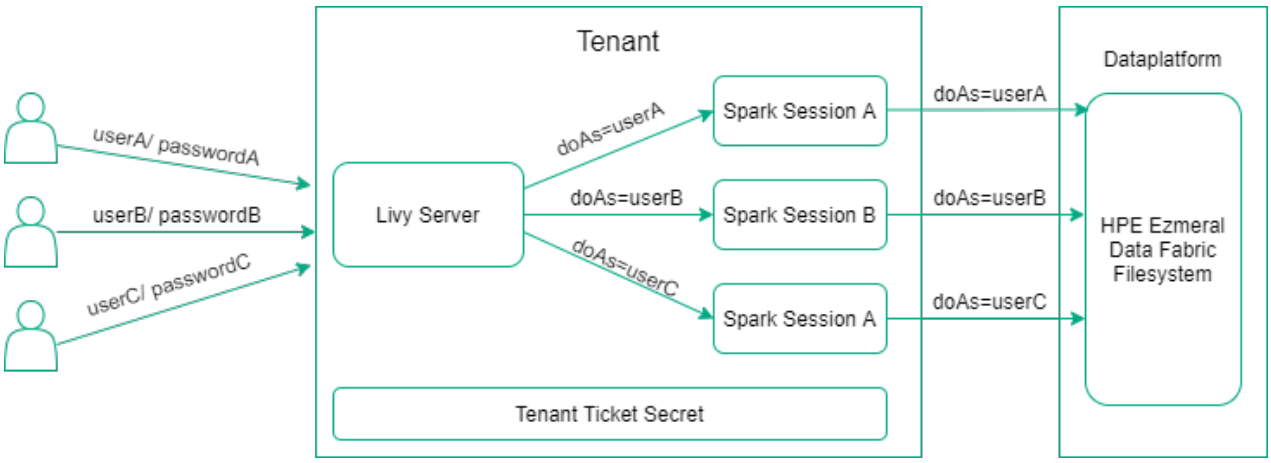


Figure 3: Apache Livy Impersonation on HPE Ezmeral Runtime Enterprise

Authentication and impersonation are enabled by default in Livy with the HPE Ezmeral Runtime Enterprise. Like any other container platform service with authentication, Livy authenticates users with LDAP credentials. Livy starts a Livy session with the tenant ticket secrets and specifies proxyUser. Livy server and all Livy sessions run with the same ticket secrets. Livy server provides the user information to the Spark session and Spark session will use that user information to access HPE Ezmeral Data Fabric Filesystem.

For this reason, curl commands need to include the `-u "username:password"` command string. For example, change this command:

```
curl -k -v \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{}' \
    https://hcp-lb1.qa.lab:10075/sessions
```

to this:

```
curl -k -v \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{}' \
    -u "username:password" \
    https://hcp-lb1.qa.lab:10075/sessions
```

Livy User Interface (UI)

The Livy user interface (UI) is secured and accessible via HTTPS from the HPE Ezmeral Runtime Enterprise web interface by navigating to **Kubernetes > Tenants > Applications > Service Endpoints**. For example:

```
https://your.co.lab:10046/ui
```

Spark History Server

The Spark History Server displays all active and completed Livy sessions. Spark History Server web UI is accessible via HTTPS from the HPE Ezmeral Runtime Enterprise web interface by navigating to **Kubernetes > Tenants > Applications > Service Endpoints**. For example:

```
https://your.co.lab:10038/
```

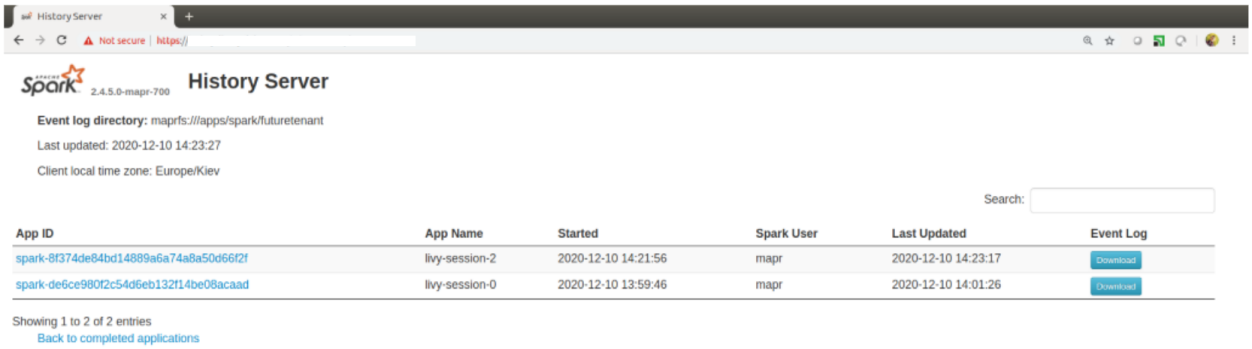


Figure 4: Spark History Server Display of Running and Completed Apps

Livy Session Recovery

Livy allows you to recover and continue working in previous sessions after deleting and restarting the Livy server pod. See Configuring Apache Livy for Session Recovery on page 293.

NOTE:

You cannot recover the Livy sessions if you delete the tenant. Livy sessions run in the tenant namespace. If you delete the tenant, it will delete the namespace and all the Livy sessions running in that namespace.

Support for Hive Metastore

Livy also supports integration with Hive Metastore. See Configuring Apache Livy for Hive Metastore on page 291.

Installing and Configuring Apache Livy

This section describes how to install and configure Apache Livy on HPE Ezmeral Runtime Enterprise.

Prerequisites

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member in HPE Ezmeral Runtime Enterprise.

About this task

In HPE Ezmeral Runtime Enterprise, install Livy for Apache Spark using the GUI or manually using the Helm chart.

You can only install one version of Livy in one tenant. You cannot simultaneously run Spark 2.x.x and Spark 3.x.x applications in the same tenant using Livy.

Learn more about supported Livy versions at Interoperability Matrix for Spark on page 246.

Installing Apache Livy Using the GUI

About this task

Install the Apache Livy for Apache Spark by using the HPE Ezmeral Runtime Enterprise GUI.

Procedure

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member on the HPE Ezmeral Runtime Enterprise GUI.
2. Click **Applications** in the main menu. You will see **Kubernetes Applications** tiles under **KubeDirector** tab.
3. Navigate to **Livy** tile and click **Launch**.
4. Configure **Cluster Detail** and **Settings** on **Create Application** screen.

Cluster Detail:	Enter the **Name** and **Description** of the application.

Settings:	Set the **CPU** and **Memory (GB)** resources.
	To connect Livy with Hive Metastore, enter ConfigMap created by Hive Metastore in **Hive Metastore Source**.
	To get the ConfigMap name, run the following command:
	`kubectl get cm -n <tenant-namespace> | grep hive`
	To enable **Session Recovery Settings**, check **Session Recovery** and select **Session Recovery Storage**.
	To set **Spark History Server Settings**, check **Spark History Server** and set **PVC Name** and **Event Log Directory**.
	To get the **PVC Name**, run the following command:
	`kubectl get pods <shs-name> -n <tenant-namespace> -o jsonpath='{.spec.volumes[*].persistentVolumeClaim.claimName}`
	To get the **Event Log Directory**, access Spark History Server web UI from HPE Ezmeral Runtime Enterprise web interface by navigating to **Applications> Service Endpoints**.
	In HPE Ezmeral Runtime Enterprise 5.4.0, to set **Air Gap Settings**, check **Air Gap** and set **Base Repository**, **Image**, **Image Tag**, **Image Pull Secret**. See Spark Images on page 249.

NOTE: To set the additional configuration options not available through HPE Ezmeral Runtime Enterprise GUI , edit the `values.yaml` file and install Livy using the helm charts as described in **Installing Apache Livy Using the Helm** section.

5. To view `yaml`, click **Edit/Launch yaml**.
6. Click **Submit**.

Results

The GUI installs the Apache Livy for Apache Spark in a tenant namespace.

Installing Apache Livy Using the Helm

Prerequisites

1. Install and configure Helm 3.
2. Install and configure kubectl.

About this task

Install the Apache Livy on Data Fabric tenants which are HPE Ezmeral Data Fabric on Kubernetes tenants or HPE Ezmeral Data Fabric on Bare Metal tenants or non Data Fabric tenants using the Helm Chart.

See Livy Helm Chart.

Procedure

- Helm install the Apache Livy on HPE Ezmeral Runtime Enterprise:
  - **Installing Apache Livy on Data Fabric tenants:**
    - To `helm install` the Apache Livy for Apache Spark 3.x.x, run the following command:

      ```
      helm install <livy-name> ./<path-to-livy-chart> -n <tenant-namespace>
      ```

    - To `helm install` the Apache Livy for Apache Spark 2.x.x, run the following command:

      ```
      helm install <livy-name> ./<path-to-livy-chart> -n
      <tenant-namespace> \
      --set image.imageName=<livy-image-name> \
      --set image.tag=<livy-image-tag> \
      --set livyVersion=<livy-version> \
      --set deImage=<spark-version>:<spark-image-name>
      ```

  - **Installing Apache Livy on non Data Fabric tenants:**
    - To `helm install` the Apache Livy for Apache Spark 3.x.x, run the following command:

      ```
      helm install <livy-name> ./<path-to-livy-chart> -n
      <tenant-namespace> --set tenantIsUnsecure=true
      ```

- To `helm install` the Apache Livy for Apache Spark 2.x.x, run the following command:

```
helm install <livy-name> ./<path-to-livy-chart> -n
<tenant-namespace> --set tenantIsUnsecure=true \
--set image.imageName=<livy-image-name> \
--set image.tag=<livy-image-tag> \
--set livyVersion=<livy-version> \
--set deImage=<spark-version>:<spark-image-name>
```

See Setting Custom KeyStore on page 281.

NOTE:

- Installing the Livy Helm chart in a non-tenant namespace can cause error due to missing configmaps and secrets.
- If you are using PVC, ensure that PVC is configured in the same tenant namespace as a Livy namespace.
- Configure Livy to work with Spark History Server and Hive Metastore on `values.yaml` file.

Running the `helm install` installs Apache Livy for Apache Spark in a tenant namespace.

Example

For HPE Ezmeral Runtime Enterprise 5.4.0:

- **Installing Apache Livy on Data Fabric tenants:**
  - To `helm install` the Apache Livy 0.7 for Apache Spark 3.1.2, run the following command:

```
helm install livy ./livy-chart -n sampletenant
```

  - To `helm install` the Apache Livy 0.5 for Apache Spark 2.4.7, run the following command:

```
helm install livy ./livy-chart -n sampletenant \
--set image.imageName=livy-0.5.0 \
--set image.tag=202112061039R \
--set livyVersion=0.5.0 \
--set deImage=spark-2.4.7:202112061039R
```

- **Installing Apache Livy on non Data Fabric tenants:**
  - To `helm install` the Apache Livy 0.7 for Apache Spark 3.1.2, run the following command:

```
helm install livy ./livy-chart -n sampletenant --set
tenantIsUnsecure=true
```

  - To `helm install` the Apache Livy 0.5 for Apache Spark 2.4.7, run the following command:

```
helm install livy ./livy-chart -n sampletenant --set
tenantIsUnsecure=true \
--set image.imageName=livy-0.5.0 \
--set image.tag=202112061039R \
--set livyVersion=0.5.0 \
--set deImage=spark-2.4.7:202112061039R
```

NOTE: You can modify the above examples based on your Livy, Spark, and HPE Ezmeral Runtime Enterprise versions for successful installation of Livy. See Interoperability Matrix for Spark on page 246.

Setting Custom KeyStore

This topic describes how to set custom KeyStore for Livy SSL encryption for non Data Fabric tenants.

A Java KeyStore is a repository of security certificates and their corresponding private keys used for SSL encryptions.

To set the custom KeyStore, perform the following steps:

1. Create a secret with KeyStore file in the tenant namespace.

   ```
   kubectl create secret generic "livy-ssl-secret" --from-file="./path/to/
   ssl_keystore"
   ```

   The secret must have a KeyStore file stored under a particular key.

2. To configure the Livy with SSL configurations, set `extraConfigs` section in `values.yaml` file.

   For example, if the secret name is `livy-ssl-secret`, KeyStore name in secret is `ssl-keystore`, and passwords are `examplepass`, update the `values.yaml` file as follows:

   ```
   livySsl:
     useCustomKeystore: true
     sslSecretName: "livy-ssl-secret"
     secretMountPath: /var/livy

   extraConfigs:
     livy.conf: |
       livy.keystore = /var/livy/ssl_keystore
       livy.keystore.password = examplepass
       livy.key-password = examplepass
   ```

Setting Custom TrustStore

This topic describes how to set custom trustStore for SSL encryption using Livy.

A Java trustStore is a repository to store the certificates from Certified Authorities (CA). CA verifies the certificate presented by the server in an SSL connection.

Add the following configuration options to set the custom trustStore during Livy session creation.

```
curl -k \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{"className": "com.mapr.example", "file": "maprfs:///user/mapr/
<example>.jar",
     "args": [
            "<args>"
        ],
     "conf":{
              .....
              "spark.driver.extraJavaOptions":
"-Djavax.net.ssl.trustStore=<path-to-java-cacerts-file>",
              "spark.executor.extraJavaOptions":
"-Djavax.net.ssl.trustStore=<path-to-java-cacerts-file>",
              .....        }    }' \
    -u "user:password" \
    https://<livy_url>/batches
```

NOTE: The default Java trustStore is `/etc/pki/java/cacerts`. For example:

```
curl -k \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{"className": "com.mapr.example", "file": "maprfs:///user/mapr/
<example>.jar",
     "args": [
            "<args>"
        ],
     "conf":{
              .....
              "spark.driver.extraJavaOptions":
"-Djavax.net.ssl.trustStore=/etc/pki/java/cacerts",
              "spark.executor.extraJavaOptions":
"-Djavax.net.ssl.trustStore=/etc/pki/java/cacerts",
              .....        }    }' \
    -u "user:password" \
    https://<livy_url>/batches
```

However, you can modify the path for the trustStore.

To install the custom certificate, see Secret Mangement.

Submitting Spark Application Using Livy

This section guides you through starting Apache Livy session and exceuting a code in a Livy session.This page shows some examples of Livy supporting multiple APIs and Livy batches.

To find out which Livy images to use with installed python packages for PySpark, installed R packages for SparkR, and for basic spark sessions with Scala. See Spark Images on page 249.

Start Livy Session

If you are an LDAP/AD user, you can navigate to **Kubernetes > Tenants > Applications > Service Endpoints** on HPE Ezmeral Runtime Enterprise to find livy-http URL or Access Point and corresponding port.

Run the following commands to submit REST API call to start a Livy session:

```
curl -k -v \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{}' \
    -u "username:password" \
    https://<livy-url>
```

Code Execution in a Livy Session

Perform the following steps to execute the code in Livy session:

1. Run the following command to input some text file into the HPE Ezmeral Data Fabric file system:

```
kubectl -n sampletenant exec -it tenantcli-0 -- hadoop fs -put /etc/
passwd
```

2. Execute the following command to run a Spark job in the Livy session:

```
curl -k \
    -X POST \
    -H "Content-Type: application/json" \
```

```
    -d '{"kind": "spark", "code": "var a = spark.read.csv(\"/user/mapr/
passwd\"); a.show();"}' \
    -u "username:password" \
    https://<livy-url>/sessions/<session-number>/statements
```

Delete Livy Session

Run the following command to delete the Livy session:

```
curl -k -X DELETE "https://<livy-URL>/sessions/<session-number>"; echo
```

When you delete a Livy session, Livy server stops the execution of the Spark job created for the current session and both driver and executor pods remain at a Completed state until it is removed by the Kubernetes API.

Livy Session Supports Multiple APIs

The following examples shows that the Livy server supports multiple (Scala, Python, and R) APIs on HPE Ezmeral Runtime Enterprise:

1. **Livy Session (PySpark)**

   Run the following commands to submit REST API call to start a Livy session for PySpark:

   ```
   curl -k \
       -X POST \
       -H "Content-Type: application/json" \
       -d '{"conf":{"spark.kubernetes.container.image":"gcr.io/mapr-252711/
   <livy-image-for-PySpark>"},"kind":"pyspark"}' \
       -u "username:password" \
       https://<livy-url>/sessions
   ```

   Execute the following command to run a Spark job using PySpark:

   ```
   curl -k \
       -X POST \
       -H "Content-Type: application/json" \
       -d '{"code": "sc.parallelize([0, 2, 3, 4, 6],
   5).glom().collect();"}' \
       -u "username:password" \
       https://<livy-url>/sessions/<session-number>/statements
   ```

2. **Livy Session (R)**

   Run the following commands to submit REST API call to start a Livy session for SparkR:

```
curl -k \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{"conf":{"spark.kubernetes.container.image":"gcr.io/mapr-252711/
<livy-image-for-SparkR"},"kind":"sparkr"}' \
    -u "username:password" \
    https://<livy-url>/sessions
```

   Execute the following command to run a Spark job using SparkR:

```
curl -k \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"code": "summary(data.frame( emp_id = c(1:5),
emp_name = c(\"Rick\",\"Dan\",\"Michelle\",\"Ryan\",\"Gary\"),
salary = c(623.3,515.2,611.0,729.0,843.25),
start_date = as.Date(c(\"2012-01-01\",\"2013-09-
23\",\"2014-11-15\",\"2014-05-11\",\"2015-03-27\")), stringsAsFactors =
TRUE));"}' \
    -u "username:password" \
    https://<livy-url>/sessions/<session-number>/statements
```

3. **Livy Session (Shared)**

   Livy server supports multiple APIs in the same Livy session. After creating a Livy session, you can configure the `kind` option for each statement to use Scala, Python, and R in a single Livy session.

   The following example shows the use of Scala and Python API in the single Livy session:

```
curl -k \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{"conf":{"spark.kubernetes.container.image":"gcr.io/mapr-252711/
<livy-image-for-PySpark"}}' \
    -u "username:password" \
    https://<livy-url>/sessions

curl -k \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"kind": "spark", "code": "var a = spark.read.csv(\"/user/mapr/
passwd\"); a.show();"}' \
    -u "username:password" \
    https://<livy-url>/sessions/<session-number>/statements

curl -k \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"kind": "pyspark", "code": "sc.parallelize([0, 2, 3, 4, 6],
5).glom().collect();"}' \
    -u "username:password" \
    https://<livy-url>/sessions/<session-number>/statements
```

Livy Supports Batch Application

You can submit batch applications in Livy through REST APIs.

Some ready-to-use sample Spark applications built into to the container image. These applications are located at `/opt/mapr/spark/spark-[version]/jars/spark-examples_[full-version].jar` and should be referenced using the `local` schema. You may also build your own applications and make them available in `/opt/mapr/`.

If the Spark application is located elsewhere, then modify the `file` field to point to that storage and interface.

For example, the Livy server supports the `https://`, `maprfs://`, `dtap://`, and `S3://` interfaces.

Run the following command to submit Spark applications using Livy batches:

```
curl -k \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{"className": "org.apache.spark.examples.SparkPi",
"file": "local:///opt/mapr/spark/spark-<version>/examples/jars/
<spark-examples.jar>"}' \
    -u "username:password" \
    https:/<livy-url>/batches

kubectl logs -f org-apache-spark-examples-sparkpi-1605535907482-driver -n
livytenant curl
https://<livy-url>/batches/0/log | jq
```

NOTE:

Do not use `jar` option to set the dependencies for Livy batch applications. Set the DataTap JAR using the `spark.driver.extraClassPath` and `spark.executor.extraClassPath` options in `conf` section of Spark application.

For example:

```
curl \
    -k \
    -s \
    -u <user1>:<password> \
    -H "Content-Type: application/json" \
    -d '{
        "file": "dtap://TenantStorage/wordcount.py"
        , "args": [
            "dtap://TenantStorage/passwd"
        ]
        , "conf":{
            "spark.ssl.enabled":"false"
            , "spark.hadoop.fs.dtap.impl":
"com.bluedata.hadoop.bdfs.Bdfs"
            , "spark.hadoop.fs.AbstractFileSystem.dtap.impl":
"com.bluedata.hadoop.bdfs.BdAbstractFS"
            , "spark.hadoop.fs.dtap.impl.disable.cache": "false"
            , "spark.kubernetes.driver.label.hpecp.hpe.com/dtap":
"hadoop2-job"
            , "spark.kubernetes.executor.label.hpecp.hpe.com/dtap":
"hadoop2-job"
            , "spark.driver.extraClassPath": "local:///opt/bdfs/
bluedata-dtap.jar"
            , "spark.executor.extraClassPath": "local:///opt/bdfs/
bluedata-dtap.jar"
        }
    }' \
    "https://$NODE_IP:$NODE_PORT/batches" | jq
```

Accessing Data on Amazon S3 Using Livy

This topic describes how to configure Amazon S3 to access data using Livy on HPE Ezmeral Runtime Enterprise.

You must configure Amazon S3 credentials to access the S3 storage using Livy.

You can configure your S3 access credentials in the following ways:

1. Configuring access to Amazon S3 for all the Livy sessions created by Livy instance.

   To configure S3 credentials for tenants, add the following configuration options in `spark-defaults.conf` section in `extraConfigs` section of `values.yaml` file of Helm chart in a tenant namespace.

   ```
   extraConfigs:
     spark-defaults.conf: |
       spark.hadoop.fs.s3a.access.key <access-key>
       spark.hadoop.fs.s3a.secret.key <secret-key>
       spark.hadoop.fs.s3a.path.style.access true
   ```

   The sensitive data provided in `extraConfigs` section are added to `spark-defaults.conf` file using the Kubernetes secret. The secret has Key-Value format where the key is `spark-defaults.conf` file and the value is sensitive data.

   You must also add the following properties on `spark-defaults.conf` section of `values.yaml` file in a tenant namespace.

   ```
   extraConfigs:
     spark-defaults.conf: |
       # Environment variables here would be replaced by its values
       # ...

   spark.driver.extraJavaOptions -Dcom.amazonaws.sdk.disableCertChecking=tru
   e

   spark.executor.extraJavaOptions -Dcom.amazonaws.sdk.disableCertChecking=t
   rue
   ```

2. Configuring access to Amazon S3 for a specific Livy session.

   Add the following configuration options to configure Livy session during session creation.

   ```
   spark.hadoop.fs.s3a.access.key <YOURACCESSKEY>
   spark.hadoop.fs.s3a.secret.key <YOURSECRETKEY>
   ```

   For example: Configuring Livy session to access S3 storage using the REST API.

   ```
   curl \
       -k \
       -s \
       -X POST \
       -H "Content-Type: application/json" \
       -d '{
           "kind": "spark",
           "conf": {
               "spark.hadoop.fs.s3a.access.key": "<YOURACCESSKEY>",
               "spark.hadoop.fs.s3a.secret.key": "<YOURSECRETKEY>"
           }
       }' \
       -u username:password \
       https://hcp-lb1.qa.lab:10075/sessions | jq
   ```

3. Configuring access to Amazon S3 during runtime.

   Set the `spark.sparkContext.hadoopConfiguration` options during runtime and submit the spark jobs.

   For example:

```
val hadoopConf = spark.sparkContext.hadoopConfiguration
hadoopConf.set("fs.s3a.access.key", "<YOURACCESSKEY>")
hadoopConf.set("fs.s3a.secret.key", "<YOURSECRETKEY>")
hadoopConf.set("fs.s3a.path.style.access", "true")

val path = "s3a://bucket/path/to/dest/"

val data = Seq(
    ("banana", "yellow"),
    ("orange", "orange"),
    ("tomato", "red"),
    ("potato", "white"),
    ("plum", "purple"),
)

val df = data.toDF

println(s"Writing DataFrame to $path")
df.write.parquet(path)
println("Write complete")

println(s"Reading DataFrame from $path")
spark.read.parquet(path).show()
println("Read complete")
```

   The output of the submitted code block is as follows:

```
hadoopConf: org.apache.hadoop.conf.Configuration = Configuration:
core-default.xml, org.apache.hadoop.conf.CoreDefaultProperties,
core-site.xml, mapred-default.xml,
org.apache.hadoop.mapreduce.conf.MapReduceDefaultProperties,
mapred-site.xml, yarn-default.xml,
org.apache.hadoop.yarn.conf.YarnDefaultProperties, yarn-site.xml,
hdfs-default.xml, hdfs-site.xml, __spark_hadoop_conf__.xml, file:/opt/
mapr/spark/spark-2.4.7/conf/hive-site.xml

path: String = s3a://bucket/path/to/dest/

data: Seq[(String, String)] = List((banana,yellow), (orange,orange),
(tomato,red), (potato,white), (plum,purple))

df: org.apache.spark.sql.DataFrame = [_1: string, _2: string]

Writing DataFrame to s3a://bucket/path/to/dest/

Write complete

Reading DataFrame from s3a://bucket/path/to/dest/

+------+------+
|    _1|    _2|
+------+------+
|tomato|   red|
|potato| white|
|  plum|purple|
|banana|yellow|
```

```
|orange|orange|
+------+------+

Read complete
```

DataTap Integration on Livy

This topic describes how to integrate DataTap on Livy with REST API and Jupyter Notebook in HPE Ezmeral Runtime Enterprise.

Using DataTap with REST API

1. Start Livy session using a `curl` command.

```
curl -k \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{
            "conf": {
                "spark.hadoop.fs.dtap.impl":
"com.bluedata.hadoop.bdfs.Bdfs",
                "spark.hadoop.fs.AbstractFileSystem.dtap.impl":
"com.bluedata.hadoop.bdfs.BdAbstractFS",
                "spark.hadoop.fs.dtap.impl.disable.cache": "false",
                "spark.kubernetes.driver.label.hpecp.hpe.com/dtap":
"hadoop2-job",
                "spark.kubernetes.executor.label.hpecp.hpe.com/dtap":
"hadoop2-job"
            },
            "jars": [
                "local:///opt/bdfs/bluedata-dtap.jar"
            ]
        }' \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions
```

NOTE: Do not use `jar` option to set the dependencies for Livy batch applications.Set the DataTap JAR using the `spark.driver.extraClassPath` and `spark.executor.extraClassPath` options in `conf` section of Livy application. For example: See Submitting Spark Application Using Livy on page 282.

2. Execute a code in Livy session. For example:

NOTE: You must have `.csv` file in your DataTap storage before executing a curl command.

```
curl -k \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{
            "kind": "spark",
            "code": "var a = spark.read.csv(\"dtap://TenantStorage/
somefile.csv\"); a.show();"
        }' \
    -u "username:password"
    https://xx-xxx-xxx.xx.lab:10075/sessions/0/statements
```

Using DataTap with Jupyter Notebook

1. Start Livy session in Kubeflow Jupyter Notebook.

a. Load the sparkmagic to configure the Livy endpoints in Jupyter Notebook.

```
%load_ext sparkmagic.magics
```

b. Run the following magic to add the Livy endpoint and to create a Livy session.

```
%manage_spark
```

Add the following configuration options to properties when creating a Livy session.

```
{
    "conf": {
        "spark.hadoop.fs.dtap.impl": "com.bluedata.hadoop.bdfs.Bdfs",
        "spark.hadoop.fs.AbstractFileSystem.dtap.impl":
"com.bluedata.hadoop.bdfs.BdAbstractFS",
        "spark.hadoop.fs.dtap.impl.disable.cache": "false",
        "spark.kubernetes.driver.label.hpecp.hpe.com/dtap":
"hadoop2-job",
        "spark.kubernetes.executor.label.hpecp.hpe.com/dtap":
"hadoop2-job"
    },
    "jars": [
        "local:///opt/bdfs/bluedata-dtap.jar"
    ]
}
```

2. Execute a code in Livy session. For example:

NOTE: You must have `.csv` file in your DataTap storage before executing a curl command.

```
%%spark

var a = spark.read.csv("dtap://TenantStorage/somefile.csv");
a.show();
```

See About DataTaps for DataTaps descriptions and configuration.

Viewing Spark Job Results on Livy

This topic describes how to view results for submitted Spark jobs on Livy in HPE Ezmeral Runtime Enterprise.

You can view the results for submitted Spark jobs in the following ways:

1. Using Notebook's UI on the interactive notebooks like Jupyter Notebook.
2. Accessing Livy UI through Livy HTTP endpoints. For example:`https://xx-xxx-xxx.xx.lab:10075/ui or https://xx-xxx-xxx.xx.lab:10075`
3. Using a REST API.

   For example, perform the following steps to view the results for submitted jobs using a REST API:

a. Create a Livy session.

```
curl -k -s \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{"kind": "spark"}' \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions | jq
```

b. Execute code on a newly created Livy session.

```
curl -k -s \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{"code": "sc.parallelize(List(1,2,3)).reduce(_*_)"}' \
    -u "username:password" \
    https://cxx-xxx-xxx.xx.lab:10075:10075/sessions/0/statements | jq
```

When you execute a block of statement, a Livy server assigns an id to that block of statement.

```
{
  "id": 0,
  "code": "sc.parallelize(List(1,2,3)).reduce(_*_)",
  "state": "waiting",
  "output": null,
  "progress": 0
}
```

c. You can either see the output for a particular block of statement or for the total number of statements using the following commands:

1. Run the following commands to see the state and output for the specific submitted block of statement for the specific session.

```
curl -k -s \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions/<session id>/
statements/<id> | jq
```

Example of output result in a session zero for a block of statement with an id zero.

```
{
  "id": 0,
  "code": "sc.parallelize(List(1,2,3)).reduce(_*_)",
  "state": "available",
  "output": {
    "status": "ok",
    "execution_count": 0,
    "data": {
      "text/plain": "res0: Int = 6\n"
    }
  },
  "progress": 1
}
```

2. Run the following commands to see the state and output of all the submitted blocks of statement for a specific session.

```
curl -k -s \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions/0/statements | jq
```

Example of output result in a session zero where the total number of submitted statements are two.

```
{
  "total_statements": 2,
  "statements": [
    {
      "id": 0,
      "code": "sc.parallelize(List(1,2,3)).reduce(_*_)",
      "state": "available",
      "output": {
        "status": "ok",
        "execution_count": 0,
        "data": {
          "text/plain": "res0: Int = 6\n"
        }
      },
      "progress": 1
    },
    {
      "id": 1,
      "code": "sc.parallelize(List(10,20,30)).reduce(_*_)",
      "state": "available",
      "output": {
        "status": "ok",
        "execution_count": 1,
        "data": {
          "text/plain": "res2: Int = 6000\n"
        }
      },
      "progress": 1
    }
  ]
}
```

Configuring Apache Livy for Hive Metastore

This page describes some common use cases for configuring Livy with Hive Metastore.

Starting from HPE Ezmeral Runtime Enterprise 5.4.0, you can configure Apache Livy for Hive Metastore in two different ways.

1. Using the HPE Ezmeral Runtime Enterprise GUI during the Livy installation, see Installing and Configuring Apache Livy on page 277.

2. Setting the ConfigMap name in the `hiveSiteSource` configuration option in `values.yaml` file in the Helm chart.

NOTE: There is a known issue related to integration of Livy with Hive Metastore. See Spark on Kubernetes Issues (5.4.0) on page 35.

Examples of SQL (Hive) Support in Livy

- Create Livy session:

```
curl -k -v \
    -X POST \
    -H "Content-Type:application/json" \
    -d '{}' \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions
```

- Create table:

```
curl -k \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"kind": "sql", "code": "CREATE TABLE test1 (id int)"}' \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions/0/statements
```

- Insert some data:

```
curl -k \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"kind": "sql", "code": "INSERT INTO test1 VALUES (1),(2),(3)"}' \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions/0/statements
```

- Select from table:

```
curl -k \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"kind": "spark", "code": "var a = sql(\"SELECT * FROM test1\");
a.show()"}' \
    -u "username:password" \
    https://xx-xxx-xxx.xx.lab:10075/sessions/0/statements
```

Managing Livy Dependencies

This topic describes how to pass the dependencies to Livy applications in HPE Ezmeral Runtime Enterprise using Notebook.

About this task

You can configure dependencies for Livy Applications using Notebook on HPE Ezmeral Runtime Enterprise. The following schemas are supported for passing dependencies to Livy applications:

- `local`
- `dtap`

Example

To configure the dependencies on Livy Applications using Notebook, perform the following steps:

1. Log in to HPE Ezmeral Runtime Enterprise as Kubernetes Tenant Administrator or Kubernetes Tenant Member.

2. Click **DataTaps** >**TenantStorage**.
3. Create `jars` directory.
4. Upload dependencies for Spark applications on `jars` directory.
5. Click **Notebooks** > **Notebook Endpoints** > **Access Points**.
6. Add the following configuration options for Livy applications:

```
%%configure -f
{
    "conf": {
        "spark.hadoop.fs.dtap.impl": "com.bluedata.hadoop.bdfs.Bdfs"
        , "spark.hadoop.fs.AbstractFileSystem.dtap.impl":
"com.bluedata.hadoop.bdfs.BdAbstractFS"
        , "spark.hadoop.fs.dtap.impl.disable.cache": "false"
        , "spark.kubernetes.driver.label.hpecp.hpe.com/dtap":
"hadoop2-job"
        , "spark.kubernetes.executor.label.hpecp.hpe.com/dtap":
"hadoop2-job"
        , "spark.driver.extraClassPath": "/opt/bdfs/bluedata-dtap.jar"
        , "spark.executor.extraClassPath": "/opt/bdfs/bluedata-dtap.jar"
    }
    , "jars": [
        "local:///opt/bdfs/bluedata-dtap.jar"
    ]
    , "pyFiles": [
        "dtap://TenantStorage/python/kazoo-2.8.0-py2.py3-none-any.whl"
    ]
}
```

7. Restart the kernel.

Configuring Apache Livy for Session Recovery
Livy supports session recovery on HPE Ezmeral Runtime Enterprise. Even if the Livy server fails, Livy allows you to recover and continue working in previous sessions after deleting and restarting the Livy server pod.

Configuring Apache Livy for Session Recovery

Starting from HPE Ezmeral Runtime Enterprise 5.4.0, you can configure Apache Livy for session recovery in two different ways.

1. Using the HPE Ezmeral Runtime Enterprise GUI during the Livy installation, see Installing and Configuring Apache Livy on page 277.
2. Setting the `kind` option in `sessionRecovery` property in `values.yaml` file.

   You can use one of the following options to configure Livy for session recovery:

a. Use the `disabled` option to disable the session recovery in Livy.

NOTE: The default value for session recovery in Livy is `disabled`.

For example:

```
sessionRecovery:
  ##supported sessionRecovery Kind: disabled, pvc
  kind: disabled
  ##use this option to configure volumeClaimTemplate for kind pvc
  pvcTemplate:
    metadata:
      name: livy-sessionstore
    spec:
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 1Gi
```

b. Use the `pvc` option to enable the filesystem session recovery in Livy.

For example:

```
sessionRecovery:
  ##supported sessionRecovery Kind: disabled, pvc
  kind: pvc
  ##use this option to configure volumeClaimTemplate for kind pvc
  pvcTemplate:
    metadata:
      name: livy-sessionstore
    spec:
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 1Gi
```

Connecting to Livy from a KubeDirector Notebook Application with Spark Magic

This topic describes how to initiate a Spark session from a Livy endpoint and how to use the `%setLivy` magic to connect to a different Livy session.

Using the `%%spark` Magic to Start Spark Sessions

If you are using a PySpark kernel in a KubeDirector Notebook application, you can use `%%spark` magic to set the Livy endpoints. Executing the magic generates a request for the user password. If the Livy session needs authentication, enter the password.

For example:

```
[1]: %%spark

Enter Livy endpoint (e.g., http://<internal-livy-session-url>:<port>) :
Enter your password:
WARNING: Restart the kernel if you entered something wrong or if the kernel fails.
Starting Spark application
```

Spark Session ID	Kind	State	Spark UI	Driver log	User	Current session?
5	None	pyspark	idle			None ✓

```
SparkSession available as 'spark'.
```

Figure 5: Using `%%spark` Magic to Start a Spark Session

Using the `%setLivy` Magic to Connect to Livy Sessions

For other kernels in a KubeDirector Notebook application, you can use the `%setLivy` magic to connect to a different Livy session.

```
%setLivy --url <livy-endpoint>
```

For example:

```
%setLivy --url http://<endpoint-details>:<port-number>
```

To connect to a remote Livy session in a different cluster or in a different system, use `%setLivy` magic. Provide the `--url` argument followed by the Livy endpoint to which you want to connect. Executing the magic generates a request for the user password. If the Livy session needs authentication, enter the password.

You can then import a Spark session and proceed to work in Spark.

Using the `%%configure` Magic to Configure Livy Sessions

You can use `%%configure` magic command on HPE Ezmeral ML Ops to override the default Livy configuration and custom configure each Livy session.

```
%%configure -f
{"driverCores": 1,
 "executorCores": 1,
 "driverMemory": "1000M",
 "executorMemory": "1000M",
 "conf": {"spark.kubernetes.container.image": "gcr.io/mapr-252711/
<livy-image-for-PySpark>",
  "spark.kubernetes.driver.limit.cores": "1",
  "spark.kubernetes.executor.limit.cores":"1"}}
```

Submitting Spark Applications Using `spark-submit`

This topic describes how to install `spark-client` Helm chart and submit Spark applications using `spark-submit` utility in HPE Ezmeral Runtime Enterprise.

In HPE Ezmeral Runtime Enterprise, you can install `spark-client` manually using the Helm chart for Apache Spark.

See Spark Client Helm Chart.

Learn more about supported Spark versions at Interoperability Matrix for Spark on page 246.

Install `spark-client` Helm Chart

To install the Helm chart for `spark-client` for Apache Spark 3.x.x in an existing tenant namespace, run:

```
helm install <spark-client-name> ./<path-to-spark-client-chart> -n
<tenant-namespace>
```

To install the Helm chart for `spark-client` for Apache Spark 3.x.x in a new tenant namespace, run:

```
helm install <spark-client-name> ./<path-to-spark-client-chart> -n
<tenant-namespace> --create-namespace
```

To install the Helm chart for `spark-client` for Apache Spark 2.x.x in an existing namespace, run

```
helm install <spark-client-name> ./<path-to-spark-client-chart> -n
<tenant-namespace> /
--set image.imageName=<spark-client-image-name> /
--set image.tag=<spark-client-imagetag>
```

See Spark Images on page 249.

Submit Spark Applications Using `spark-submit` on Cluster Mode

You can configure and submit the Spark applications using the `spark-submit` on cluster deploy mode in HPE Ezmeral Runtime Enterprise.

When your `spark-client` pod is up and running, perform the following steps:

1. Determine the gateway FQDN and port to SSH into the client pod. Run:

   ```
   kubectl describe svc spark-client-spark-client-chart -n t1 | grep gateway\
   ```

2. SSH into that FQDN and port, and authenticate as a tenant member AD user.
3. When you are in the spark-client pod, run maprlogin password and enter the AD password again.
4. Locate the Spark binary of your choice under `/opt/mapr/spark/spark-<version>/bin`. See Interoperability Matrix for Spark on page 246.
5. Submit the Spark applications. See Launching Applications with spark-submit.

NOTE: HPE Ezmeral Runtime Enterprise does not support client deploy mode.

Delete `spark-client` Helm Chart

Run the following command to delete the `spark-client` Helm chart.

```
helm delete <spark-client-name> -n <tenant-namespace>
```

For example:

```
helm delete spark-client -n sampletenant
```

Delta Lake with Apache Spark

This section describes the Delta Lake that provides ACID transactions for Apache Spark 3.x.x on HPE Ezmeral Runtime Enterprise.

Delta Lake is an open-source storage layer that supports ACID transactions to provide reliability, consistency, and scalability to Apache Spark applications. Delta Lake runs on the top of the existing storage and is compatible with Apache Spark APIs. For more details see Delta Lake documentation.

ACID Transactions with Delta Lake

ACID stands for Atomicity, Consistency, Isolation and Durability. ACID transactions for Spark applications are supported out of box with Delta Lake on HPE Ezmeral Runtime Enterprise.

You can use any Apache Spark APIs to read and write data with Delta Lake. Delta Lake stores the data in Parquet format as versioned Parquet files. Delta Lake has a well-defined open protocol called Delta Transaction Protocol that provides ACID transactions to Apache Spark applications.

Delta Lake stores the commits of every successful transaction (Spark job) as a DeltaLog or a Delta Lake transaction log.

For example: You can view these commit logs on MinIO Browser by navigating to `/<table_name>/_delta_Log/`.

Commits in the transaction log:

```
/<table_name>/_delta_log/00000000000000000000.json
/<table_name>/_delta_log/00000000000000000001.json
/<table_name>/_delta_log/00000000000000000003.json
```

Delta lake uses optimistic concurrency control to provide ACID transactions between writes operation. See Concurrency Control.

Examples of Enabling Delta Lake for Apache Spark 3.x.x

Perform the following steps to enable Delta Lake support for Spark Applications for S3 storage:

1. Add the following Delta Lake jar to `deps` option of `spec` property of the Spark job.

   ```
   deps:
       jars:
         - local:///tmp/delta-core_2.12-1.0.0.jar
   ```

   NOTE: In HPE Ezmeral Runtime Enterprise, Delta Lake jar is included in the Apache Spark image.

2. Add the following security configurations to `sparkconf` option of `spec` property of the Spark job.

   ```
   "spark.hadoop.fs.s3a.endpoint": #minio endpoint
   "spark.hadoop.fs.s3a.access.key": #minio access key
   "spark.hadoop.fs.s3a.secret.key": #minio secret key
   "spark.hadoop.fs.s3a.impl": org.apache.hadoop.fs.s3a.S3AFileSystem
   ```

   NOTE: If you are using a Python API, you need to add the following additional configurations to `sparkconf` option of `spec` property of the Spark job.

   ```
   "spark.sql.extensions": "io.delta.sql.DeltaSparkSessionExtension"
   "spark.sql.catalog.spark_catalog":
   "org.apache.spark.sql.delta.catalog.DeltaCatalog"
   ```

For examples of Delta lake enabled Spark jobs in HPE Ezmeral Runtime Enterprise 5.6, see Delta Lake Spark Examples.

To locate Delta Lake examples for other versions of HPE Ezmeral Runtime Enterprise, navigate to the release branch of your choice at Spark on K8s GitHub location and find the examples in `examples` folder.

Learn more about supported Spark versions at Interoperability Matrix for Spark on page 246.

Spark History Server

This topic provides an overview of Spark History Server.

Spark History Server provides a web UI to monitor and view the status of submitted Spark Applications. It shows the status of Running, Completed, and Failed (completed but failed) Spark Applications.

Spark History Server pulls the details of Spark Applications from event logs directory when you enable the event logging to submit the Spark Applications. The event logs are written to the event log directory on persistent storage.

You can access the Spark History Server web UI through node IP address and node port number as exposed by NodePort Service at

```
http://<node-IP-address>:<node-port-number>
```

The default node port number is 18080.

For more details about Spark History Server, see Apache Monitoring and Instrumentation.

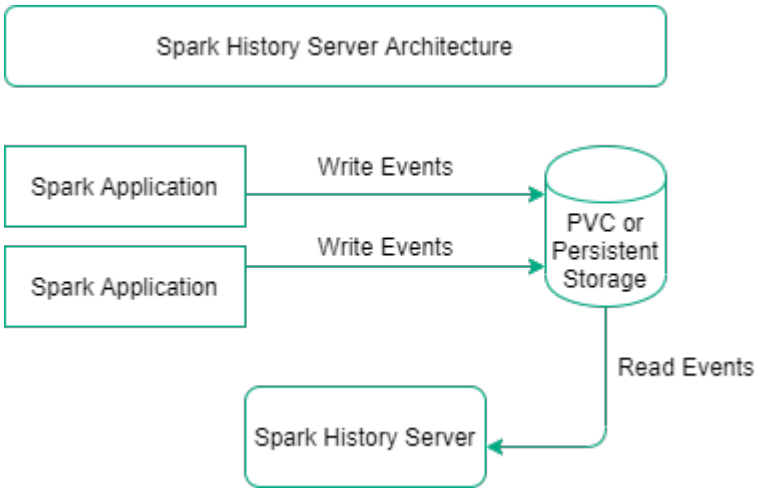


Figure 6: Spark History Server Architecture

Installing and Configuring Spark History Server

This section describes how to install and configure Spark History Server on HPE Ezmeral Runtime Enterprise.

Prerequisites

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member in HPE Ezmeral Runtime Enterprise.
2. Install Spark Operator and enable the webhook. See Spark Operator on page 264.
3. If you are using the PVC:
   a. Configure the PVC of type `ReadWriteMany` in the tenant namespace.
   b. To ensure you can write data to the file system storage (for example, `maprfs`) of PV, set the permissions to `777` on the target folder. Platform Administrator can set the permissions on the target folder of the file system storage.

About this task

In HPE Ezmeral Runtime Enterprise, you can install Spark History Server using GUI or manually using the Helm chart for Apache Spark.

Apache Spark supports the Data Fabric filesystem (`maprfs`) and PersistentVolumeClaim (PVC) as the persistent storage on HPE Ezmeral Runtime Enterprise.

Learn more about supported Spark versions at Interoperability Matrix for Spark on page 246.

Installing Spark History Server Using the GUI

About this task

Install the Spark History Server by using the HPE Ezmeral Runtime Enterprise GUI.

Procedure

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member on the HPE Ezmeral Runtime Enterprise GUI.
2. Click **Applications** in the main menu. You will see **Kubernetes Applications** tiles under **KubeDirector** tab.
3. Navigate to **Spark History Server** tile and click **Launch**.
4. Configure **Cluster Detail** and **Settings** on **Create Application** screen.

Cluster Detail:	Enter the **Name** and **Description** of the application.
Settings:	Set the **CPU** and **Memory (GB)** resources. To set **Event Log Storage Settings**, check **Event Log Storage** and select **Storage Type**. To set the additional configuration options not available through HPE Ezmeral Runtime Enterprise GUI , edit the `values.yaml` file and install Spark History Server using the helm charts as described in **Installing Spark History Server Using the Helm** section.

5. To view `yaml`, click **Edit/Launch yaml**.
6. Click **Submit**.

Results

The GUI installs the Spark History Server in a tenant namespace.

Installing Spark History Server Using the Helm

Prerequisites

1. Install and configure Helm 3.
2. Install and configure kubectl.

About this task

Install the Spark History Server on Data Fabric tenants which are HPE Ezmeral Data Fabric on Kubernetes tenants or HPE Ezmeral Data Fabric on Bare Metal tenants or non Data Fabric tenants by using the Helm chart.

See Spark History Server Helm Chart.

Procedure

- Helm install the Spark History Server on HPE Ezmeral Runtime Enterprise:

- **Installing Spark History Server on Data Fabric tenants:**

  To `helm install` the Spark History Server, run the following command:

  ```
  helm dependency update ./<path-to-spark-hs-chart>
  ```

  ```
  helm install <spark-hs-name> ./<path-to-spark-hs-chart>
  ```

  The `helm install` creates the helm chart in the default namespace. To create the chart in a different existing namespace, use the flag `-n <tenant-namespace>`.

  To set the tenant namespace during installation, use the flag `--set tenantNameSpace=<tenant-namespace>`.

  To configure PVC, set the following flags:

  ```
  --set pvc.enablePVC=true --set pvc.ExistingClaimName=<pvc-name> --set
  pvc.eventsDir=<path-to-directory>
  ```

  Alternatively, you can configure the PVC in the `values.yaml` file.

- **Installing Spark History Server on non Data Fabric tenants:**

  To `helm install` the Spark History Server for the tenant type none, run the following command. You must use PVC as a persistent storage for event logs for the tenant type none. Create a PVC to start a Spark History Server pod and set the PVC name for the event log storage.

  ```
  helm install -f <path-to-values.yaml-file> <spark-hs-name>
  <path-to-spark-history-server-chart> \
  --namespace <tenant-namespace> \
  --set tenantNameSpace=<tenant-namespace>  \
  --set tenantIsUnsecure=true \
  --set eventlogstorage.kind=pvc \
  --set eventlogstorage.pvcname=<pvc-name>
  ```

  See Using Custom KeyStore on page 300.

Running the `helm install` installs Spark History Server in a tenant namespace.

Using Custom KeyStore

This topic describes how to use custom KeyStore for Spark History Server SSL encryption for non data-fabric (none) tenants.

A Java keystore is a repository of security certificates and their corresponding private keys used for SSL encryptions.

To use the custom KeyStore, perform the following steps:

1. Create a secret with KeyStore file in a tenant namespace.

   ```
   kubectl create secret generic "spark-ssl-secret" --from-file="./path/to/
   ssl_keystore"
   ```

   The secret must have a keystore file stored under a particular key.

2. To configure the Spark History Server with SSL configurations, set `sparkExtraConfigs` section on `values.yaml` file.

   For example, if the secret name is `spark-ssl-secret`, KeyStore name in secret is `ssl-keystore`, and passwords are `examplepass`, update the `values.yaml` file as follows:

```
sparkSsl:
  useCustomKeystore: true
  sslSecretName: "spark-ssl-secret"
  secretMountPath: /var/spark

sparkExtraConfigs: |
  spark.ssl.historyServer.enabled           true
  spark.ssl.historyServer.keyStore          /var/spark/ssl_keystore
  spark.ssl.historyServer.keyStorePassword  examplepass
  spark.ssl.historyServer.keyPassword       examplepass
  spark.ssl.historyServer.protocol          TLSv1.2
  spark.ssl.historyServer.keyStoreType      PKCS12
```

Configuring Spark Applications to Write and View Logs

This section guides you through configuring your Spark Application CRs to write logs in the event directory and view the Spark Application details in Spark web UI.

Configuring Spark Applications to Write Logs

Perform the following steps to configure the Spark Application CR to write logs to PVC:

1. Configure the `volumes` options under `spec` section of SparkApplication as follows:

```
volumes:
  -name: <some-name>
  persistentVolumeClaim:
    claimName:<same-volume-name-as-in-history-server>
```

   For example:

```
volumes:
  -name: data
  persistentVolumeClaim:
    claimName: spark-pvc
```

   You must ensure the `claimName` is the same name as `ExistingClaimName` in `values.yaml` file of the Helm chart.

2. Configure the `volumeMounts` option under `Driver` and `Executor` pods as follows:

```
volumeMounts:
  -name:<some-name>
mountPath: "<same-path-as-event-directory-on-history-server>"
```

   For example:

```
volumeMounts:
  -name: data
  mountPath: "/mnt/hs-logs"
```

   You must ensure the `mountPath` is the same path as `eventsDir` path in `values.yaml` file of the Helm chart.

3. Configure the `sparkconf` options of SparkApplication for Spark Event Log Service as follows:

```
"spark.eventLog.enabled": "true"
"spark.eventLog.dir": "<same-path-as-event-directory-on-history-server>"
```

For example:

```
"spark.eventLog.enabled": "true"
"spark.eventLog.dir": "file:/mnt/hs-logs"
```

4. Run the following command to submit the Spark Application:

```
kubectl apply -f <path-to-example-spark-application-CRs>
```

Viewing Application Details Using Web UI

You can view the application details for Completed, Failed (completed but failed), or Running Spark Applications using the Spark history web UI.

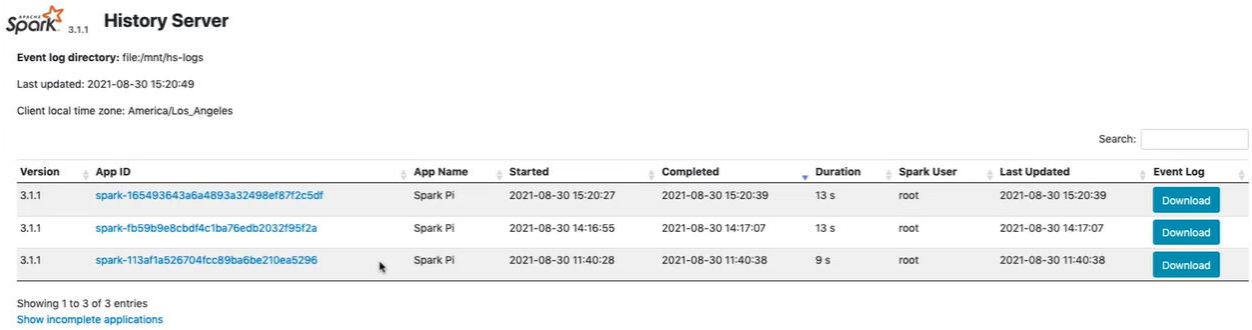


Figure 7: Spark History Server Web UI

Run the export command to get the node IP and node port to navigate to the Spark web UI.

```
export NODE_PORT=$(kubectl get --namespace {{ .Release.Namespace }} -o
jsonpath="{.spec.ports[0].nodePort}" services {{ include
"spark-hs-chart.fullname" . }})
```

```
export NODE_IP=$(kubectl get nodes --namespace {{ .Release.Namespace }} -o
jsonpath="{.items[0].status.addresses[0].address}")
```

```
echo http://$NODE_IP:$NODE_PORT
```

Access the Spark History Server web UI using the following URL:

```
http://<NODE_IP>:<NODE_PORT>
```

The default node port is 18080.

Monitor the status of all applications using the following URL:

```
http://<NODE_IP>:<NODE_PORT>/api/v1/applications
```

View the details of single application using the following URL:

```
http://<NODE_IP>:<NODE_PORT>/api/v1/applications/<spark-job-id>
```

See REST API list for Spark History Server.

NOTE: There is a limitation related to Spark History Server with Amazon S3. See Spark Limitations on page 248.

Configuring Resource Limits on Spark History Server

This section guides you through configuring resource limits for Spark History Server on `ResourceQuota` configured namespace.

A resource quota is defined by the `ResourceQuota` object. It limits the total consumption of compute resources(CPU, memory, etc.) per namespace. See Resource Quotas.

If you are using Spark History Server to monitor the Spark Applications on `ResourceQuota` configured namespace, you can configure resource limits for Spark History Server in `values.yaml` file. See `values.yaml` file in the Helm chart.

Configuring Resource Limits on Spark History Server

To configure resource limits for Spark History Server, set the resources property in `values.yaml` file.

For example:

```
resources:
   limits:
     cpu: 8000m
     memory: 8Gi
     ephemeral-storage: 30Gi
   requests:
     cpu: 2000m
     memory: 8Gi
     ephemeral-storage: 30Gi
```

You can update the resource limits and requests for Spark History Server after installation in two different ways:

1. Update the resources property in `values.yaml` file.

2. Run the `helm upgrade` command using `-set` flag.

```
helm upgrade <spark-hs-name> ./<path-to spark-hs-chart> /  -n
<your-namespace> --set resources.limits.cpu=<value>
```

For example:

```
helm upgrade spark-hs ./charts/spark-hs-2.4.7/spark-hs-chart/  -n
sampletenant --set resources.limits.cpu=10000m
```

The helm upgrade command will terminate the existing pod and create a new Spark History Server pod. Run the `export` command to get the new node IP and node port to access the Spark web UI.

```
export NODE_PORT=$(kubectl get --namespace {{ .Release.Namespace }} -o
jsonpath="{.spec.ports[0].nodePort}" services {{ include
"spark-hs-chart.fullname" . }})
```

```
export NODE_IP=$(kubectl get
nodes --namespace {{ .Release.Namespace }} -o
jsonpath="{.items[0].status.addresses[0].address}")
```

```
echo http://$NODE_IP:$NODE_PORT
```

Using Amazon S3 to Store Logs

Amazon Web Services (AWS) offers Amazon Simple Storage Service (Amazon S3). Amazon S3 provides the storage and retrieval of objects through a web service interface.

Configure the Spark History Server with existing Amazon S3 storage buckets to store the event logs.

To store logs on Amazon S3 buckets,

1. Set the following flags during Spark History Server installation. See Installing and Configuring Spark History Server on page 298.

```
--set tenantIsUnsecure=true \
--set eventlogstorage.kind=s3 \
--set eventlogstorage.s3Endpoint=http://s3host:9000 \
--set eventlogstorage.s3path=s3a://bucket/<path-to-folder> \
--set eventlogstorage.s3AccessKey=<access-key \
--set eventlogstorage.s3SecretKey=<secret-key>
```

The configuration options like `s3AccessKey` and `s3SecretKey` are passed to Spark History Server using a Kubernetes secret.

You can also securely pass the Amazon S3 credentials by setting `sparkExtraConfigs` option in `values.yaml` file.

```
sparkExtraConfigs: |
  spark.hadoop.fs.s3a.access.key [access_key]
  spark.hadoop.fs.s3a.secret.key [secret_key]
```

2. Set the following options in `values.yaml` file in a tenant namespace.

```
# Space separated Java options for Spark HS (Will be added to
SPARK_HISTORY_OPTS in spark-env.sh)
HSJavaOpts: -Dcom.sun.net.ssl.checkRevocation=false -Dcom.amazonaws.sdk.d
isableCertChecking=true
```

Deleting Spark History Server

This section describes how to delete or uninstall Spark History Server from HPE Ezmeral Runtime Enterprise.

Run the following command to delete the Spark History Server.

```
helm delete <spark-hs-name> -n <tenant-namespace>
```

For example:

```
helm delete spark-hs -n sampletenant
```

Running the `helm delete` command does not delete the PVC. You must delete he PVC manually.

Spark Thrift Server

This topic provides an overview of Spark Thrift Server.

Spark Thrift server is the Thrift JDBC/ ODBC server which corresponds to the HiveServer2 built-in Hive.

Spark Thrift server allows JDBC/ODBC clients to execute SQL queries over JDBC and ODBC protocols on Spark.

Installing and Configuring Spark Thrift Server

This section describes how to install and configure Spark Thrift Server on HPE Ezmeral Runtime Enterprise.

Prerequisites

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member in HPE Ezmeral Runtime Enterprise.
2. Install Spark Operator and enable the webhook. See Spark Operator on page 264.

About this task

In HPE Ezmeral Runtime Enterprise, you can install Spark Thrift Server using GUI or manually using the Helm chart for Apache Spark.

Learn more about supported Spark versions at Interoperability Matrix for Spark on page 246.

Installing Spark Thrift Server Using the GUI

About this task

Install the Spark Thrift Server for Apache Spark by using the HPE Ezmeral Runtime Enterprise GUI.

Procedure

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member on the HPE Ezmeral Runtime Enterprise GUI.
2. Click **Applications** in the main menu. You will see **Kubernetes Applications** tiles under **KubeDirector** tab.
3. Navigate to **Spark Thrift Server** tile and click **Launch**.
4. Configure **Cluster Detail** and **Settings** on **Create Application** screen.

Cluster Detail:	Enter the **Name** and **Description** of the application.
Settings:	Set the **CPU** and **Memory (GB)** resources.

Set the number of **Executor Cores**.

Set **Hive Metastore** with the name of a ConfigMap with `hive-site.xml` configuration of a Hive Metastore. The default ConfigMap name for a Hive Metastore installed in the same tenant is `hivesite-cm`. To learn more, see Issues and Workarounds on page 15.

In HPE Ezmeral Runtime Enterprise 5.4.0, to set **Air Gap Settings**, check **Air Gap** and set **Base Repository**, **Image**, **Image Tag**, **Image Pull Secret**. See Spark Images on page 249.

5. To view `yaml`, click **Edit/Launch yaml**.
6. Click **Submit**.

Results

Spark Thrift Server is installed in a tenant namespace.

Installing Spark Thrift Server Using the Helm

Prerequisites

1. Install and configure Helm 3.
2. Install and configure kubectl..

About this task

Install the Spark Thrift Server on Data Fabric tenants which are HPE Ezmeral Data Fabric on Kubernetes tenants or HPE Ezmeral Data Fabric on Bare Metal tenants or non Data Fabric tenants using the Helm Chart.

See Spark Thrift Server Helm Chart.

Procedure

- Helm install the Spark Thrift Server on HPE Ezmeral Runtime Enterprise:
  - **Installing Spark Thrift Server on Data Fabric tenants:** To `helm install` the Spark Thrift Server on data-fabric (`internal` or `external`) tenants, run the following command:

    ```
    helm install <spark-hs-name> ./<path-to-spark-hs-chart> -n <namespace>
    ```

  - **Installing Spark Thrift Server on non Data Fabric tenants:** To `helm install` the Spark Thrift Server on non data-fabric (`none`) tenants, run the following command:

    ```
    helm install <spark-hs-name> ./<path-to-spark-hs-chart> -n
    <namespace> --set tenantIsUnsecure=true
    ```

NOTE:

Installing the Spark Thrift Server Helm chart in a non-tenant namespace can cause error due to missing ConfigMaps and Secrets.

Running the `helm install` installs the Spark Thrift Server in a tenant namespace.

last-updated: May 06, 2024

Example

- To `helm install` the Spark Thrift Server on Data Fabric (`internal` or `external`) tenants, run the following command:

```
helm install spark-ts ./spark-ts-chart -n sampletenant
```

- To `helm install` the Spark Thrift Server on non Data Fabric (`none`) tenants, run the following command:

```
helm install spark-ts ./spark-ts-chart -n sampletenant --set
tenantIsUnsecure=true
```

Creating a Service Account

This section describes how to create a new Service Account and RBAC or use an existing Service Account for Spark Thrift Server.

When you install and configure the Spark Thrift Server using the helm chart, it does not create service account and RBAC.

To use an existing Service Account, set the following flags with `helm install` command.

```
--set serviceaccount.name=<name> --set serviceaccount.create=false
```

Alternativey, you can configure the service account options in values.yaml file of the Helm chart.

To create a new Service Account, set the following flags with `helm install` command.

```
--set serviceaccount.create=true
```

To create a new RBAC, set the following flags with `helm install` command.

```
--set rbac.create=true
```

Integrating Spark Thrift Server with Hive Metastore

This topic describes how to integrate Spark Thrift Server with Hive Metastore in HPE Ezmeral Runtime Enterprise.

You can integrate Spark Thrift Server with Hive Metastore in two ways:

Using YAML	Set `hiveSiteSource` parameter in `values.yaml` file of Spark Thrift Server Helm chart.
Using HPE Ezmeral Runtime Enterprise GUI	Set **Hive Metastore** box with the name of a ConfigMap with `hive-site.xml` configuration of a Hive Metastore during Spark Thrift Server installation. See Installing and Configuring Spark Thrift Server on page 305.

The value for `hiveSiteSource` parameter in `values.yaml` file or **Hive Metastore** box in GUI is ConfigMap. You must enter the ConfigMap with `hive-site.xml` configuration of the Hive Metastore during the Spark Thrift Server installation.

There are three separate ConfigMap values for three situations:

1. If you are installing and configuring the Spark Thrift Server in the same tenant namespace as the Hive Metastore, configure the Spark Thrift Server by using the default `hivesite-cm` ConfigMap.

   When you install the Hive Metastore in a tenant namespace, Hive Metastore auto generates a ConfigMap with the name `hivesite-cm` that contains the `hive-site.xml` configuration of the Hive Metastore.

2. If you are using Hive Metastore installed in external Data Fabric, Hive Metastore auto generates a ConfigMap with the name `hivesite-external-cm` that contains the `hive-site.xml` configuration of the Hive Metastore.

   Configure the Spark Thrift Server by using `hivesite-external-cm` ConfigMap.

   NOTE: External Data Fabric is HPE Ezmeral Data Fabric on Kubernetes configured in external Kubernetes cluster or HPE Ezmeral Data Fabric on Bare Metal.

3. If you are using the Hive Metastore installed in another namespace or some external Hive Metastore, you must manually create a ConfigMap for that Hive Metastore in the Spark Thrift Server tenant namespace.

   Example of a ConfigMap with `hive-site.xml` configuration of the Hive Metastore:

   ```
   kind: ConfigMap
   apiVersion: v1
   metadata:
     name: hivesite-cm
   data:
     hive-site.xml: <your-hive-site-configurations>
   ```

   To create a ConfigMap in the tenant namespace, run:

   ```
   kubectl apply -f -n <tenant-namespace>
   ```

   Configure the Spark Thrift Server with manually created ConfigMap.

Spark Thrift Server Feature Support

This topic describes the expected behavior and feature support for Spark Thrift Server.

Running Multiple Spark Thrift Server Instances (HA Mode)

Spark Thrift Server on Kubernetes supports HA mode using the native Kubernetes service concept. You use the `spec.tenantservices.sparkts.count` value to set the quantity of Spark Thrift Server deployments in the application configuration. All deployments are exposed through a single NodePort service. This endpoint should be used by the Spark Thrift Server clients.

Expected Behavior for Spark Thrift Jobs

Executor pods transition to a **Completed** state but are not removed after stopping or restarting the Spark Thrift Server. This is a normal behavior, and the status of the job can be checked in the completed log.

Deleting Spark Thrift Server

This section describes how to delete or uninstall Spark Thrift Server from HPE Ezmeral Runtime Enterprise.

Run the following command to delete the Spark Thrift Server.

```
helm delete <spark-ts-name> -n <tenant-namespace>
```

For example:

```
helm delete spark-ts -n sampletenant
```

Hive Metastore

This section describes enhancements to the Hive Metastore for HPE Ezmeral Runtime Enterprise.

Global Hive Metastore Support

Beginning with HPE Ezmeral Runtime Enterprise 5.3, Hive Metastore can be used outside the Kubernetes cluster, making it possible to configure a common Hive Metastore for tenants.

In HPE Ezmeral Runtime Enterprise, you can install Hive Metastore using GUI or manually using the Helm chart for Apache Spark.

Installing and Configuring Hive Metastore

This section describes how to install and configure Hive Metastore on HPE Ezmeral Runtime Enterprise.

Prerequisites

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member in HPE Ezmeral Runtime Enterprise.

About this task

In HPE Ezmeral Runtime Enterprise, you can install Hive Metastore using GUI or manually using the Helm chart for Apache Spark.

Learn more about supported Spark versions at Interoperability Matrix for Spark on page 246.

Installing Hive Metastore Using the GUI

About this task

Install the Hive Metastore by using the HPE Ezmeral Runtime Enterprise GUI.

Procedure

1. Log in as a Kubernetes Tenant Administrator or a Kubernetes Tenant Member on the HPE Ezmeral Runtime Enterprise GUI.
2. Click **Applications** in the main menu. You will see **Kubernetes Applications** tiles under **KubeDirector** tab.
3. Navigate to **Hive Metastore** tile and click **Launch**.
4. Configure **Cluster Detail** and **Settings** on **Create Application** screen.

Cluster Detail:	Enter the **Name** and **Description** of the application.
Settings:	Set the **CPU** and **Memory (GB)** resources. To set **MySQL Database Settings**, check **MySQL Database** and set **MySQL URL**, **MySQL Username**, and **MySQL Password**. For example: Enter the clear text password in the **Create Application** screen.

When you enter the clear text MySQL password in **Create Application** screen, the MySQL password will be Base64-encoded in YAML. To edit **MySQL Password** using YAML, provide the Base64-encoded value for `mysql_password` option.

For example:

```
data:
  mysqlDB: "true"
  mysql_host: ""
  mysql_user: "user"
  #Note: "Provide mysql_password as Base64-encoded value in Yaml."
  mysql_password: "YWRtaW4xMjM2Nzg="
```

In HPE Ezmeral Runtime Enterprise 5.4.0, to set **Air Gap Settings**, check **Air Gap** and set **Base Repository**, **Image**, **Image Tag**, **Image Pull Secret**. See Spark Images on page 249.

5. To view `yaml`, click **Edit/Launch yaml**.
6. Click **Submit**.

Results

The GUI installs the Hive Metastore in a tenant namespace.

Installing Hive Metastore Using the Helm

Prerequisites

1. Install and configure Helm 3.
2. Install and configure kubectl..

About this task

Install the Hive Metastore on Data Fabric tenants which are HPE Ezmeral Data Fabric on Kubernetes tenants or HPE Ezmeral Data Fabric on Bare Metal tenants or non Data Fabric tenants using the Helm chart.

See Hive Metastore Helm Chart.

To configure Hive Metastore to work with the MySQL Database, you must have a secret with MySQL server credentials in a tenant namespace. See Creating a Hive Metastore Secret on page 311.

Procedure

- Helm install the Hive Metastore on HPE Ezmeral Runtime Enterprise:
  - **Installing Hive Metastore on Data Fabric tenants:**

    To `helm install` the Hive Metastore on data-fabric (`internal` or `external`) tenants, run the following command:

    ```
    helm install <hive-metastore-name> ./<path-to-hive-metastore-chart> -n
    <tenant-namespace>
    ```

- **Installing Hive Metastore on non Data Fabric tenants:**

  To `helm install` the Hive Metastore on non data-fabric (`none`) tenants, run the following command:

  ```
  helm install <hive-metastore-name> ./<path-to-hive-metastore-chart> -n
  <tenant-namespace> --set tenantIsUnsecure=true
  ```

To configure Hive Metastore to work with the MySQL Database, set the following flags during `helm install`:

```
--set mysqlDB=true --set hiveSecret=<hive-metastore-secret-name>
```

NOTE:

Installing the Helm chart in a non-tenant namespace can cause error due to missing configmaps and secrets.

Running the `helm install` installs the Hive Metastore in a tenant namespace.

Example

- To `helm install` the Hive Metastore on Data Fabric (`internal` or `external`) tenants, run the following command:

  ```
  helm install hivemeta ./hivemeta-chart -n sampletenant
  ```

- To `helm install` the Hive Metastore on non Data Fabric (`none`) tenants, run the following command:

  ```
  helm install hivemeta ./hivemeta-chart -n sampletenant --set
  tenantIsUnsecure=true
  ```

To configure Hive Metastore to work with the MySQL Database, set the following flags during `helm install`:

```
--set mysqlDB=true --set hiveSecret=hivemeta-secret
```

More information

Interoperability Matrix for Spark on page 246
This section provides information about support and interoperability for Spark and its components with HPE Ezmeral Runtime Enterprise.

Creating a Hive Metastore Secret

This section describes how to create a secret with MySQL credentials in HPE Ezmeral Runtime Enterprise.

Prerequisites

1. Running MySQL server and its endpoint and user credentials.

About this task

Create a secret with MySQL server credentials in HPE Ezmeral Runtime Enterprise.

Procedure

1. Create an XML file containing username, password, service endpoint, and driver name.

2. To create a secret, run the following command:

```
kubectl create secret generic
<hivemeta-secret-name> --from-file=hive-site.xml=<local-xml-file-name> -n
 <tenant-namespace>
```

Example

Example of a XML file name example.xml:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>user</value>
    <description>USERNAME-FOR-MYSQL-SERVER-CONNECTION</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>password</value>
    <description>PASSWORD-FOR-MYSQL-SERVER-CONNECTION</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://"SVC_NAME.POD_NAMESPACE.svc.DNS_DOMAIN":MYSQL_PORT/
metastore_db?createDatabaseIfNotExist=true</value>
    <description>MYSQL-SERVICE-ENDPOINT-FOR-SERVER-CONNECTION</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.cj.jdbc.Driver</value>
  </property>
</configuration>
```

To create a secret named `hivemeta-secret` using `example.xml` file in a `sampletenant` namespace, run the following command:

```
kubectl create secret generic
hivemeta-secret  --from-file=hive-site.xml=example.xml -n sampletenant
```

Creating a Service Account

This section describes how to create a new Service Account and RBAC or use an existing Service Account for Hive Metastore.

When you install and configure the Hive Metastore using the Helm chart, it does not create service account and RBAC.

To use an existing Service Account, set the following flags with `helm install` command.

```
--set serviceaccount.name=<name> --set serviceaccount.create=false
```

Alternativey, you can configure the service account options in `values.yaml` file in the Helm chart.

To create a new Service Account, set the following flags with `helm install` command.

```
--set serviceaccount.create=true
```

To create a new RBAC, set the following flags with `helm install` command.

```
--set rbac.create=true
```

Customizing the Hive Metastore Configuration

This topic describes how to customize the Hive Metastore configuration in tenant namespace in HPE Ezmeral Runtime Enterprise.

The `hivemeta-cm` is the default ConfigMap for the Hive Metastore.

The `hivemeta-cm` ConfigMap is mounted into the Hive Metastore pod in the tenant namespace.

After the `helm` installation, the Hive Metastore pod configures itself and creates a `hive-site.xml` file based on the `- template-hivemeta-cm.yaml` file in the `hpe-templates-compute` namespace.

When the Hive Metastore pod is ready, the pod creates a `hivesite-cm` ConfigMap in the tenant namespace. This ConfigMap contains the updated version of `hive-site.xml` file.

To update the configuration properties of a running Hive Metastore pod, modify the `hivemeta-cm` ConfigMap in the tenant namespace and restart the pod. Delete the autogenerated `hivesite-cm` ConfigMap.

To restart the pod, run the following command:

```
kubectl rollout restart statefulset <hivemeta-pod-name> -n <namespace>
```

Accessing Spark Thrift Server Using Beeline

This topic describes how to access Spark Thrift Server using Beeline on HPE Ezmeral Runtime Enterprise.

About this task

You can use Spark Thrift Server to provide JDBC connectivity to Spark. Spark Thrift server allows JDBC or ODBC clients (for example: Beeline) to execute SQL queries over JDBC and ODBC protocols on Spark.

After setting up the tenant and HPE Ezmeral Data Fabric on Kubernetes, as described in Configuring Spark to Work with Hive Metastore on page 314, you can access the Spark Thrift Server:

Procedure

1. Get the Spark Thrift Server external endpoint. The endpoint is exposed through the `spark-ts` NodePort service. The target port is 2304.

2. Exec or ssh into the `spark-client` pod in the tenant namespace. Alternatively, you can use the `tenantcli-0` pod.

3. Use the following command to obtain a user ticket:

   ```
   maprlogin password
   ```

4. Start Beeline:

   ```
   /opt/mapr/spark/spark-[spark-version]/bin/beeline
   ```

5. Connect to the Thrift Server:

   ```
   !connect jdbc:hive2://[spark-thrift-host-and-port]/
   default;ssl=true;auth=maprsasl
   ```

More information

Configuring Spark to Work with Hive Metastore on page 314
This topic describes how to configure Spark to use Hive Metastore on HPE Ezmeral Runtime Enterprise.

Configuring Spark to Work with Hive Metastore

This topic describes how to configure Spark to use Hive Metastore on HPE Ezmeral Runtime Enterprise.

The main concept of running a Spark application against Hive Metastore is to place the correct `hive-site.xml` file in the Spark `conf` directory. To do this in Kubernetes:

- The tenant namespace should contain a ConfigMap with hivesite content (for example, `my-hivesite-cm`). Contents of the `hive-site.xml` should be stored by any key in the configmap. In default ConfigMaps, the key is `hive-site.xml`. You can create the ConfigMap manually, or use any available ConfigMap, such as the Hive Metastore default ConfigMaps.
- Assuming that your ConfigMap name is `mapr-hivesite-cm` and the key is `hive-site.xml`, you can mount it to the Spark application CR.

  For example:

```
# Declare a volume in spec
volumes:
  - name: hive-site-volume
    configMap:
      name: mapr-hivesite-cm.......
      driver:… driver spec ...

# Mount volume to driver pod
volumeMounts:
  - name: hive-site-volume
    mountPath: /opt/mapr/spark/spark-<version>/conf/hive-site.xml
    subPath: hive-site.xml
```

Related tasks

Accessing Spark Thrift Server Using Beeline on page 313
This topic describes how to access Spark Thrift Server using Beeline on HPE Ezmeral Runtime Enterprise.

Deleting Hive Metastore

This section describes how to delete or uninstall Hive Metastore from HPE Ezmeral Runtime Enterprise.

To delete the Hive Metastore, run the following command:

```
helm delete <hive-metastore-name> -n <tenant-namespace>
```

For example:

```
helm delete hivemeta -n sampletenant
```

Using Airflow to Schedule Spark Applications

This topic describes how to use Airflow to schedule Spark applications on HPE Ezmeral Runtime Enterprise.

To get started with Airflow on HPE Ezmeral Runtime Enterprise, see Airflow on page 515.

Run DAGs with SparkKubernetesOperator

To launch Spark jobs, you must select the **Enable Spark Operator** check box during Kubernetes cluster creation.

For more information, see the Apache Airflow documentation.

The following configuration changes has been made to the Airflow SparkKubernetesOperator provided by Hewlett Packard Enterprise in comparison to the open source Airflow SparkKubernetesOperator.

- Airflow SparkKubernetesOperator provided by Hewlett Packard Enterprise has three additional positional parameters at the end of the constructor:

  ```
  enable_impersonation_from_ldap_user: bool = True,
  api_group: str = 'sparkoperator.k8s.io',
  api_version: str = 'v1beta2',
  ```

  Where:

  - `enable_impersonation_from_ldap_user`: Launches Spark job with autoticket-generator
  - `api_group: str = 'sparkoperator.k8s.io'`: Specifies Spark API group
  - `api_version: str = 'v1beta2'`: Specifies Spark API version

- The API group of the open source SparkKubernetesOperator and SparkKubernetesOperator offered by Hewlett Packard Enterprise is different.

  You must set `enable_impersonation_from_ldap_user` to `False`.

See DAG Example and Spark Job Example on Hewlett Packard Enterprise GitHub repository.

To generate the appropriate ticket for a Spark job, log in to the `tenantcli` pod in the tenant namespace as follows:

```
kubectl exec -it tenantcli-0 -n sampletenant -- bash
```

Execute the following script. For the ticket name, specify a Secret name that will be used in the Spark application yaml file.

```
ticketcreator.sh
```

Creating and Connecting Tenants to HPE Ezmeral Data Fabric on Bare Metal

This topic describes how to create tenants to connect to HPE Ezmeral Data Fabric on Bare Metal not registered as Tenant Storage.

Prerequisites

Set up HPE Ezmeral Data Fabric on Bare Metal cluster. To learn more, see HPE Ezmeral Data Fabric Documentation.

Procedure

1. Create Kubernetes cluster and enable Spark Operator. See Creating a New Kubernetes Cluster on page 463 and Installing and Configuring Spark Operator on page 265.

2. Log in to HPE Ezmeral Runtime Enterprise GUI and create a default `<sampletenant>` tenant. See Creating a New Kubernetes Tenant or Project on page 452.

3. Run `kubectl get tenant sampletenant -o jsonpath={.spec}` command.

   Save the generated information about `<sampletenant>` tenant.

4. Delete existing `<sampletenant>` tenant.

   ```
   kubectl delete tenant sampletenant
   ```

5. Run `gen-external-secrets.sh` script to generate `<df-external-secrets.yaml>` file . The `gen-external-secrets.sh` script is available on HPE Ezmeral df-on-k8s tools. When prompted, Hewlett Packard Enterprise recommends changing the default names of secrets and ConfigMaps. For example: `df-external-cm`, `df-client-secrets`.

6. To create generated secrets and ConfigMaps on Kubernetes cluster, run:

```
kubectl apply -f <df-external-secrets.yaml>
```

7. Manually create an external `<sampletenant>` tenant.

```
kubectl apply -f <external-tenant-CR.yaml>
```

NOTE: This external tenant is the tenant created on HPE Ezmeral Data Fabric on Bare Metal cluster.

Ensure the following:

- Set `metadata.name` field with the same tenant name as the one created using HPE Ezmeral Runtime Enterprise GUI in step 2. For example: `sampletenant`.
- Set `spec.clustername` to HPE Ezmeral Data Fabric on Bare Metal cluster name. For example: `my.cluster.com`.
- Set `spec.security.external****` field with the same values as the ones used while running `gen-external-secrets.sh` script. You can check the secrets and ConfigMap names in `hpe-externalclusterinfo` namespace.

Example Tenant CR template:

```
apiVersion: hcp.hpe.com/v1
kind: Tenant
metadata:
  name: [tenant-name]
spec:
  clustername: [external-cluster-name]
  clustertype: external
  baseimagetag: [pick-from-default-tenant]
  imageregistry: [pick-from-default-tenant]
  loglocation: /var/lib/docker/mapr/logs
  corelocation: /var/lib/docker/mapr/cores
  podinfolocation: /var/lib/docker/mapr/podinfo
  security:
    environmenttype: hcp
    usesssd: true
    externalconfigmap: [external-cm-name]
    externalhivesiteconfigmap: [external-hivesite-cm-name]
    externalusersecret: [external-user-secret-name]
    externalserversecret: [external-server-secret-name]
    externalclientsecret: [external-client-secret-name]
  tenantservices:
    tenantcli:
      count: 1
  grouplist:
    - [pick-from-default-tenant]]
```

Results

You can now run Spark applications on the tenants created on HPE Ezmeral Data Fabric on Bare Metal cluster not registered as Tenant Storage.

Pulling Images from GCR repository on Local Workstation

This topic describes how to pull images from GCR repository on your local workstation using minikube single-node environment.

Prerequisites

1. Install minikube. See minikube documentation.
2. Start single node environment in minikube.
3. Get `imagePullSecrets` of the private GCR repository. For Spark images, see imagePullSecrets file.

About this task

Using minikube single-node environment	To pull the images from private GCR repository, you need `imagePullSecrets` and to use `imagePullSecrets`, you need Kubernetes cluster. minikube runs a single-node Kubernetes cluster on your local workstation.

You can pull images from the GCR repository by executing the `download-images.sh` script. See download-image.sh script.

Procedure

1. Download images to the local workstation. To excecute `download-images.sh` script, run:

   ```
   ./download-image.sh <complete-GCR-repo-path-to-image>
   ```

   For example:

   ```
   ./download-image.sh gcr.io/<repository-name>/spark-<version>:<image-tag>
   ```

2. Save and compress the downloaded image file as `tar` file.

   ```
   docker save -o <complete-GCR-repo-path-to-image> | gzip >
   <any-filename>.tar.gz
   ```

3. Copy the compressed `<any-filename>.tar.gz` to your desired destination location.
4. Load the docker images from `.tar.gz` file in your destination location.

   ```
   docker load <any-filename>.tar.gz
   ```

(Optional) Connect a Local Workstation

NOTE: This procedure is only required if you are planning on executing the Spark operator from your laptop or other local machine. If you are planning on running the Spark operator from the Web Terminal, then skip to Spark Operator.

To connect your local workstation to a Kubernetes tenant to execute the Spark operator:

1. Log in to HPE Ezmeral Runtime Enterprise as the Tenant Administrator user that you created in Preparing the Spark Environment.
2. Download the HPE Kubectl plugin from the **Dashboard** screen, as described in Dashboard – Kubernetes Tenant/Project Administrator.
3. Copy `kubectl-hpecp` to `/usr/local/bin/` on your local workstation, and set the path by executing the following command:

```
# export PATH=$PATH:/usr/local/bin/
```

4. Change the permissions of kubectl-hpecp by executing the following command:

```
# chmod +x /usr/local/bin/kubectl-hpecp
```

5. Establish a connection to the HPE Ezmeral Runtime Enterprise environment by executing the following command:

```
# kubectl hpecp refresh <HPECP-Gateway-LB-IP-address> --insecure
```

NOTE: Omit the `--insecure` flag if your HPE Ezmeral Runtime Enterprise deployment is configured to use HTTPS.

6. Input the username and password of the Kubernetes Cluster Administrator user.

```
                        User name with which to authenticate to HPECP:
tenadmin
                        Password for user tenadmin: <tenant_admin_password>
                        The next step is to send credentials across the
network.
                        Since the TLS connection will not be verified,
there is some risk in this.



                        Would you like to continue? [y/N] y



                        Retrieved new Kube Config from HPECP server at
<HPECP-Gateway-LB-IP-address:8080>.
                        Config file:
                        KUBECONFIG="/tenadmin/.kube/.hpecp/<HPECP-GW-IP>/
config:/tenadmin/.kube/config" kubectl config view
```

7. From the `KUBECONFIG` output of the previous step, execute the following command to set the path for your local workstation to communicate with the Kubernetes tenant:

```
# export KUBECONFIG="/john/...config:/john/.../config"
```

8. Display the Kubernetes cluster configurations to verify that you are properly connected to the Kubernetes tenant by executing the following command:

```
# kubectl config view
```

Example Output:

```
# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://hpecp-lab.mip.storage.hpecorp.net:9500
  name: HPECP K8s Demolab
contexts:
- context:
    cluster: HPECP K8s Demolab
    user: HPECP K8s Demolab-admin
  name: HPECP K8s Demolab-admin
current-context: HPECP K8s Demolab-admin
kind: Config
preferences: {}
users:
- name: HPECP K8s Demolab-admin
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1beta1
      args:
      - hpecp
      - authenticate
      - hpecp-lab.mip.storage.hpecorp.net:8080
      - --hpecp-user=admin
      - --hpecp-token=/api/v2/session/
09ad3d7c-e7e5-4345-ac14-5afcc9a8cb0e
      - --hpecp-token-expiry=1586039843
      - --force-reauth=false
      - --insecure=true
      - --insecure-skip-tls-verify=true
      command: kubectl
```

NOTE: If you receive the error connection to the server localhost:8080 was refused, then verify the previous steps.

9. Verify the Spark operator is setup by executing the following command:

```
# kubectl get pods -n <tenant-namespace>
```

The sample output will look something like this:

```
cspaceterminal-578586bff9-78pkc
        hivemeta-7dd6d76d69-87wjz
        sparkhs-5bfdc7fc8b-rmjwm
```

Kubernetes

The Kubernetes functionality in HPE Ezmeral Runtime Enterprise simplifies the creation and upgrade of virtual Kubernetes clusters that can be located on local physical hosts, virtual machines, or as cloud

instances. The flexible multi-cluster and multi-tenant control plane allows you to deploy multiple open source Kubernetes clusters and/or manage cloud Kubernetes clusters (e.g. EKS) with no lock-in or modification to native Kubernetes required.

Data engineers, ML architects, and others can spin up containerized Kubernetes environments on scalable compute clusters with their choice of machine learning tools and frameworks for Big Data use cases. Some of the key features of Kubernetes on HPE Ezmeral Runtime Enterprise include:

- HPE Ezmeral Runtime Enterprise can be installed on physical or virtual hosts located either locally (on-premises) or in a public cloud (see Controller, Gateway, and Worker Hosts).
- Pre-integrated persistent container storage (see Tenant and Project Storage and Node Storage).
- DataTaps and FS Mounts allow access to existing data sources, with no need to copy data back and forth. See About DataTaps and About FS Mounts.
- All of the above features are bundled into multi-tenant, multi-cluster management for containerized environments using open-source Kubernetes orchestration to run a variety of database, analytics, AI/ML, app modernization, CI/CD pipeline, and other applications.
- **Big Data Kubernetes tenants:** You can deploy KubeDirector applications or onboard Kubectl applications from the built-in **Kubernetes Applications** screen. See The Kubernetes Applications Screen, Deploying Applications (KubeDirector), and Onboarding Applications (Kubectl). Please also see Getting Started with General Kubernetes Functionality and AI and ML Project Workflow on page 150 for a high-level overview of the general and AI/ML Kubernetes workflows, which also contains link to additional articles with detailed instructions for each step of the process.
- Kubernetes Physical Architecture describes the physical structure (hosts, tenant, clusters, etc.) of Kubernetes within HPE Ezmeral Runtime Enterprise.

Kubernetes Physical Architecture

Workstations

Local workstations are used to do the following:

- Access the web interface.
- Directly access service endpoints running on containers via the Gateway hosts in the format `<gateway_ip>:<port>`, where `<gateway_ip>` is the IP address of a Gateway host and `<port>` is the mapped port of the service endpoint.

  For example, assume that a Kubernetes container is running a service endpoint that can be accessed remotely, and that the Gateway host has an IP address of `192.168.100.150`. If the Gateway host has mapped the service endpoint running on the Kubernetes container to Port `12345`, then you can access that endpoint by navigating to `192.169.100.150:12345`.
- Access the REST API.
- Access Kubernetes clusters using Kubeconfig and Kubectl.

Platform Control Plane

The Platform Control Plane consists of the following:

- Controller host and, if Platform HA is enabled, a Shadow Controller and an Arbiter host. See Controller, Gateway, and Worker Hosts.

  The Controller hosts authenticate users via the authentication proxy, using either the internal database or an LDAP/AD server. See User Authentication. The Authenticating Proxy consists of:

  - A server-side application that receives API requests from clients (usually from the `kubectl` tool) and (if they are properly authenticated) adds one or more groups to the request. The authenticating proxy then forwards the request to the `kube-apiserver` pod, and forwards any responses to the request back to the user.
  - A client-side `kubectl` plugin.
- One or more Gateway hosts. Gateway hosts enable access to user-facing services such as Notebooks and SSH running on containers via an instance of the High Availability Proxy service (HAproxy service on page 110). For more information about Gateway hosts, see Gateway Hosts on page 106.

The Platform Control Plane handles the installation, configuration, upgrade, and monitoring of Kubernetes hosts, clusters, and tenants.

Kubernetes Cluster Nodes

A deployment of HPE Ezmeral Runtime Enterprise can include multiple Kubernetes clusters. A host that is part of a Kubernetes cluster is referred to in Kubernetes as a node.

Each Kubernetes cluster has its own control plane, consisting of at least one control plane node. The Kubernetes control plane is separate from the Platform Control Plane. A high-availability Kubernetes cluster has multple control plane nodes, as described in High Availability on page 132.

Kubernetes clusters contain worker nodes that run the containers and pods that process jobs in HPE Ezmeral Runtime Enterprise.

For more information about hosts and Kubernetes clusters, see Controller, Gateway, and Worker Hosts.

Hewlett Packard Enterprise Distributions of Kubernetes

The Hewlett Packard Enterprise distribution of Kubernetes, identified by the `-hpe<number>` suffix, incorporates the `containerd` runtime, which is required for all Kubernetes clusters created with HPE Ezmeral Runtime Enterprise version 5.5.0 and later.

Beginning with HPE Ezmeral Runtime Enterprise 5.5.0, Hewlett Packard Enterprise provides its own CNCF-certified distribution of Kubernetes, HPE Ezmeral Runtime Enterprise, as its default Kubernetes distribution.

By creating its own distribution of Kubernetes, Hewlett Packard Enterprise can add advanced features such as security improvements. There is no change to the core Kubernetes functions and no impact to your day to day Kubernetes usage.

The `-hpe` Suffix Identifies the Kubernetes Distribution

The suffix `-hpe<number>` labels the Hewlett Packard Enterprise distributions of Kubernetes. Other than the suffix, the version numbers are equivalent to the community version of Kubernetes.

For example:

- 1.22.12-hpe1
- 1.23.9-hpe1

Runtime is `containerd`

The Kubernetes distribution provided with HPE Ezmeral Runtime Enterprise is based on the `containerd` runtime.

The `containerd` runtime is used on all hosts except for the following:

- The HPE Ezmeral Runtime Enterprise control plane hosts (Controller, Shadow Controller, Arbiter, and Gateway), which continue to use the Docker runtime.

- Kubernetes clusters that were created in on deployments running releases prior to HPE Ezmeral Runtime Enterprise 5.5.0 that are now on a deployment that has been upgraded to HPE Ezmeral Runtime Enterprise 5.5.0 or later. These legacy clusters are supported for a limited time. See Kubernetes Cluster Types and Compatibility on page 322.

Related reference

Kubernetes Cluster Types and Compatibility on page 322
An existing HPE Ezmeral Runtime Enterprise deployment that is upgraded from a previous release might contain Kubernetes clusters that use the containerd runtime and Kubernetes clusters that use the Docker runtime. All nodes in a Kubernetes cluster must use the same type of runtime.

More information

HPE Kubernetes Cluster Troubleshooting on page 935
Troubleshooting Kubernetes clusters that are running the Hewlett Packard Enterprise distribution of Kubernetes can involve examining the `.service` files, environment variables, and using journald to examine logs.

Kubernetes Cluster Types and Compatibility

An existing HPE Ezmeral Runtime Enterprise deployment that is upgraded from a previous release might contain Kubernetes clusters that use the containerd runtime and Kubernetes clusters that use the Docker runtime. All nodes in a Kubernetes cluster must use the same type of runtime.

HPE Kubernetes Clusters

The container runtime used by Hewlett Packard Enterprise distributions of Kubernetes is `containerd`.

Kubernetes clusters that are created in deployments of Hewlett Packard Enterprise 5.5.0 and later use the containerd runtime. In addition, for worker hosts that are added to the deployment, the default container runtime is also containerd.

Legacy Kubernetes Clusters

Kubernetes cluster that was created on a previous version of HPE Ezmeral Runtime Enterprise and has not been migrated to use the Hewlett Packard Enterprise distribution of Kubernetes is a legacy Kubernetes cluster. Hosts in legacy Kubernetes clusters use the Docker runtime.

Existing legacy Kubernetes clusters can be expanded, but the hosts you add to the cluster must use the Docker container runtime. Additional steps are required to prepare the host before you add the host to the deployment. See Kubernetes Worker Installation Overview on page 528.

Creating new Kubernetes clusters that use the Docker runtime is not supported.

Legacy Kubernetes clusters will be supported for a limited time after Hewlett Packard Enterprise provides a procedure to migrate from Docker to containerd. See Migrating Kubernetes Clusters from Docker to containerd on page 323.

Cluster Compatibility

If a deployment has been upgraded from an earlier version of HPE Ezmeral Runtime Enterprise with existing Kubernetes clusters, the deployment can have a mixture of legacy Kubernetes clusters and new containerd-based Kubernetes clusters. However, within a Kubernetes cluster, the same runtime (either containerd or Docker) is required.

The HPE Ezmeral Runtime Enterprise control plane hosts (Controller, Shadow Controller, Arbiter, and Gateway), are not part of a Kubernetes cluster. Control plane hosts use the Docker runtime and can manage both the new clusters and the legacy Kubernetes clusters in the same deployment.

Migrating Kubernetes Clusters from Docker to containerd

This topic describes migrating legacy Kubernetes clusters from Docker container runtime to the the new Hewlett Packard Enterprise distribution of Kubernetes, which implements containerd runtime.

Prerequisites

- You must upgrade the existing HPE Ezmeral Runtime Enterprise deployment to version 5.5.0 or later. See Upgrading to HPE Ezmeral Runtime Enterprise 5.6.x on page 885.
- You must migrate data from HPE Ezmeral Data Fabric on Kubernetes or Embedded Data Fabric to HPE Ezmeral Data Fabric on Bare Metal. See HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579.

About this task

Legacy Kubernetes clusters using Docker container runtime must be migrated to the Hewlett Packard Enterprise distribution of Kubernetes, which implements containerd runtime.

NOTE: The Kubernetes cluster is unavailable for user workloads during migration.

IMPORTANT: Setups with only 1 master host require extra hardware to ensure ETCD data is not lost.

To simplify this migration process, Hewlett Packard Enterprise can provide a command line migration tool that runs on the HPE Ezmeral Runtime Enterprise Controller node, either as root or as the user who installed HPE Ezmeral Runtime Enterprise. For more information, contact Hewlett Packard Enterprise support.

Procedure

1. Migrate the Kubernetes master hosts:
   a. Shrink the Kubernetes masters, one host at a time.
   b. Delete each Kubernetes master host removed in the previous step.
   c. Re-add each host that you deleted in the previous step as a Kubernetes host, and assign it the master role. Each newly added host comes up with the Hewlett Packard Enterprise distribution of Kubernetes and containerd runtime.
   d. Repeat this process for any remaining masters that are still on Docker container runtime.
2. Migrate the Kubernetes worker hosts:
   a. After migrating the Kubernetes masters to containerd, delete the Kubernetes workers one host at a time.
   b. Re-add each host that you deleted in the previous step as a Kubernetes worker host. Each newly added host comes up with the Hewlett Packard Enterprise distribution of Kubernetes and containerd runtime.
   c. Repeat this process for any remaining workers that are still on Docker container runtime.

Related reference

Kubernetes Cluster Types and Compatibility on page 322
An existing HPE Ezmeral Runtime Enterprise deployment that is upgraded from a previous release might contain Kubernetes clusters that use the containerd runtime and Kubernetes clusters that use the Docker runtime. All nodes in a Kubernetes cluster must use the same type of runtime.

About HPE Ezmeral Data Fabric on Kubernetes

A typical Kubernetes environment may have pods frequently coming and going. Large Kubernetes environments, such as in a public cloud, may handle pools of systems where new hosts are added to support pod and cluster placement. In HPE Ezmeral Runtime Enterprise, a Data Fabric cluster is a Kubernetes Custom Resource that functions as a storage cluster that provides access to PVCs, tenant storage, shares, and other storage needs.

HPE Ezmeral Data Fabric on Kubernetes is not supported in HPE Ezmeral Runtime Enterprise Essentials.

In a Data Fabric cluster:

- The hosts (called nodes) commit considerable disk resources that may include NVMe and enterprise-class SSDs.
- The Data Fabric cluster may only need to come up on a few nodes.
- Pods are unlikely to be deleted frequently;
- The Data Fabric CR must account for host resource profiles to guarantee core pod availability.

HPE Ezmeral Runtime Enterprise includes native support for HPE Ezmeral Data Fabric. This avoids many manual steps and allows you to create Data Fabric clusters in a manner similar to that used for creating Compute Kubernetes clusters (see Creating a New Data Fabric Cluster on page 611 and Creating a New Kubernetes Cluster on page 463). Each Data Fabric cluster resides on nodes. See Kubernetes Worker Installation Overview on page 528 and Kubernetes Data Fabric Node Installation Overview on page 531.

Features

HPE Ezmeral Runtime Enterprise automates the following functionality for a Data Fabric backed by a HPE Ezmeral Data Fabric on Kubernetes cluster:

- Pre-checking nodes before tagging them for use with HPE Ezmeral Data Fabric on Kubernetes clusters.
- Checking for sufficient resources to being up core and service pods when creating a HPE Ezmeral Data Fabric on Kubernetes cluster
- Boostrapping software installation, namespace creation, and other functions.
- Automatic Data Fabric CR creation based on scanning node system information and resource profiles. This CR helps determine how many CLDB, ZK, and MFS pods can be created and ensure proportional resource requests relative to node resources or grouped disk profiles. HPE Ezmeral Runtime Enterprise updates the standard “template' Data Fabric CR at cluster creation time. Users may view/download the Data Fabric CR after cluster creation.
- Auto-registration of Tenant Storage/PVCs, along with clean-up functionality to allow deregistration if needed for another Data Fabric cluster.
- Data Fabric clusters automatically become the default storageclass for Compute Kubernetes clusters.
- Gateway hosts (see Gateway Hosts) expose HPE Ezmeral Data Fabric services such as the HPE Ezmeral Data Fabric Control System, Kibana, and Grafana via clickable links in the web interface.
- User-settable configuration parameters allow fine-tuning a cluster to suit specific needs. See User-Configurable Data Fabric Cluster Parameters.
- Data Fabric clusters can be expanded by adding additional nodes, as described in Expanding a Data Fabric Cluster on page 616. The original cluster size and the number and composition of new node determine whether CLDB, ZK, and/or MFS pods will be added. Once expanded, a Data Fabric cluster cannot be shrunk.

- HPE Ezmeral Data Fabric packages can be started automatically when creating a Kubernetes cluster in HPE Ezmeral Runtime Enterprise. The user can also select Compute packages to install by clicking the available options during cluster creation.
- The POSIX client type (“Basic” or “Platinum”) can be specified on a per-node basis.

Limitations

The following limitations apply to HPE Ezmeral Data Fabric on Kubernetes clusters:

- Only one HPE Ezmeral Data Fabric on Kubernetes cluster can be created. This one HPE Ezmeral Data Fabric on Kubernetes cluster therefore registers the Tenant Storage and Share for all Kubernetes tenants.
- Migrating from an integrated/embedded form of HPE Ezmeral Data Fabric (versions 5.1.1. and below) to an HPE Ezmeral Data Fabric on Kubernetes cluster (versions 5.2 and above) requires manual steps. Contact HPE Technical Support for assistance.

Kubernetes Tenant RBAC

Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise.

The following three key elements are involved in Kubernetes RBAC:

- **Subjects:** The set of users and processes that want to access the Kubernetes API.
- **Resources:** The set of Kubernetes API Objects available in the cluster. Examples include Pods, Deployments, Services, Nodes, and PersistentVolumes, among others.
- **Verbs:** The set of operations that can be executed to the resources above. Different verbs are available (examples: get, watch, create, delete, etc.), but ultimately all of them are Create, Read, Update or Delete (CRUD) operations.

With these three elements in mind, the key idea of RBAC is the Context subjects, API resources, and operations. In other words, we want to specify which operations can be executed over a set of resources for a given user.

Creating a new Kubernetes tenant via the web interface creates a corresponding set of roles and role bindings within the namespace of that new tenant. Each role is assigned a set of resources and allowed CRUD operations. Creating a Kubernetes tenant creates the following roles:

- Administrator. See Default Admin RBACS on page 326.
- Member. See Default Member RBACS on page 328.
- SA (not used). See Default SA (Service Account) RBACS on page 332.

Kubernetes roles and assigned resources/operations are stored in the file `/opt/bluedata/common-install/bd_mgmt/bd_mgmt_default_tenant_k8s.cfg` on the host. Platform Administrator users may add, edit, or delete roles by editing this file, which will change the allowed defaults for all Kubernetes tenants created after the changes have been made.

NOTE: Adding, editing, and/or deleting roles/privileges by making changes to `bd_mgmt_default_tenant_k8s.cfg` does not affect Kubernetes tenants that were created prior to making the changes.

If you need to edit the RBACs for a running Kubernetes tenant:

1. Access the Kubernetes tenant as either the Platform Administrator or the Kubernetes Cluster Administrator for the cluster that contains the affected tenant.

2. Execute this command on any Kubernetes master node:

```
kubectl edit hpecptenants.hpecp.hpe.com -n hpecp
```

3. Make and then save your desired changes.

Default Admin RBACS

```
- roleID: admin
    rules:
    - apiGroups:
      - ""
      resources:
      - bindings
      - podtemplates
      - replicationcontrollers
      - pods
      - resourcequotas
      - services
      - serviceaccounts
      - endpoints
      - persistentvolumeclaims
      - events
      - configmaps
      - secrets
      - pods/exec
      - pods/log
      - pods/portforward
      verbs:
      - '*'
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - roles
      - rolebindings
      verbs:
      - '*'
    - apiGroups:
      - apps
      resources:
      - controllerrevisions
      - statefulsets
      - deployments
      - replicasets
      verbs:
      - '*'
    - apiGroups:
      - deployment.hpe.com
      resources:
      - hpecpmodels
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
    - apiGroups:
      - kubedirector.hpe.com
      resources:
      - kubedirectorclusters
      - kubedirectorapps
```

```
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
      - patch
    - apiGroups:
      - hpecp.hpe.com
      resources:
      - hpecpfsmounts
      - hpecptenants
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      - ingresses
      verbs:
      - '*'
    - apiGroups:
      - policy
      resources:
      - poddisruptionbudgets
      - poddisruptionbudgets/status
      verbs:
      - '*'
    - apiGroups:
      - metrics.k8s.io
      resources:
      - pods
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - authorization.k8s.io
      resources:
      - localsubjectaccessreviews
      verbs:
      - '*'
    - apiGroups:
      - autoscaling
      resources:
      - horizontalpodautoscalers
      verbs:
      - '*'
    - apiGroups:
      - batch
      resources:
      - cronjobs
      - jobs
      verbs:
      - '*'
    - apiGroups:
      - coordination.k8s.io
      resources:
```

```
      - leases
      verbs:
      - '*'
    - apiGroups:
      - discovery.k8s.io
      resources:
      - endpointslices
      verbs:
      - '*'
    - apiGroups:
      - snapshot.storage.k8s.io
      resources:
      - volumesnapshots
      verbs:
      - '*'
    - apiGroups:
      - sparkoperator.k8s.io
      resources:
      - scheduledsparkapplications
      - sparkapplications
      verbs:
      - '*'
    - apiGroups:
      - sparkoperator.hpe.com
      resources:
      - scheduledsparkapplications
      - sparkapplications
      verbs:
      - '*'
    - apiGroups:
      - machinelearning.seldon.io
      resources:
      - seldondeployments
      verbs:
      - '*'
    - apiGroups:
      - serving.kubeflow.org
      resources:
      - inferenceservices
      verbs:
      - '*'
    - apiGroups:
      - kubeflow.org
      resources:
      - pytorchjobs
      - tfjobs
      - experiments
      verbs:
      - '*'
```

Default Member RBACS

```
- roleID: member
    rules:
    - apiGroups:
      - ""
      resources:
      - pods
      - bindings
      - podtemplates
      - replicationcontrollers
      - resourcequotas
```

```
      - services
      - endpoints
      - persistentvolumeclaims
      - events
      - configmaps
      - pods/log
      - pods/portforward
      verbs:
      - '*'
    - apiGroups:
      - ""
      resources:
      - pods/exec
      verbs:
      - get
    - apiGroups:
      - ""
      resources:
      - secrets
      verbs:
      - get
      - create
      - update
      - patch
    - apiGroups:
      - apps
      resources:
      - controllerrevisions
      - daemonsets
      - statefulsets
      - deployments
      - replicasets
      verbs:
      - '*'
    - apiGroups:
      - deployment.hpe.com
      resources:
      - hpecpmodels
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
    - apiGroups:
      - kubedirector.hpe.com
      resources:
      - kubedirectorclusters
      verbs:
      - create
      - update
      - delete
      - get
      - list
      - watch
      - patch
    - apiGroups:
      - kubedirector.hpe.com
      resources:
      - kubedirectorapps
      verbs:
      - create
      - get
```

```
      - list
      - watch
    - apiGroups:
      - hpecp.hpe.com
      resources:
      - hpecpfsmounts
      - hpecptenants
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      - ingresses
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - policy
      resources:
      - poddisruptionbudgets
      - poddisruptionbudgets/status
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - metrics.k8s.io
      resources:
      - pods
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - authorization.k8s.io
      resources:
      - localsubjectaccessreviews
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - autoscaling
      resources:
      - horizontalpodautoscalers
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - batch
      resources:
      - cronjobs
      - jobs
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - coordination.k8s.io
```

```
      resources:
      - leases
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - discovery.k8s.io
      resources:
      - endpointslices
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - snapshot.storage.k8s.io
      resources:
      - volumesnapshots
      verbs:
      - get
      - list
      - watch
    - apiGroups:
      - sparkoperator.k8s.io
      resources:
      - scheduledsparkapplications
      - sparkapplications
      verbs:
      - create
      - update
      - get
      - list
      - watch
    - apiGroups:
      - sparkoperator.hpe.com
      resources:
      - scheduledsparkapplications
      - sparkapplications
      verbs:
      - create
      - update
      - get
      - list
      - watch
    - apiGroups:
      - machinelearning.seldon.io
      resources:
      - seldondeployments
      verbs:
      - '*'
    - apiGroups:
      - serving.kubeflow.org
      resources:
      - inferenceservices
      verbs:
      - '*'
    - apiGroups:
      - kubeflow.org
      resources:
      - pytorchjobs
      - tfjobs
      - experiments
```

```
      verbs:
      - '*'
```

Default SA (Service Account) RBACS

```
  - roleID: sa
    rules:
    - apiGroups:
      - ""
      resources:
      - pods
      - resourcequotas
      - serviceaccounts
      - services
      - endpoints
      - persistentvolumeclaims
      - events
      - configmaps
      - secrets
      - pods/exec
      verbs:
      - '*'
    - apiGroups:
      - rbac.authorization.k8s.io
      resources:
      - roles
      - rolebindings
      verbs:
      - '*'
    - apiGroups:
      - apps
      resources:
      - daemonsets
      - statefulsets
      - deployments
      - replicasets
      verbs:
      - '*'
    - apiGroups:
      - deployment.hpe.com
      resources:
      - hpecpmodels
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
    - apiGroups:
      - kubedirector.hpe.com
      resources:
      - kubedirectorclusters
      - kubedirectorapps
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
      - patch
```

```
    - apiGroups:
      - hpecp.hpe.com
      resources:
      - hpecpfsmounts
      - hpecptenants
      verbs:
      - get
      - list
      - watch
      - create
      - update
      - delete
    - apiGroups:
      - networking.k8s.io
      resources:
      - networkpolicies
      - ingresses
      verbs:
      - '*'
```

Disabling or Enabling the Kubernetes Web Terminal

As a Platform Administrator, you can enable or disable user access to the Kubernetes Web terminal. The Kubernetes Web Terminal is not available in HPE Ezmeral Runtime Enterprise Essentials.

Prerequisites

Required Access Rights: Platform Administrator

About this task

The Kubernetes Web Terminal is accessible from the HPE Ezmeral Runtime Enterprise Web UI by default. Disabling the Web Terminal removes the Kubernetes Web terminal button from the HPE Ezmeral Runtime Enterprise Web UI. The DISABLE_WEBTERM setting is a global setting that applies to all Kubernetes clusters and to all users.

Procedure

1. Execute the following command on the Controller host:

   ```
   vi /usr/share/bdswebui/bdswebui/settings.py
   ```

2. Do one of the following:

   - To disable user access to the Kubernetes Web terminal, change the value of DISABLE_WEBTERM to `True`.

     ```
     'DISABLE_WEBTERM' : True,
     ```

   - To enable user access to the Kubernetes Web terminal, change the value of DISABLE_WEBTERM to `False`.

     ```
     'DISABLE_WEBTERM' : False,
     ```

3. Save and close the file.

4. Restart the httpd service by executing the following command:

```
systemctl restart httpd.service
```

5. If platform HA is enabled, then repeat this procedure on the Shadow Controller host.

More information

Kubernetes Web Terminal on page 349
The Kubernetes Web Terminal includes the HPE Kubectl plug-in, Helm, and access to the Kubernetes tenant FS mounts. Kubernetes Web Terminal is not available in HPE Ezmeral Runtime Enterprise Essentials. Privileges to execute commands are granted according to the user role.

Kubernetes Metadata

NOTE: This page is intended for Kubernetes administrators and other advanced Kubernetes users.

HPE Ezmeral Runtime Enterprise includes two sets of custom resources and two operators running in the `hpecp` namespace that manage those resources:

- The `kubedirector` operator manages the `kubedirectorcluster`, `kubedirectorapp`, and `kubedirectorconfig` resources.
  - `kubedirectorconfig` is a singleton resource that does not usually require modification.
  - The `kubedirectorapp` (app definition) and especially `kubedirectorcluster` (app instantiation) resources are more likely to be actively created, edited, etc. KubeDirector is an open-source project documented on GitHub, but some aspects of its behavior are documented in this article (link opens an external website in a new browser tab/window).
- The `hpecp-agent` operator manages the `hpecptenant`, `hpecpfsmount`, and `hpecpconfig` resources.
  - `hpecpconfig` is a singleton resource that does not usually require modification.
  - `hpecptenant` (located in the `hpecp` namespace) models a tenant.
  - `hpecpfsmount` (located in a tenant namespace) models an FS Mount that was likely created in a tenant.

These custom resources can have labels and annotations on them that communicate useful information about their properties or context. Users who manually/explicitly create Kubernetes pod and service resources can also choose to add certain labels or other properties to those objects to trigger additional feature behaviors in the `hpecp-agent` operator.

Labels That Can be Used to Trigger Features

- Pod label usable to trigger FS mount: `hpecp.hpe.com/fsmount: <FS mount namespace>`

  (Can be auto-generated by HPECP Agent for KubeDirector pods; see below.)

- Pod label usable to trigger DataTap setup: `hpecp.hpe.com/dtap`

  (The value is not important, just the label key existence.)

- Service label usable to control gateway mapping (NodePort only): `hpecp.hpe.com/hpecp-internal-gateway: <"true" or "false">`

  (Can be auto-generated by HPECP Agent; see below.)

- Service label usable to force port name (single-port service only): `hpecp.hpe.com/portname-override: <desired port name>`

  (This label is useful for tools like `kubectlexpose` that don't allow direct specification of port names.)

Other Feature Controls

If the port name within a Kubernetes service object starts with the prefix `http-` or `https-`, then this can affect its exposure through the Gateway host and the web interface:

- Only endpoints with such port name prefixes will get clickable links in the **Kubernetes Service Endpoints screen**. See Kubernetes Service Endpoints Tab.
- If `https-` prefixed, then that UI link will correctly be an https link regardless of the SSL configuration (or lack thereof) for the Gateway hosts.
- If `http-` prefixed, and if the Gateway does not support SSL termination, then the service will be exposed as normal http through the Gateway and the interface links.
- If `http-` prefixed, and if the gateway supports SSL termination, then this service will get SSL termination at the gateway, and the interface link will be https.

Services and port names generated by KubeDirector will always have a port name prefix that comes from the `urlScheme` for that endpoint, as defined by the KubeDirector app. Manual explicit port naming is therefore usually only of interest when you are creating http/https services outside of KubeDirector.

Labels Generated by KubeDirector

The labels generated by KubeDirector on any statefulset, pod, or service (either per-member or headless) are:

- `kubedirector.hpe.com/kdcluster: <kdcluster resource name>`
- `kubedirector.hpe.com/kdapp: <kdapp resource name>`
- `kubedirector.hpe.com/appCatalog: <either local or system>`

Labels generated by KubeDirector on any statefulset, pod, or per-member service created by KubeDirector :

- `kubedirector.hpe.com/role: <kdapp role ID>`

Labels generated by KubeDirector on any statefulset or pod created by KubeDirector :

- `kubedirector.hpe.com/headless: <name of headless cluster service>`

Labels generated by HPECP Agent on any statefulset pod created by KubeDirector :

- `hpecp.hpe.com/fsmount: <pod namespace>` (only created by HPECP Agent if label does not already exist in the statefulset pod template)

Labels generated by HPECP Agent on any NodePort service:

- `hpecp.hpe.com/hpecp-internal-gateway: <true or false>` (only created if label does not already exist; if in a tenant namespace, the value is driven by the tenant setting; otherwise `false`.)

Labels generated by HPECP Agent on any namespace associated with an HPE Ezmeral Runtime Enterprise Tenant:

- `hpecp.hpe.com/hpecptenant: <hpecptenant resource name>`

User-Requested Labels through KubeDirector

- The optional `podLabels` array in a role in a KubeDirector cluster can be used to specify additional labels to put on its generated statefulset pods, and/or to override the labels that would normally be generated for those pods. For example, this is used on cluster admin webterms to mount all FS mount namespaces.
- The optional `serviceLabels` array in a role in a KubeDirector cluster can be used to specify additional labels to put on its generated member services, and/or override the labels that would normally be generated for those services. E.g. our platform uses this on webterms to enable gateway mapping (setting `hpecp.hpe.com/hpecp-internal-gateway` to `true`) even though the webterm is not in a tenant namespace.

Annotations Generated by KDKubeDirector

Annotations generated by KDKubeDirector on any statefulset, pod, or service created by KubeDirector :

- `kubedirector.hpe.com/kdapp-prettyName: <KD app label name>`

Annotations generated by HPECP Agent on any service where gateway mapping is enabled:

- `hpecp-internal-gateway/<pod port>: <gateway hostname>:<gateway port>`

Other Conventions

HPE Ezmeral Runtime Enterprise gives the following annotations to a Kubernetes tenant resource. These annotations are not required by the tenant CRD, but they are useful as FYIs for anyone examining the tenant object.

- `hpecp-tenant: <HPECP tenant path, e.g. "/api/v2/tenant/4">`
- `hpecp-tenant-name: <HPECP tenant label name>`

The tenant Kubernetes resource name also always follows the convention `hpecp-tenant-<tenant ID>`. For example, if the tenant in the API is `/api/v2/tenant/44`, then the Kubernetes resource will be named `hpecp-tenant-44`. Some functionality around reporting existing tenant/namespace associations relies on this convention.

Centralized Policy Management

Defines centralized policy management and describes the features and benefits of applying policies to Kubernetes clusters managed by HPE Ezmeral Runtime Enterprise. Not available in HPE Ezmeral Runtime Enterprise Essentials.

This feature is not available in HPE Ezmeral Runtime Enterprise Essentials.

What Is Centralized Policy Management?

Policy management is the fine-grained control of objects in your Kubernetes cluster using pre-written policies. *Centralized* policy management is the ability to define and manage policies stored in a Git repository and apply them automatically to Kubernetes clusters managed by HPE Ezmeral Runtime Enterprise.

Challenges Addressed by Centralized Policy Management

Centralized policy management addresses some specific challenges faced by operations personnel in managing Kubernetes clusters:

- **Maintaining control over sprawling Kubernetes clusters**

  Because it is relatively easy to create Kubernetes clusters on premises, off premises, or in the cloud, many installations have too many of them. The nature and number of Kubernetes clusters can make it difficult to apply and govern policies consistently.

- **Inconsistent policies or a lack of policies pose a security threat**

  Inconsistent policies or a lack of policies increase the management burden on operations personnel, rendering clusters less secure.

- **Manual policy management is tedious and burdensome for operations**

  Policy drifts are hard to govern manually. This can lead to an endless cycle of defining and redefining and deploying and redeploying policies.

Features of Centralized Policy Management

The centralized policy management product capabilities in HPE Ezmeral Runtime Enterprise 5.3 provide the following features:

- **Git integration**

  Git integration enables policies to be stored (backed up) in a source-control repository. For more information about GitOps, see What is GitOps?

- **Policy enforcement through an admission controller**

  The HPE Ezmeral Runtime Enterprise policy controller leverages OPA Gatekeeper as an admission controller to validate and enforce policies on the cluster. OPA Gatekeeper is installed as a system add-on. For more information about OPA Gatekeeper, see Open Policy Agent.

- **Drift detection, reconciliation, and automatic policy synchronization (Argo CD)**

  HPE Ezmeral Runtime Enterprise leverages Argo CD as the policy synchronizer engine for the continuous monitoring of policies on running Kubernetes clusters. The policy synchronizer watches for policy drifts and reconciles the changes by automatically synchronizing with the centralized policy defined in Git. Synchronization ensures policy immutability and the continuous compliance of each Kubernetes cluster.

  Versions 5.3 and later of the HPE Ezmeral Runtime Enterprise deploy Argo CD as a system add-on in every Kubernetes cluster created by the platform.

  HPE Ezmeral Runtime Enterprise uses Argo CD only for synchronization and policy validation. For more information about Argo CD, see Argo CD - Declarative GitOps CD for Kubernetes.

Benefits of Centralized Policy Management

Centralized policy management offers the following benefits:

- **Policy guardrails ensure consistent clusters across hybrid installations**

  Policies serve as a blueprint for creating your clusters. Once applied, the policies are immutable and can only be changed by updating them in Git. This makes policies secure and centrally governed.

- **Policies ensure continuous compliance, control, and improved operations efficiency**

  Policies give you greater control over objects, and the same policies can be applied to multiple clusters, ensuring consistency in your deployments.

- **Policies are subject to version control**

  With version control, you can maintain multiple different versions of policies. And if you apply a policy that does not work as intended, you can roll back the policy.

Limitations of Centralized Policy Management

See Limitations of Centralized Policy Management on page 348.

Viewing Policy Management Information

Describes how to view information about the Git repositories and policies currently being used by HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

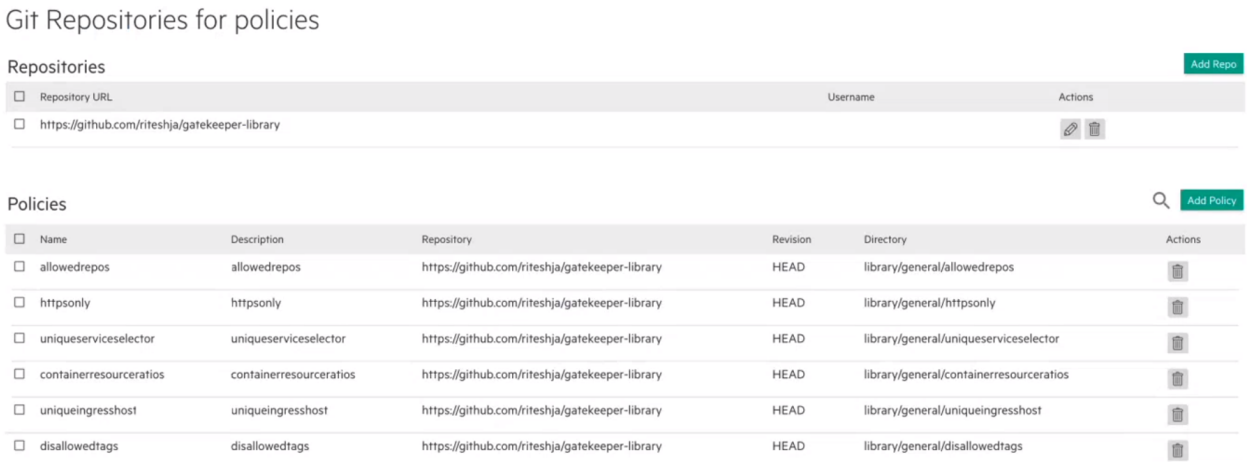
About this task

The **Policy Management** tab shows the Git repositories and GitOps policies currently configured for the platform. From this tab, you can view basic details about repositories and policies. You can also add, edit, or delete a repository or policy.

Procedure

1. Log in to the web interface for HPE Ezmeral Runtime Enterprise, as described in Launching and Signing In on page 136.
2. In the main menu, click the **Policy Management** tab.

   For example:



Viewing Policy Violations

Describes how to view a detailed log of policy violations and denials triggered on a Kubernetes cluster managed by HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

About this task

The **Policy Violations** tab shows a detailed log of violations and denials triggered on a Kubernetes cluster by policies created with Centralized Policy Management on page 336.

Procedure

1. In the main menu, click the **Clusters** tab. The **Clusters** screen opens.

2. Click the name of a cluster. The **<cluster-name>** screen opens.

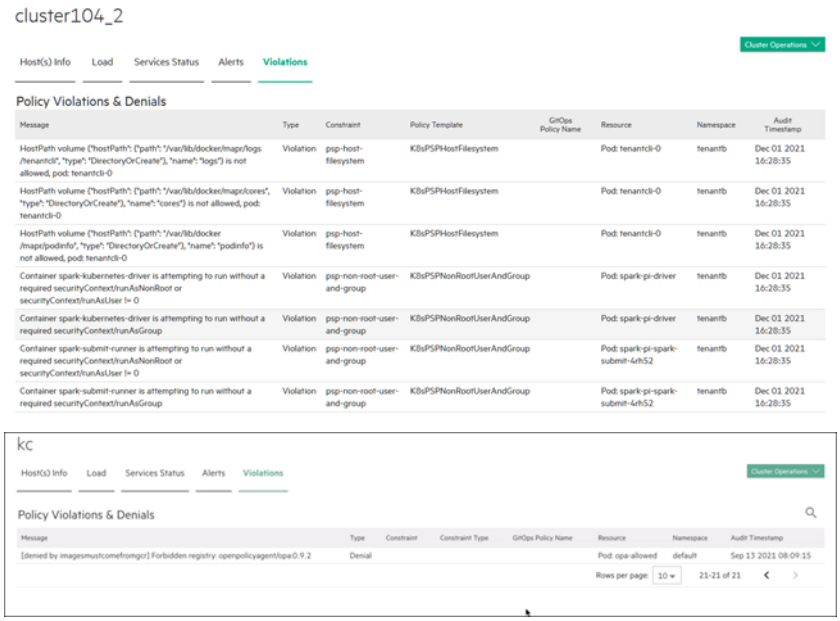
3. Click the **Violations** tab to view detailed information about policy violations and denials triggered on this cluster.

   About violations and denials:

   - A policy **violation** will be listed when any *pre-existing resources* attempt to perform an action which violates the policies present in the cluster.

   - A policy **denial** will be listed when the *creation of resources* which violate the policies present in the cluster is prevented.

   For example:



Configuring Centralized Policy Management

List of the major tasks for configuring centralized policy management.

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

About this task

This topic lists the major tasks for configuring centralized policy management. Each task can consist of multiple steps.

Procedure

- Create policies. See Creating Policies for Centralized Policy Management on page 340.

- Configure a Git repository to store the policy YAML files. See Creating the Git Repository for Centralized Policy Management on page 342.

- Add the Git repository to the HPE Ezmeral Runtime Enterprise. See Adding a Git Repository for Centralized Policy Management on page 343.

- Add policies to the policy list. See Adding a Policy for Centralized Policy Management on page 344.

- Register policies with your cluster. See Registering Policies with Your Kubernetes Cluster on page 345 .
- Log on to the Argo CD server to view a dashboard of your policies. See Logging in to the Argo CD Server on page 347.

Creating Policies for Centralized Policy Management

Required access rights: Platform Administrator or Cluster Administrator

In HPE Ezmeral Runtime Enterprise Centralized Policy Management, policies are expressed as a directory of YAML files in a Git repository. Each YAML file contains one or more pairs of OPA Gatekeeper *constraint* and *template* objects.

Rego Policy Language

To write OPA Gatekeeper template objects, you need to learn Rego. Rego is the policy language for OPA Gatekeeper. For more information about Rego and working with policies and constraints, see these resources:

- Rego
- Policies and Constraints
- How to use Gatekeeper

Organizing Template and Constraint Objects

You can organize pairs of template and constraint objects in two ways:

- **Combine multiple template and constraint objects into one YAML file.** This “one big YAML file” becomes a collection of policies – or one big policy – that includes pairs of templates and constraints. See this example (`onebigpolicy.yaml`).
- **Create a directory of policies with each policy represented as a single YAML file** that contains a pair of constraint and template objects. See this example directory.

Example Policy

The following example policy (`allowedrepo-policy.yaml`) validates all pods in the cluster and ensures that they come from the `openpolicyagent` repo.

In this example, the constraint object appears first, followed by the template object. The template object contains logic for how the policy should be validated. In the template object, lines of code in **bold face** indicate Rego commands. The constraint object contains the values that the template will validate against.

If a pod that is not from the `openpolicyagent` repo is detected, an error is generated.

```
---
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sAllowedRepos
metadata:
  name: repo-is-openpolicyagent
  annotations:
    argocd.argoproj.io/sync-options: SkipDryRunOnMissingResource=true
spec:
  match:
    kinds:
    - apiGroups: [""]
      kinds: ["Pod"]
    namespaces:
    - "default"
  parameters:
```

```
    repos:
    - "openpolicyagent"
---
apiVersion: templates.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: k8sallowedrepos
  annotations:
    description: Requires container images to begin with a repo string from
a specified
      list.
spec:
  crd:
    spec:
      names:
        kind: K8sAllowedRepos
      validation:
        # Schema for the `parameters` field
        openAPIV3Schema:
          properties:
            repos:
              type: array
              items:
                type: string
  targets:
  - target: admission.k8s.gatekeeper.sh
    rego: |
      package k8sallowedrepos
      violation[{"msg": msg}] {
        container := input.review.object.spec.containers[_]
        satisfied := [good | repo = input.parameters.repos[_] ; good =
startswith(container.image, repo)]
        not any(satisfied)
        msg := sprintf("container <%v> has an invalid image repo
<%v>, allowed repos are %v", [container.name, container.image,
input.parameters.repos])
      }
      violation[{"msg": msg}] {
        container := input.review.object.spec.initContainers[_]
        satisfied := [good | repo = input.parameters.repos[_] ; good =
startswith(container.image, repo)]
        not any(satisfied)
        msg := sprintf("container <%v> has an invalid image repo
<%v>, allowed repos are %v", [container.name, container.image,
input.parameters.repos])
      }
```

Required Annotation for All OPA Gatekeeper Constraint Objects

As shown in the constraint section of the example, all OPA Gatekeeper constraint objects must include the following annotation:

```
argocd.argoproj.io/sync-options: SkipDryRunOnMissingResource=true
```

Policy Enforcement Example

After your policies are created and applied to a cluster, you can observe the enforcement of them when operations violate a policy. The following example shows the effect of applying an object that violates multiple policies configured for a cluster:

```
# kubectl apply -f disallowedcontainerlimit.yaml
Error from server ([denied by container-image-must-have-digest] container
<opa> uses an image with a digest <openpolicyagent/opa:0.9.2>
[denied by container-must-have-limits] container <opa> memory limit <2Gi>
is higher than the maximum allowed of <1Gi>
[denied by must-have-probes] Container <opa> in your <Pod> <opa-disallowed>
has no <readinessProbe>
[denied by must-have-probes] Container ,opa> in your <Pod>
<opa-disallowed> has no <livenessProbe>): error when creating
"disallowedcontainerlimit.yaml": admission
 webhook "validation.gatekeeper.sh" denied the request: <denied by
container-image-must-have-digest] container <opa> uses an image without a
digest <openpolicyagent/opa:0.9.2>
[denied by container-must-have-limits] container <opa> memory limit <2Gi>
is higher than the maximum allowed of <1Gi>
[denied by must-have-probes] Container <opa> in your <Pod> <opa-disallowed>
has no <readinessProbe>
[denied by must-have-probes] Container <opa> in your <Pod> <opa-disallowed>
has no <livenessProbe>
```

Git Repository Containing Example Policies (for HPE-Internal Personnel)

The following HPE Git repository contains policy examples tested with HPE Ezmeral Runtime Enterprise:

https://github.hpe.com/hpe/opa-gatekeeper-policies

Overly Restrictive Policies

As with any security system, it is possible to create policies that interfere with normal system operations and that result in unwanted behavior. For example, when creating policies for HPE Ezmeral Runtime Enterprise, setting the root file system directory to “read only” results in numerous errors, because fsmount daemonset pods must have write access to the `/opt/bluedata/share` directory on all of the Kubernetes hosts and the `/opt/bluedata/share` directory inside the pod. One such error is the failure to configure a Kubernetes Web Terminal.

To display a JSON-formated list of policy violations that are occurring in a cluster, enter the following command:

```
kubectl get constraints -o json
```

Creating the Git Repository for Centralized Policy Management

Required access rights: Platform Administrator or Cluster Administrator

You can designate either a public or private Git repository for use with centralized policy management. But you must configure the repository before adding it. For information about creating a repository, see Creating a New Repository.

If you create a public repository, the HPE Ezmeral Runtime Enterprise must have Internet access. To facilitate Internet access, you must configure a web proxy on all of the platform hosts and run the installer with the `--proxy` option. For more information, see Web Proxy Requirements on page 821 and Standard Installation on page 854.

Directory Structure for Policies

A policy typically consists of two files (a `template.yaml` and `constraint.yaml`) in the same directory. The web interface displays this directory as a single policy. See the following example. If you combine multiple `template.yaml` and `constraint.yaml` files in the same directory, the objects will be applied, but all of the policies that they represent will be displayed as one policy. For more information about the directory structure, see Creating Policies for Centralized Policy Management on page 340.

Policies

Name	Description	Repository	Revision	Directory
allowedrepos	allowedrepos	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/allowedrepos
httpsonly	httpsonly	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/httpsonly
uniqueserviceselector	uniqueserviceselector	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/uniqueserviceselector
containerresourceratios	containerresourceratios	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/containerresourceratios
uniqueingresshost	uniqueingresshost	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/uniqueingresshost
disallowedtags	disallowedtags	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/disallowedtags
containerlimits	containerlimits	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/containerlimits
requiredprobes	requiredprobes	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/requiredprobes
requiredlabels	requiredlabels	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/requiredlabels
externalip	externalip	https://github.com/riteshja/gatekeeper-library	HEAD	library/general/externalip

Rows per page: 10 1-10 of 12

Adding a Git Repository for Centralized Policy Management

Describes how to add a Git repository for centralized policy management.

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

About this task

This page describes how to add the Git repository for centralized policy management. You must create the Git repository before you can add it. For more information, see Creating the Git Repository for Centralized Policy Management on page 342.

Procedure

1. On the **Policy Management** tab, click **Add Repo**. The **Add Repository** dialog box appears:

Add Repository

Repository URL* https://github.com/your_repo.git

Username

Password

Client TLS Cert Browse

Client TLS Cert Key Browse

Server TLS Cert Browse

Skip server verification

Submit

2. Specify the repository configuration information.

   The following table describes each field. An asterisk (*) at the end of a field name indicates a required field:

Field	Description
Repository URL*	HTTP-based URL of the Git repository.
Username	Username to authenticate to the Git repository using HTTP.
Password	Password or Access Token to authenticate to the Git repository using HTTP.
Client TLS Cert	Client TLS cert to authenticate to the Git repository using HTTP.
Client TLS Cert Key	Client TLS cert key to authenticate to the Git repository using HTTP.
Server TLS Cert	Certificate for server verification.
Skip server verification	Whether to skip the verification of Git server.

3. Click **Submit**.

Adding a Policy for Centralized Policy Management
Describes how to add a policy for use with policy management.

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

About this task

This page describes how to add a policy for use with policy management. You must create the policy before you can add it. For more information, see Creating Policies for Centralized Policy Management on page 340.

Adding a policy using these steps does not immediately apply the policy to the cluster. You must register the policy before it is applied to the cluster.

Procedure

1. On the **Policy Management** tab, click **Add Policy**. The **Add Policy** dialog box appears:

2. Specify the policy configuration information.

   The following table describes each field. An asterisk (*) at the end of a field name indicates a required field:

Field	Description
Name*	Enter a unique name for the policy.
Description	Enter details about the policy.
Repository*	Repository resource as source for this policy.
Target Revision	Git branch name, tag, commit sha tag or symbolic reference like HEAD to which the application will sync.
Path*	Specify a directory in the repository to sync to.

3. Click **Submit**. The policy is added to the policy list and can be registered with multiple clusters. See Registering Policies with Your Kubernetes Cluster on page 345.

Editing a Policy for Centralized Policy Management

HPE Ezmeral Runtime Enterprise 5.4 does not support editing a policy.

Only adding or deleting a policy is currently supported. Therefore, to edit a policy, you must delete the policy and add it back again with the desired changes. See Deleting a Repository or Policy for Centralized Policy Management on page 345 and Adding a Policy for Centralized Policy Management on page 344.

Deleting a Repository or Policy for Centralized Policy Management

Required access rights: Platform Administrator or Cluster Administrator

This page describes how to delete a repository or policy.

Deleting a Repository

Deleting a repository is not allowed if policies are using that repository.

To delete a repository:

1. In the **Policy Management** tab, select the repository.
2. Click the trash can (**Delete**) icon. The repository is deleted immediately, and a confirmation message appears:

```
Git repository <repo-name> deleted successfully.
```

Deleting a Policy

Deleting a policy is not allowed if the policy is registered with one or more Kubernetes clusters.

To delete a policy:

1. In the **Policy Management** tab, select the policy.
2. Click the trash can (**Delete**) icon. The policy is deleted immediately, and a confirmation message appears:

```
Policy <policy-name> deleted successfully.
```

Registering Policies with Your Kubernetes Cluster

Describes how to register policies with a Kubernetes cluster.

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

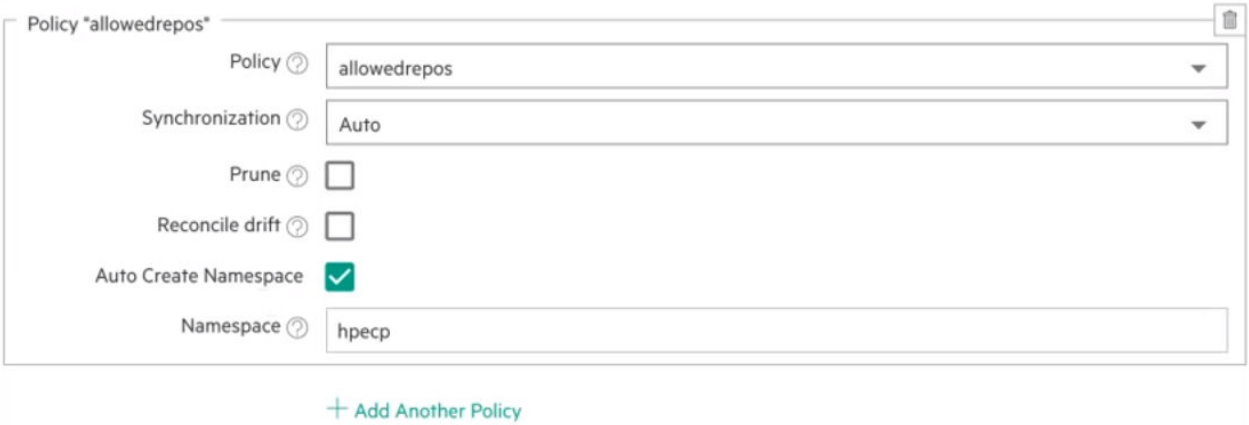
About this task

Before you can use a policy, you must register it with your cluster. You can register policies during cluster creation or after the cluster is created. Also, any policies that you want to register with a cluster must first be added to the policy list, as described in Adding a Policy for Centralized Policy Management on page 344.

The following procedure assumes that the cluster has already been created.

Procedure

1. Click the icon to edit the Kubernetes cluster, and navigate to the **Application Configurations** page.
2. In the **Policy Settings** box, click **Add Policy** or **Add Another Policy**. A new policy field appears. For example:



3. Specify the desired values for the policy.

   The following table describes each field:

Field	Description
Policy	Displays the policies that have been added to the policy list. Click the drop-down arrow to select the policy that you want to register with the cluster.
Synchronization	Controls the automatic synchronization of policies. Possible values are **Auto** and **Manual**. Select **Auto** to configure the auto-sync feature. If you select **Auto**, ArgoCD synchronizes the policy with the cluster as soon as the policy is registered with the cluster. If you select **Manual**, the policy is in ArgoCD, but it is not applied to the cluster.
Prune	Specifies if resources should be pruned during auto-syncing. With Prune specified, if the objects in a policy are deleted from the policy directory, the corresponding objects are deleted from the cluster.
Reconcile Drift	Activates drift detection. Specifies if partial app sync should be executed when resources are changed only in the target Kubernetes cluster and no git change is detected. When the feature is selected and ArgoCD detects a change in the policy, the policy is reverted back to the state before the change.
Auto Create Namespace	Controls the namespace in which the policy objects are created. The default namespace is `hpecp`. When this box is checked, if the namespace doesn't exist, it is created automatically.
Namespace	Specifies the namespace to be created by the Auto Create Namespace option.

4. After policies are registered with the cluster, you can use the Policy Viewer link, as described in Logging in to the Argo CD Server on page 347.

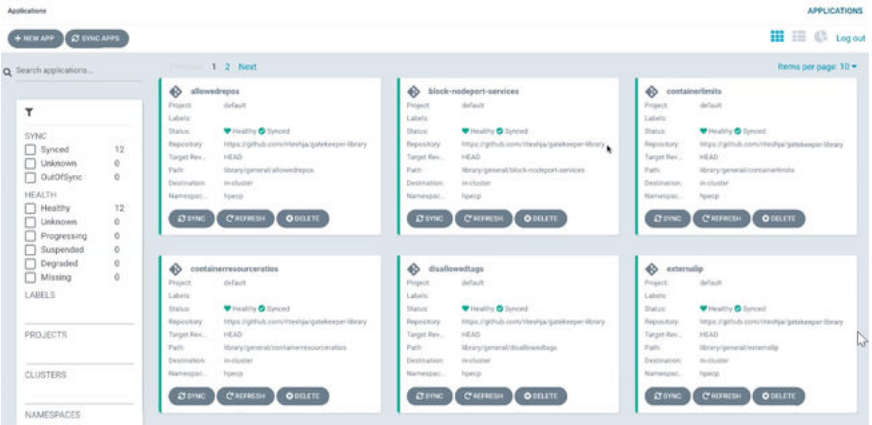
Logging in to the Argo CD Server

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

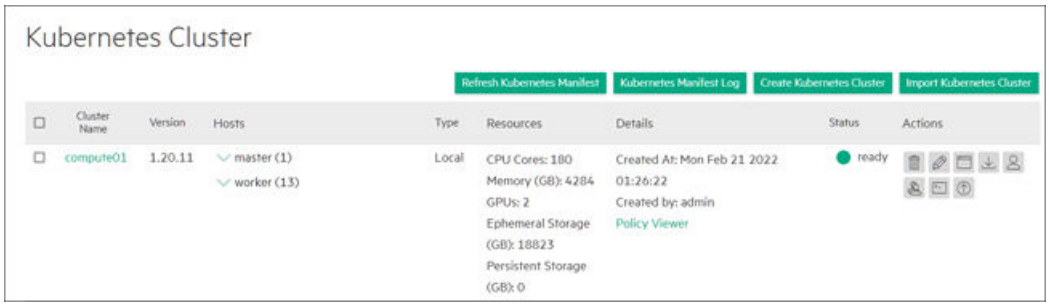
About this task

After policies are registered with your Kubernetes cluster, you can log in to the Argo CD server to view a dashboard of your policies:



Procedure

1. Navigate to the **Clusters** tab:



2. Click the **Policy Viewer** link. The Argo login page is displayed.

3. Enter a **Username**. The default user is `admin`.

4. Enter a **Password**.

   To obtain the default password, enter the following command on a Kubernetes master node:

   ```
   kubectl -n argocd get secret argocd-initial-admin-secret -o
   jsonpath="{.data.password}" | base64 -d
   ```

   To change the password, see this FAQ (link opens an external site in a new browser tab or window).

5. Click **SIGN IN**.

Deregistering a Policy for Centralized Policy Management

Describes how to remove a policy that has already been registered with the cluster.

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

About this task

Deregistering a policy removes the policy from the list of associated policies for the cluster and ensures that the policy will no longer be enforced by the cluster.

Procedure

1. Click the **Clusters** tab.
2. Click the **Edit Cluster** icon to edit the Kubernetes cluster.
3. Click **Next** repeatedly to advance through the screens until the **Application Configurations** page appears. The **Policy Management** tab shows the list of registered policies.
4. In the pane for the policy that you want to deregister, click the **Remove** (trash can) icon.
5. Click **Submit** to submit the **Edit Cluster** changes. Once the edit has completed successfully, the cluster transitions to a "ready" state. If the edit failed, the cluster might transition to a "warning" or "error" state. If that happens, check the error in the cluster setup log.
6. To confirm that the policy was deregistered, create an object that would have been blocked when the policy was registered. After deregistration, creation of the object should be allowed.

Limitations of Centralized Policy Management

This page describes some limitations that apply to the current release of centralized policy management.

Limitation for Imported Clusters

The policy-management framework included in HPE Ezmeral Runtime Enterprise has not been tested for use in imported clusters.

Limitation for hpecp-bootstrap-argocd Deployment Object

If you register or deregister a policy after the cluster is created, you must make sure that none of the policies that you are registering or deregistering blocks the `hpecp-bootstrap-argocd` deployment object from scaling up. The container platform uses the `hpecp-bootstrap-argocd` deployment object to register and deregister policies.

This limitation applies only if you register or deregister a policy **after** the cluster is created. A workaround for this limitation is to log on to the Argo CD Server as described in Logging in to the Argo CD Server on page 347 and add the policy directly to Argo CD. For instructions, see Creating Apps Via UI.

To make sure that none of the policies that you are registering or deregistering blocks the `hpecp-bootstrap-argocd` deployment object:

1. Compare your policy definition (template and constraint objects) against the definition of the `hpecp-bootstrap-argocd` deployment to make sure the policy does not block deployment. To display the definition of the `hpecp-bootstrap-argocd` deployment, use one of these commands:

   ```
   kubectl get deployment hpecp-bootstrap-argocd -n hpecp-bootstrap -o json
   ```

   or

   ```
   kubectl describe deployment hpecp-bootstrap-argocd -n hpecp-bootstrap
   ```

2. After comparing, modify or create your policy to allow the `hpecp-bootstrap-argocd` deployment to scale up. Or modify the `hpecp-bootstrap-argocd` deployment to conform to the policy you are creating.

3. If necessary, edit the `hpecp-bootstrap-argocd` deployment using the following command:

```
kubectl edit deployment hpecp-bootstrap-argocd -n hpecp-bootstrap
```

Kubernetes Troubleshooting Overview

The following articles contain information on troubleshooting Kubernetes:

- Kubernetes Installation Issues
- Kubernetes Node Issues
- Kubernetes Cluster Creation Issues
- Tenant Management Issues
- Node Port Service Issues
- Web Interface Issues
- General Kubernetes Application/Deployment Issues
- Issues and workarounds in the HPE Ezmeral Runtime Enterprise Release Notes on page 11

You can also find links to additional support and troubleshooting information in Troubleshooting Overview.

Using Kubernetes

The topics in this section describe information and tasks for non-administrator users of Kubernetes in HPE Ezmeral Runtime Enterprise.

Kubernetes Web Terminal

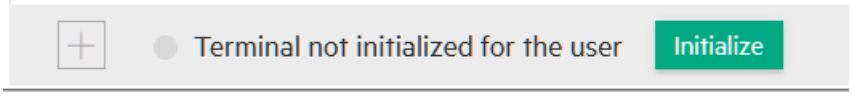
The Kubernetes Web Terminal includes the HPE Kubectl plug-in, Helm, and access to the Kubernetes tenant FS mounts. Kubernetes Web Terminal is not available in HPE Ezmeral Runtime Enterprise Essentials. Privileges to execute commands are granted according to the user role.

Accessing the Kubernetes Web Terminal

To access the Kubernetes Web Terminal:

1. Log in to the web interface, and then navigate to the appropriate Kubernetes cluster or tenant according to your credentials and role (Member, Tenant Administrator, or Cluster Administrator.)

2. Click the green **Initialize** button that appears at the bottom of most Kubernetes screens within the web interface.



   The screen displays the message: **Waiting for terminal to be ready** or **Connecting to the terminal** and the green **Initialize** button is replaced by a red Terminate button.

   If this is the first time you are accessing the Web Terminal, it takes a few minutes for the Web Terminal to be ready because HPE Ezmeral Runtime Enterprise must launch a new webterm service pod.

3. Once the Web Terminal is ready, click the **Launch** icon (plus sign) to launch the terminal window.

NOTE:

The Kubernetes Web Terminal enables CLI command execution, but it does not implement a fully functional terminal. For example, using the `vi` command to edit a file might only show a partial file if it is a large file. You can enlarge the screen and use the small font option (default is `Regular`) to see fit more lines in the window. However, it might not be possible to see the entire file if it is large. To work around this issue, you can do one of the following:

- Execute the `cat/more` command to view the file.
- Edit the file on your local machine and then upload it using an FS mount.

The Web Terminal environment includes Kubectl, and the appropriate kubeconfig is configured. This configuration behaves in the same way as a locally downloaded config, as described in Role Privileges. You should never need to manually refresh or recreate the kubeconfig.

This example shows the `kubectl config view` command. In this example, the Member user does not have the ability to execute the command `kubectl get namespaces`.

```
k8suser@kd-977sb-0:~$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://mip.storage.enterprise.net:9500
  name: clust1
  contexts:
  - context:
      cluster: clust1
      namespace: hpecp-tenant-4-gtx9s
      user: hpecp-admin
    name: clust1-eng-tenant-admin
  current-context: clust1-eng-tenant-admin
  kind: Config
  preferences: {}
  users:
  - name: hpecp-admin
    user:
      exec:
        apiVersion: client.authentication.k8s.io/v1beta1
        args:
        - epic
        - authenticate
        - mip-bd-vm38.mip.storage.enterprise.net:8080
        - --hpecp-user=admin
        - --hpecp-token=/api/v2/session/37391bb6-fac9-44a0-ae08-cf0806bd54bf
        - --hpecp-token-expiry=1574976938
        - --insecure=true
        command: kubectl
        env: null
k8suser@kd-977sb-0:~$ kubectl get namespaces
error: You must be logged in to the server (Unauthorized)
```

Kubernetes Role Privileges

Users who perform Kubernetes API operations in a namespace through the built-in authentication proxy (see Kubernetes Physical Architecture), will have privileges in that namespace as granted by the role they have (if any) in the corresponding Kubernetes cluster or tenant. If the user has a Platform Administrator role or a Kubernetes Cluster Administrator role in the current cluster, then that user has those access rights regardless of any explicit tenant role assignments that user may also have.

The following screens show the Kubernetes ACLs for Kubernetes Member and Kubernetes Tenant Administrator users.

NOTE: This information is a sample that is subject to change. You can view the current ACLs by user role by executing the commands, listed in bold, from the Kubernetes Web Terminal. For more information about ACLs, see Kubernetes Tenant RBAC.

```
# kubectl describe role hpecp-tenant-4-member-99zrv -n my-tenant-namespace
 Name:         hpecp-tenant-4-member-99zrv
 Labels:       <none>
 Annotations:  <none>
 PolicyRule:
   Resources                                        Non-Resource URLs  Resource
Names  Verbs
   ---------                                        -----------------
 --------------  -----
   configmaps                                       []
[]              [*]
   endpoints                                        []
[]              [*]
   events                                           []
[]              [*]
   namespaces                                       []
[]              [*]
   persistentvolumeclaims                           []
[]              [*]
   pods/exec                                        []
[]              [*]
   pods/logs                                        []
[]              [*]
   pods                                             []
[]              [*]
   resourcequotas                                   []
[]              [*]
   secrets                                          []
[]              [*]
   services                                         []
[]              [*]
   daemonsets.apps                                  []
[]              [*]
   deployments.apps                                 []
[]              [*]
   replicasets.apps                                 []
[]              [*]
   statefulsets.apps                                []
[]              [*]
   networkpolicies.networking.k8s.io                []
[]              [*]
   rolebindings.rbac.authorization.k8s.io           []
[]              [*]
   roles.rbac.authorization.k8s.io                  []
[]              [*]
   storageclasses.storage.k8s.io                    []
[]              [*]
   kubedirectorclusters.kubedirector.bluedata.io    []
[]              [create update delete get list watch]
   hpecpfsmounts.hpecp.hpe.com                      []
[]              [get list watch]
   hpecptenants.hpecp.hpe.com                       []
[]              [get list watch]
   kubedirectorapps.kubedirector.bluedata.io        []
[]              [get list watch]
   poddisruptionbudgets.policy/status               []
```


last-updated: May 06, 2024

```
[]              [get list watch]
   poddisruptionbudgets.policy                    []
[]              [get list watch]

 # kubectl describe role hpecp-tenant-4-admin-g8vtg -n my-tenant-namespace
 Name:         hpecp-tenant-4-admin-g8vtg
 Labels:       <none>
 Annotations:  <none>
 PolicyRule:
   Resources                                      Non-Resource URLs  Resource
Names  Verbs
   ---------                                      -----------------
 --------------  -----
   configmaps                                     []
[]              [*]
   endpoints                                      []
[]              [*]
   events                                         []
[]              [*]
   namespaces                                     []
[]              [*]
   persistentvolumeclaims                         []
[]              [*]
   pods/exec                                      []
[]              [*]
   pods/logs                                      []
[]              [*]
   pods                                           []
[]              [*]
   resourcequotas                                 []
[]              [*]
   secrets                                        []
[]              [*]
   serviceaccounts                                []
[]              [*]
   services                                       []
[]              [*]
   daemonsets.apps                                []
[]              [*]
   deployments.apps                               []
[]              [*]
   replicasets.apps                               []
[]              [*]
   statefulsets.apps                              []
[]              [*]
   networkpolicies.networking.k8s.io              []
[]              [*]
   poddisruptionbudgets.policy/status             []
[]              [*]
   poddisruptionbudgets.policy                    []
[]              [*]
   rolebindings.rbac.authorization.k8s.io         []
[]              [*]
   roles.rbac.authorization.k8s.io                []
[]              [*]
   storageclasses.storage.k8s.io                  []
[]              [*]
   hpecpfsmounts.hpecp.hpe.com                    []
[]              [get list watch create update delete]
   hpecptenants.hpecp.hpe.com                     []
[]              [get list watch create update delete]
   kubedirectorapps.kubedirector.bluedata.io      []
[]              [get list watch create update delete]
```

```
   kubedirectorclusters.kubedirector.bluedata.io  []
[]              [get list watch create update delete]
```

Related tasks

Disabling or Enabling the Kubernetes Web Terminal on page 333
As a Platform Administrator, you can enable or disable user access to the Kubernetes Web terminal. The Kubernetes Web Terminal is not available in HPE Ezmeral Runtime Enterprise Essentials.

Using the HPE Kubectl Plugin

The `kubectl-hpecp` binary is a `kubectl` plugin that can be installed from a Kubernetes **Dashboard** screen. For detailed information about `kubectl` plugins, see the official Kubernetes docs (link opens an external website in a new browser tab or window).

This plugin includes the following commands:

- Version
- Refresh
- Authenticate

Usage Notes

When you use the kubectl plugin from a headless system, SAML authentication will not work natively. Instead, download the kubeconfig file from the HPE Ezmeral Runtime Enterprise UI and install it on the headless system either one of the following locations:

- `~/.kube/config`
- A path pointed to by the KUBECONFIG environment variable.

For all OS types, ensure that the `kubectl` executable and the `kubectl-hpecp` plugin executable are made available on the user's path.

The sample commands in this topic will vary for Windows users because all commands output by `kubectl-hpecp` are intended to be run in `bash` or `zsh` on the Linux and MacOS operating systems, while all commands for Windows are intended to be run in `cmd.exe`.

Version Command

The `kubectl hpecp version` command prints a version-specific string to the console in either a valid JSON or YAML object, as specified by the flags passed in to the command. For example (on MacOS):

```
 > kubectl hpecp version
{
  "major": "3",
  "minor": "0",
  "gitVersion": "v3.0-159",
  "gitCommit": "15d398acdc03760f0ce269acdf88cc4b5d8cd7e1",
  "gitTreeState": "clean",
  "buildDate": "2020-02-11 00:02:04",
  "goVersion": "go1.13.7",
  "compiler": "gc",
  "platform": "darwin/amd64"
}
>  kubectl hpecp version --output=yaml
major: "3"
minor: "0"
gitversion: v3.0-159
gitcommit: 15d398acdc03760f0ce269acdf88cc4b5d8cd7e1
gittreestate: clean
```

```
builddate: "2020-02-11 00:02:04"
goversion: go1.13.7
compiler: gc
platform: darwin/amd64
```

Refresh Command

The `kubectl hpecp refresh` command gets the user a new Kubeconfig specific to their needs, as specified by the Kubernetes KUBECONFIG documentation (link opens an external website in a new browser tab or window). This new Kubeconfig contains only contexts that the user can interact with, based on the user's assigned role. See Kubernetes Tenant RBAC.

If HPE Ezmeral Runtime Enterprise is set up for SAML, the user is taken through the SAML login workflow. This requires the user to have a compatible browser. See also Usage Notes on page 353.

The semantics for the command are as follows:

```
>  kubectl hpecp refresh
<ip_address-or-host_alias-or-hostname> --insecure --hpecp-user=<new_username>
 --hpecp-pass=<new_password>
User name with which to authenticate to HPECP:
<username>
Password for user [admin]: <password>
The next step is to send credentials across the network.
Since the TLS connection will not be verified, there is
some risk in this.

Would you like to continue? [y/N]
y
Got a new kubeconfig from the server.

Retrieved new Kube Config from HPECP server at hpe-2:8080.
The KUBECONFIG environment variable HAS NOT been set.
Your current session WILL NOT have the new configuration.
To persist these changes by loading all current Kube Config
values into your default Kube Config file, run the
following command:

    KUBECONFIG="/Users/tom/.kube/.hpecp/hpe-2/config:/Users/tom/.kube/
config-backup" kubectl config view --raw > /Users/tom/.kube/config

To persist these changes by changing your local KUBECONFIG
environment variable, run the following command:

    export KUBECONFIG="/Users/tom/.kube/.hpecp/hpe-2/config"

CAUTION - both of these commands will OVERWRITE your current
Kube Config settings. This is probably what you want, but
to confirm that this command will not break your system,
run the following command to view the resulting Kube
Config file:

    KUBECONFIG="/Users/tom/.kube/.hpecp/hpe-2/config:/Users/tom/.kube/config"
kubectl config view
```

Where:

- `<ip_address-or-host_alias-or-hostname>` is the IP address, alias, or hostname of the host on which to perform the refresh.
- `<username>` is the user name of the current user, assuming that the `--hpecp-user` flag is not present.

- `<password>` is the password of the current user, assuming that the `--hpecp-pass` flag is not present.
- `--insecure` is used when the HPE Ezmeral Runtime Enterprise API is not protected by TLS. This situation is not common.
- `--hpecp-user` is optionally used when you want to authenticate to the server as a user (the `<new_username>`) that is different from the currently logged-in user.
- `--hpecp-pass` is optionally used to supply a different password (the `<new_password>`), such as when using the `--hpecp-user` flag.

Altering the Kubeconfig for a user is potentially risky, since doing so overwrites any item that has a name conflict. For example, if two `kubeconfig` files have a user with the name `john`, only the first `kubeconfig` to register the name `john` will show up in the final `kubeconfig` file.

If a user is expected to interact with more than one HPE Ezmeral Runtime Enterprise deployment, then Hewlett Packard Enterprise recommends configuring each deployment with different custom install names. Custom install names function as a human-readable differentiator between the deployments.

For example, if the user `john` is expected to interact with two different HPE Ezmeral Runtime Enterprise deployments, and if that user received a different `kubeconfig` file from each deployment, then that user cannot use both `kubeconfigs` in the same context, because the user name `HPECP-john` would not be unique between the `kubeconfigs`. However, if each deployment has a custom install name (such as `test` and `prod`), then `john` can interact with both systems from the same context, because the user name on each deployment is different. The user from the `prod` deployment is `prod-john`, and the user from the `test` deployment is `test-john`.

There are some other circumstances that are covered by the `kubectl-hpecp refresh` command. To view the command-line help, run the `kubectl hpecp --help` command.

Authenticate Command

This command retrieves the current authentication object from the file system. Plugin users should never need to call this command manually.

General Functionality

The topics in this section describe general Kubernetes functionality on HPE Ezmeral Runtime Enterprise.

Getting Started with General Kubernetes Functionality

The Kubernetes workflow allows you to add dedicated hosts, create one or more clusters, add one or more tenants to a cluster, and then create virtual nodes/containers that run in virtual clusters (pods) to run Kubernetes applications. This workflow consists of three high-level steps that must be performed by users with different roles in the following order:

- Kubernetes Administrator
- Kubernetes Cluster Administrator
- Kubernetes Tenant/Project Administrator
- Kubernetes Tenant Member

Kubernetes Administrator

1. Log into the web interface as a Kubernetes Administrator, as described in Launching and Logging In.
2. Verify that HPE Ezmeral Runtime Enterprise is licensed for at least the number of CPU cores that will be used for the new Kubernetes cluster. See and License Tab.

3. If needed, configure LDAP/AD authentication.

   See Configuring User Authentication Settings.

4. If you will be using HPE Ezmeral Data Fabric on Kubernetes and have not done so already, then add one or more data fabric nodes, as described in Kubernetes Data Fabric Node Installation Overview. See also About HPE Ezmeral Data Fabric on Kubernetes on page 324.

5. Add one or more Kubernetes Worker hosts, as described in Kubernetes Worker Installation Overview.

6. If one does not already exist and if you will be using HPE Ezmeral Data Fabric on Kubernetes, then create a Data Fabric cluster, as described in Creating a New Data Fabric Cluster. You may create a single Data Fabric cluster per HPE Ezmeral Runtime Enterprise deployment. If needed, you may expand an existing Data Fabric cluster, as described in Expanding a Data Fabric Cluster on page 616.

7. Create a Kubernetes cluster, as described in Creating a New Kubernetes Cluster on page 463.

8. Assign at least one user to be a Kubernetes Administrator for the Kubernetes cluster you just created. See Managing Kubernetes Admin Users (to assign a user role using local authentication) or Updating External Kubernetes Administrator Groups (to assign a user role using LDAP/AD groups).

Kubernetes Cluster Administrator

1. Confirm that the Kubernetes Administrator has completed all of the steps described in Kubernetes Administrator, above.

2. Log in to the web interface as a Kubernetes Cluster Administrator, as described in Launching and Logging In.

3. If needed, use the Kubernetes Dashboard and/or Web Terminal to configure the Kubernetes cluster, as described in Accessing the Kubernetes Dashboard and Kubernetes Web Terminal, respectively. See Kubernetes Tenant RBAC for the privileges allowed to Kubernetes Cluster Administrator users.

4. Assign the Kubernetes Tenant Administrator and/or Kubernetes Member roles to the appropriate users, as described in Viewing and Assigning Kubernetes Cluster Users.

Kubernetes Tenant Administrator

1. Confirm that the Kubernetes Cluster Administrator has completed all of the steps described in Kubernetes Cluster Administrator, above.

2. If needed, use the Kubernetes Dashboard and/or Web Terminal to configure the Kubernetes cluster, as described in Accessing the Kubernetes Dashboard and Kubernetes Web Terminal, respectively. See Kubernetes Tenant RBAC for the privileges allowed to Kubernetes Tenant Administrator users.

3. Create one or more DataTaps to allow the tenant to access remote data storage resources. See About DataTaps and Creating a New DataTap.

4. Assign one or more Kubernetes Member roles to the appropriate users, as described in Viewing and Assigning Kubernetes Tenant Users.

Kubernetes Tenant Member

1. Confirm that the Kubernetes Tenant Administrator has completed all of the steps described in Kubernetes Tenant Administrator, above.

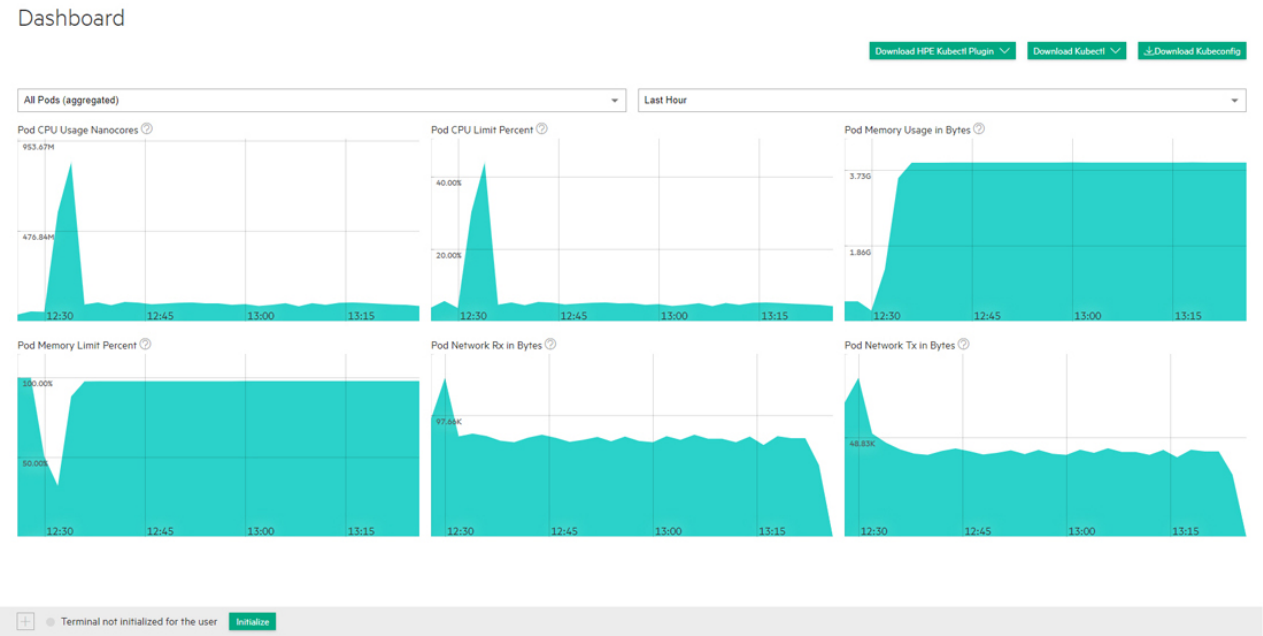
2. Log in to the web interface as the Kubernetes Member user that was created or assigned in Step 4 of the Kubernetes Tenant Administrator workflow described above.

3. Either:
   - Deploy KubeDirector apps, as described in Deploying KubeDirector Apps.
   - Onboard Kubectl apps, as described in Onboarding Kubectl Apps.

You may also use the Kubernetes Dashboard and/or Web Terminal to deploy Kubernetes objects such as pods, as described in Accessing the Kubernetes Dashboard and Kubernetes Web Terminal, respectively. See Kubernetes Tenant RBAC for the privileges allowed to Kubernetes Tenant Member users.

Dashboard - Kubernetes Tenant Member

Users who are logged into a Kubernetes tenant with the Member role can access the Kubernetes Member **Dashboard** screen by selecting **Dashboard** in the main menu.



The top of this screen has three buttons that allows you to download the plugins that you need to access Kubernetes pods within a cluster. The buttons are:

- **Download HPE Kubectl Plugin:** Downloads the HPE installer for the `kubectl` command line tool for controlling a Kubernetes cluster. Please click here for more information (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl, below, and Using the HPE Kubectl Plugin.
- **Download Kubectl:** Downloads the generic installed for the `kubectl` command line tool for controlling a Kubernetes cluster. Please click here for more information (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl, below.

NOTE: You may see a warning that `kubectl-hpecp` cannot be opened because the publisher cannot be verified. You may safely ignore this warning and proceed with the installation.

- **Download Kubeconfig:** Downloads the `kubeconfig` file that configures access to Kubernetes when used in conjunction with either the `kubectl` command line tool or other clients. Please click here for more information (link opens an external website in a new browser tab/window).

The top of this screen has two pull-down menus that allow you to filter the data by pod and time frame. You may also choose to view information for all applications or only for KubeDirector applications by moving the **Filter KubeDirector Applications** slider. Hovering your mouse over the graphs displays a popup with additional information. The following charts are available:

- **Pod CPU Use Nanocores:** Number of CPU nanocores in use.
- **Pod CPU Limit Percent:** Percentage of maximum number of pods that are currently running inside the current cluster.
- **Pod Memory Usage in Bytes:** Bytes of memory being used.
- **Pod Memory Limit Percent:** Percentage of memory limit being used.
- **Pod Network Rx in Bytes:** Bytes received over the network.
- **Pod Network Tx in Bytes:** Bytes transmitted over the network.
- **GPU Utilization (percent):** If GPUs are present, displays aggregate GPU utilization in percent.
- **GPU Memory Usage:** If GPUs are present, displays aggregate GPU memory usage in percent.

NOTE: Please see Downloading Kubernetes Usage Details for information about how to download detailed usage and uptime information in comma-delimited (.csv) format.

Installing Kubectl

To install Kubectl on your local system:

1. Download either of the Kubectl plugins:
   - If you are on a Windows system, then this download will be an .exe file.
   - If you are on a UNIX system, then you will need to execute one of the following commands:
     - **HPE Kubectl:** `chmod +x kubectl-hpecp`
     - **Generic Kubectl:**`chmod +x kubectl`
2. Place the Kubectl executable into a folder that is on your system's PATH.
3. Execute the command `kubectl hpecp refresh {HPE Ezmeral Runtime Enterprise controller/gateway ip address}`. If HTTPS is not enabled, then add the argument `--insecure=true`.

Toolbar & Main Menu - Kubernetes Tenant Member

Describes the toolbar and navigation sidebar available to users with Kubernetes Tenant Member access rights to tenants that are not ML Ops tenants in HPE Ezmeral Runtime Enterprise.

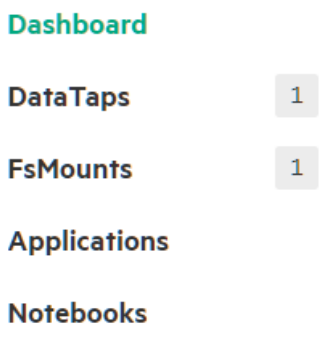
This article describes the UI items for Tenant Members accessing Kubernetes tenants that are not ML Ops projects.

Toolbar

The layout of the Toolbar is the same as described in Navigating the GUI on page 143.

Main Menu - Kubernetes Tenant Member

The Kubernetes Tenant Member main menu for tenants that are not ML projects appears as shown in the following image:

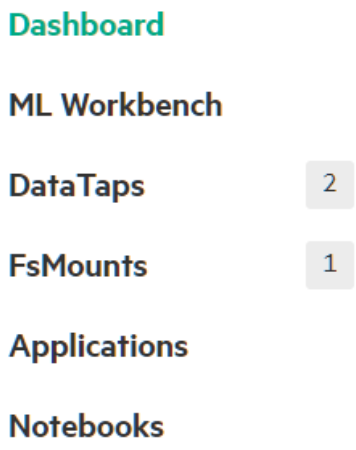


Dashboard	Opens the Kubernetes **Dashboard** screen. See Dashboard - Kubernetes Tenant Member on page 357
DataTaps	Opens the **DataTaps** screen, which enables you to upload and download files.
FS Mounts	Opens the **FS Mounts** screen, which enables you to upload and download files.
Applications	Opens the **Kubernetes Applications** screen, which enables you to launch applications within Kubernetes pods and access service endpoints and virtual endpoints.
Notebooks	Opens the **Notebooks** screen, from which you can launch notebook servers and view notebook endpoints.

Accessing the Kubeflow Dashboard

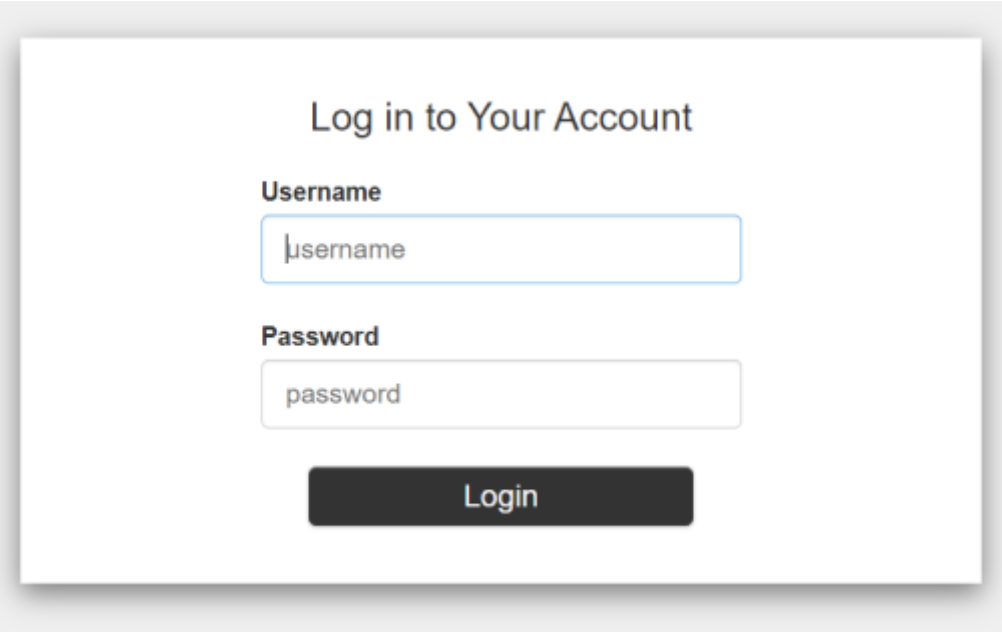
To access the Kubeflow dashboard:

1. Verify that the Kubernetes Administrator has installed Kubeflow, as described in Kubeflow Installation.
2. Log in to HPE Ezmeral Runtime Enterprise with your LDAP/AD credentials.
3. In the main menu, click **ML Workbench**. See Toolbar and Main Menu - Kubernetes Tenant Member.

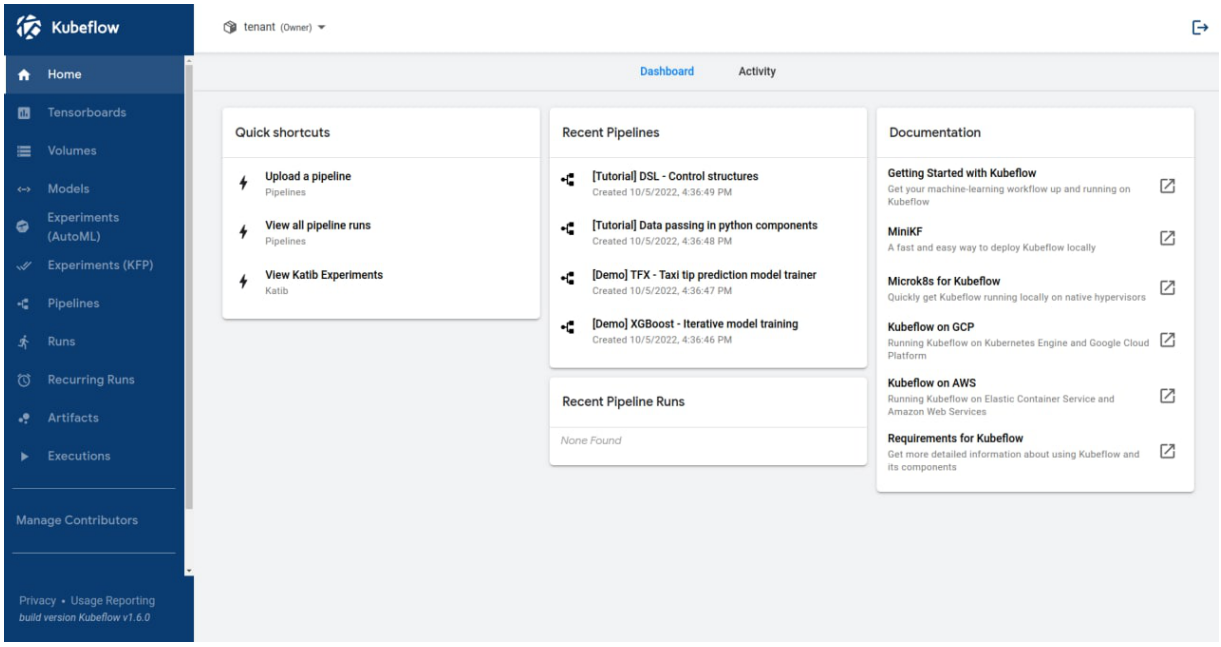


4. In the **Training and Workflow** block, click the **Kubeflow** link:

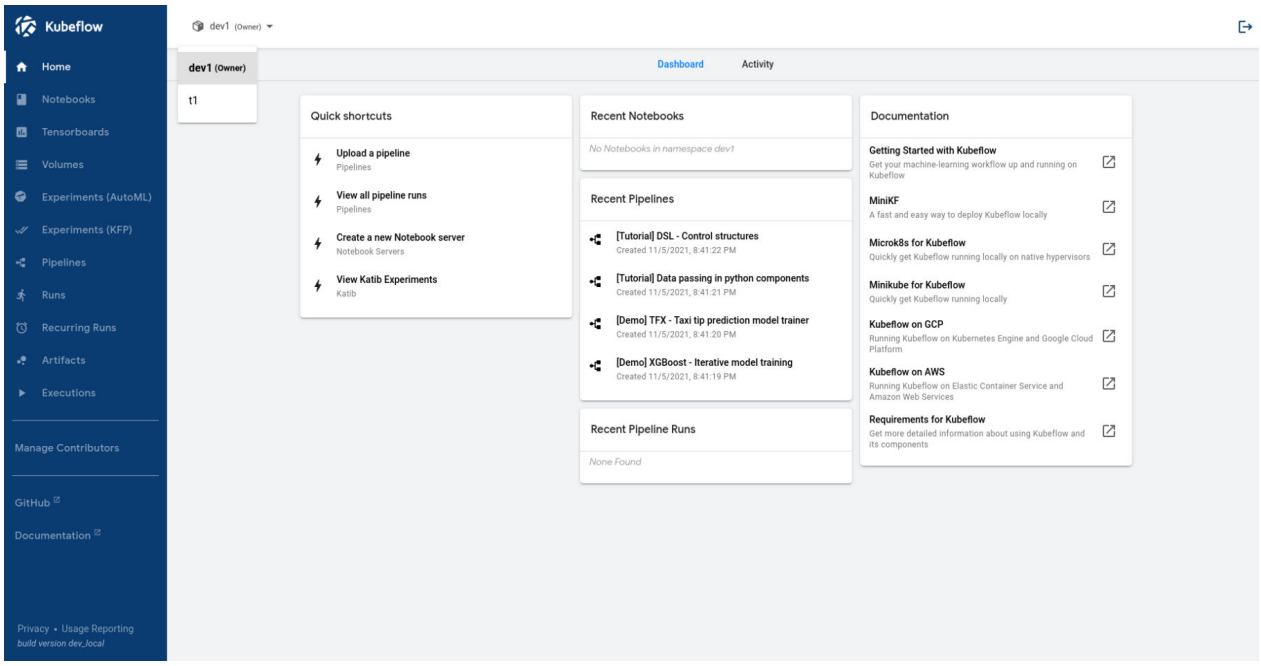
5. The Kubeflow dashboard **Login** screen appears in a new browser tab. Log in with your AD/LDAP credentials.



6. On first time login, HPE Ezmeral Runtime Enterprise prompts you to create a new profile namespace. Continue to get to the **Kubeflow dashboard** screen. From this screen, you can create Tensorboards and run Experiments, pipelines, and more.



You can switch to the tenant namespace by specifying the tenant's name in the **namespace dropdown** list:



Accessing the Airflow Dashboard

Prerequisites

Verify that the Airflow cluster is created. See:

- Creating an Airflow Cluster Automatically on page 517
- Creating an Airflow Cluster Manually on page 520

About this task

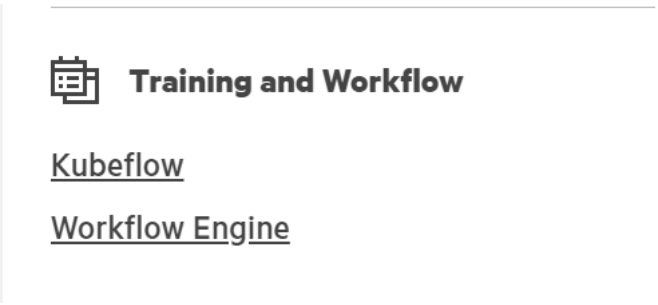
After installation, you can access the Airflow Dashboard.

Apache Airflow is a workflow automation and scheduling system that can be used to author and manage data pipelines. Airflow uses workflows made of directed acyclic graphs (DAGs) of tasks.

For more information, see the official Apache Airflow Documentation (link opens an external site in a new browser tab or window.

Procedure

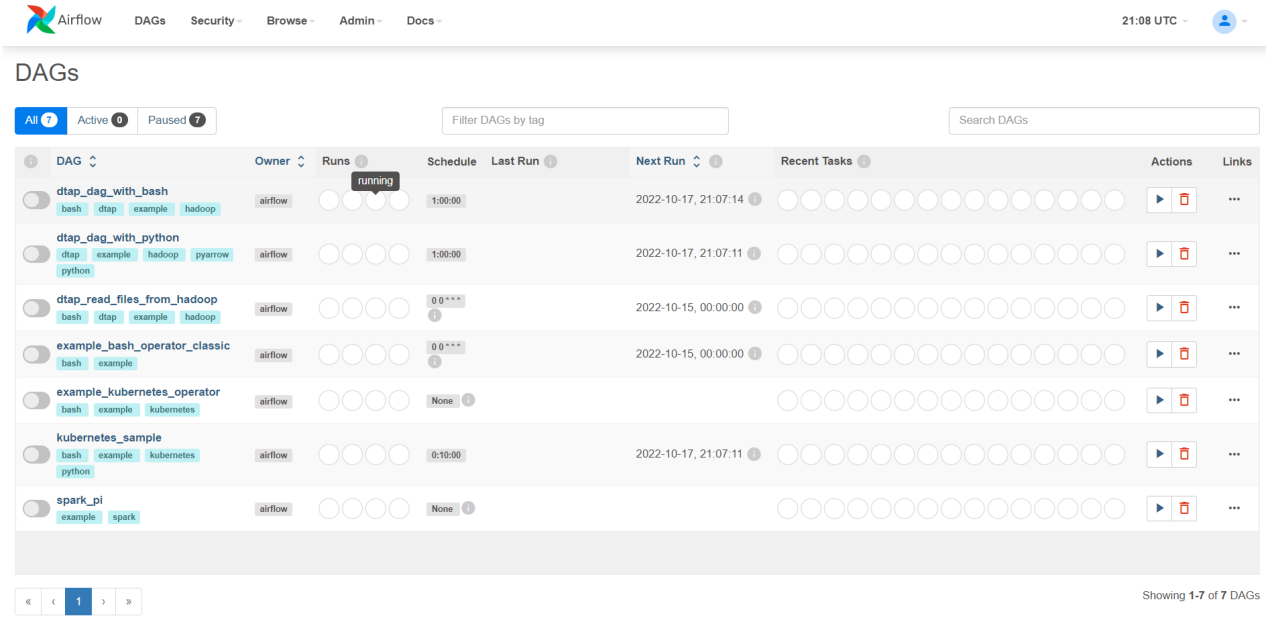
1. Log in to the HPE Ezmeral Runtime Enterprise control plane with your AD or LDAP credentials.
2. In the main menu, select **ML Workbench**.



   The HPE Ezmeral Runtime Enterprise new UI opens.

3. On the **Training and Workflow** panel, select **Workflow Engine**.
   The **Log in to Your Account** screen opens in a new browser tab.
4. Log in with your AD or LDAP credentials.
5. From the Airflow dashboard, you can access the components of your Airflow deployment.

Results



DataTaps

The topics in this section describe DataTaps on HPE Ezmeral Runtime Enterprise.

About DataTaps

DataTaps expand access to shared data by specifying a named path to a specified storage resource. Applications running within virtual clusters that can use the HDFS filesystem protocols can then access paths within that resource using that name, and DataTap implements Hadoop File System API. This allows

you to run jobs using your existing data systems without the need to make time-consuming copies or transfers of your data. Tenant/Project Administrator users can quickly and easily build, edit, and remove DataTaps using the **DataTaps** screen, as described in The DataTaps Screen (Admin). Tenant Member users can access DataTaps by name.

Each DataTap requires the following properties to be configured, depending on the type of storage being connected to (MapR, HDFS, HDFS with Kerberos, or NFS):

- **Name:** A unique name for each DataTap. This name may contain letters (A-Z or a-z), digits (0-9), and hyphens (-), but may not contain spaces. You can use the name of a valid DataTap to compose DataTap URIs that you pass to applications as arguments. Each such URI maps to some path on the storage system that the DataTap points to. The path indicated by a URI might or might not exist at the time you start a job, depending on what the application wants to do with that path. Sometimes the path must indicate a directory or file that already exists, because the application intends to use it as input. Sometimes, the path must not currently exist, because the application expects to create it. The semantics of these paths are entirely application- dependent, and are identical to their behavior when running the application on a physical Hadoop or Spark platform.
- **Description:** Brief description of the DataTap, such as the type of data or the purpose of the DataTap.
- **Type:** Type of file system used by the shared storage resource associated with the DataTap (**MAPR**, **HDFS**, or **NFS**). This is completely transparent to the end job or other process using the DataTap.

The following fields depend on the DataTap type:

- MapR
- HDFS
- NFS on page 365
- GCS on page 365

MapR

NOTE: All of the links to MapR articles in this section will open in a new browser tab/window.

A MapR DataTap is configured as follows:

- **Cluster Name:** Name of the MapR cluster. See the MapR articles Creating the Cluster and Creating a Volume articles.
- **CLDB Hosts:** DNS name or address of the container location database of a MapR cluster. See the MapR article Viewing CLDB Information.
- **Port:** Port for the namenode service on the host used to access the MapR file system. See the MapR article Specifying Ports.
- **Mount Path:** Complete path to the directory containing the data within the specified MapR file system. You can leave this field blank if you intend the DataTap to point at the root of the MapR cluster. See the MapR articles Viewing Volume Details and Creating a Volume.
- **MapR Secure:** Checking this check box if MapR cluster is secured. When the MapR cluster is secured, all network connections require authentication, and moving data is protected with wire-level encryption. MapR allows applying direct security protection for data as it comes into and out of the platform without requiring an external security manager server or a particular security plug-in for each ecosystem component. The security semantics are applied automatically on data being retrieved or stored by any ecosystem component, application, or users. See the MapR article Security.
- **Ticket Source:** Select the ticket source. This will be one of the following:

  - **Upload Ticket File:** This is enabled when Ticket source is selected as **Use Existing File**.
  - **Use the existing one:** To use the existing ticket details.
- **Ticket file:** This will be one of the following:
  - When **Upload Ticket File** is selected, **Browse** button is enabled to select the tiket file.
  - When **Use the Existing One** is selected, it is the name of the existing ticket file.
- **Enable Impersonation:** When you enable impersonation, when a user signs into the container and creates a file in the MapR cluster through the DataTap connection, ownership of that file is assigned to that user. If the user does not exist in the MapR cluster, then the connection between the DataTap and the MapR cluster is rejected. Typically, administrators ensure that the same users exist in both the container and the MapR cluster by configuring both the container and the MapR cluster with the same AD/LDAP settings.
- **Select Ticket Type:** Select the ticket type. This will be one of the following:
  - **User:** Grants access to individual users with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service**: Accesses services running on client nodes with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service (with impersonation):** Accesses services running on client nodes to run jobs on behalf of any user. The ticket cannot be used to impersonate the `root` or `mapr` users.
  - **Tenant:** Allows tenant users to access tenant volumes in a multi-tenant environment. The ticket can impersonate any user.
- **Ticket User:** Username to be included in the ticket for authentication.
- **MapR Tenant Volume:** Indicates whether or not the mount path is a MapR tenant volume. See the MapR article Setting Up a Tenant.
- **Enable Passthrough**: Select this box to enable Passthrough mode.

See the following examples for additional information:

- Sample MAPR DataTap - No Impersonation
- Sample MAPR DataTap - Impersonation

HDFS

An HDFS DataTap is configured as follows:

- **Host:** DNS name or IP address of the server providing access to the storage resource. For example, this could be the host running the namenode service of an HDFS cluster.
- **Standby NameNode:** DNS name or IP address of a standby namenode host that an HDFS DataTap will try to reach if it cannot contact the primary host. This field is optional; when used, it provides high-availability access to the specified HFDS DataTap.
- **Port:** For HDFS DataTaps, this is the port for the namenode server on the host used to access the HDFS file system.
- **Path:** Complete path to the directory containing the data within the specified HDFS file system. You can leave this field blank if you intend the DataTap to point at the root of the specified file system.

- **Kerberos parameters:** If the HDFS DataTap has Kerberos enabled, then you will need to specify additional parameters. HPE Ezmeral Runtime Enterprise supports two modes of user access/ authentication.
  - Proxy mode permits a “proxy user” to be configured to have access to the remote HDFS cluster. Individual users are granted access to the remote HDFS cluster by the proxy user configuration. Mixing and matching distributions is permitted between the compute Hadoop cluster and the remote HDFS.
  - Passthrough mode passes the credentials of the current user to the remote HDFS cluster for authentication.
- HDFS file systems configured with TDE encryption as well as cross-realm Kerberos authentication are supported. See HDFS DataTap TDE Configuration and HDFS DataTap Cross-Realm Kerberos Authentication for additional configuration instructions.

NFS

NOTE: This option is not available for Kubernetes tenants.

An NFS DataTap is configured as follows:

- **Host:** DNS name or IP address of the server providing access to the storage resource.
- **Share:**This is the exported share on the selected host.
- **Path:** Complete path to the directory containing the data within the specified NFS share. You can leave this field blank if you intend the DataTap to point at the root of the specified share.

GCS

An GCS DataTap is configured as follows:

- **Bucket Name:** Specify the bucket name for GCS.
- **Credential File Source:** This will be one of the following:
  - When **Upload Ticket File:** is selected, **Browse** button is enabled to select in the **Credential File**. The credential file is a JSON file that contains the service account key.
  - When **Use the Existing One:** is selected, enter the name of the previously uploaded credential file. The credetial file is a JSON file that contains the service account key.
- **Proxy:** This is optional. Specify http proxy to access GCS.
- **Mount Path:**Enter a path within the bucket that will serve as the starting pointfor the DataTap. If the path is not specified, the starting point will default to the bucket.

Using a DataTap

The storage pointed to by a DataTap can be accessed via a URI that includes the name of the DataTap.

A DataTap points to the top of the “path” configured for the given DataTap. The URI has the following form:

```
dtap://datatap_name/
```

In this example, `datatap_name` is the name of the DataTap that you wish to use. You can access files and directories further in the hierarchy by appending path components to the URI:

```
dtap://datatap_name/some_subdirectory/another_subdirectory/some_file
```

For example, the URI `dtap://mydatatapr/home/mydirectory` means that the data is located within the `/home/mydirectory` directory in the storage that the DataTap named `mydatatap` points to.

DataTaps exist on a per-tenant basis. This means that a DataTap created for Tenant A cannot be used by Tenant B. You may, however, create a DataTap for Tenant B with the exact same properties as its counterpart for Tenant A, thus allowing both tenants to access the same storage resource. Further, multiple jobs within a tenant may use a given DataTap simultaneously. While such sharing can be useful, be aware that the same cautions and restrictions apply to these use cases as for other types of shared storage: multiple jobs modifying files at the same location may lead to file access errors and/or unexpected job results.

Users who have a Tenant Administrator role can view and modify detailed DataTap information. Members can only view general DataTap information and are unable to create, edit, or remove a DataTap.

CAUTION: Data conflicts can occur if more than one DataTap points to a location being used by multiple jobs at once.

CAUTION: Editing or deleting a DataTap while it is being used by one or more running jobs can cause errors in the affected jobs.

More information

Troubleshooting DataTap Issues on page 944

The DataTaps Screen

Selecting **DataTaps** in the main menu opens the **DataTaps** screen. The information and functions on this screen will vary depending on your role. For Members, the **DataTaps** screen appears as shown in the following image.

This screen displays the following information and is read-only; you cannot edit any of these parameters:

- **Name:** Name of the DataTap. Clicking a name in this column opens the **DataTap Browser** screen for the selected DataTap. See The DataTap Browser Screen.
- **Host:** Host where the DataTap is located.
- **Path**: Path to the DataTap on the host.
- **Details:** This section only appears if you have the Tenant Administrator role for the tenant that contains the selected DataTap. This section contains a table that presents the following detailed information about the selected DataTap:
  - **Type:** Type of DataTap (**MAPR**, **HDFS**, or **NFS**).
  - **Authentication (Kerberos) Details:** This column appears if Kerberos protection is enabled for the current DataTap. See HDFS DataTap Kerberos Security.
    - **Host:** IP address(es) of the Kerberos host(s) and port.
    - **Access Method:** How the DataTap access the storage resource. This will be either **Proxy** or **Passthrough**.
    - **Keytab File:** Kerberos keytab file.

    - **Client Principal:** If the DataTap uses proxy access, this lists the client principal whose credentials grant access to the storage resource.
    - **Service ID:** ID of the service providing the DataTap (such as HPE Ezmeral Data Fabric).
    - **Realm:** Kerberos realm.
  - Whether (**True**) or not (**False**) the DataTap is read-only.
- **Status:** Status of the DataTap.

The DataTap Browser Screen

In the **DataTaps** screen, clicking the name of a DataTap opens the **DataTap Browser** screen for the selected DataTap.

This screen contains the following information and functions:

- **File/Directory Buttons (1):** These buttons allow you to create and delete directories and files, upload files, and rename files and directories. See Uploading and Downloading Files.
- **DataTap URI (2):** This field provides the full path to the currently-selected directory or file.
- **Directory listing (3):** This list presents a hierarchical view of the directories and files that can be accessed by the selected DataTap. The **File/Directory** buttons are enabled or disabled depending on your selections in this listing.
  - Clicking an item in this list selects that item and makes additional functions available. See Uploading and Downloading Files.
  - Clicking an **Expand** icon (+) next to a collapsed directory expands that directory to reveal any subdirectories and/or files within that directory.
  - Clicking a **Collapse** icon (-) next to an expanded directory collapses that directory to hide any subdirectories and/or files within that directory.

Uploading and Downloading Files

The **Directory Listing** area of the **DataTap Browser** screen contains an expandable tree view of the directories underneath the root directory of the selected DataTap.

NOTE: Various **File/Directory** buttons will become available depending on your directory/file selection. This image shows all five buttons enabled for documentation purposes, but this will not happen during actual DataTap use.

In this view:

- Clicking a plus sign (**+**) next to a directory expands that directory to display the file(s) and sub-directories (if any) under the selected directory.
- Clicking a minus sign (**-**) next to a directory collapses the view of the file(s) and sub-directories (if any) under the selected directory.

When you are browsing locations within a locally-shared storage service created at deployment install time, the **File/Directory** buttons allow you to add, rename, and remove files and directories. For any other DataTap, the **DataTap Browser** screen will allow you to view the file/directory structure and select paths for various UI purposes. In this case, you will need to upload/download files and/or create/remove directories from outside the web interface using some native client appropriate for the storage service. For certain operations (like creating a directory), it may also be useful to access the DataTap from within a virtual node and then manually perform `hadoop fs` operations on it.

From left to right, the **File/Directory** buttons are:

- Selecting a directory and then clicking the **Create Directory** button (plus sign) opens the **Create new directory under /directory** window, where **/directory** is the name of the currently selected directory. Entering a name in the field and then clicking **OK** creates a new sub-directory and closes the window.

- Selecting a directory or file and then clicking the **Rename** button (pencil) opens the **Rename item** window, where **item** is the name of the currently selected directory or file. Entering a name in the field and then clicking **OK** renames the selected directory or file.

- Selecting a directory and then clicking the **Upload** button (up arrow) opens a standard **Upload** dialog, which allows you to locate, select, and upload a file to the selected directory. The dialog appearance will vary based on your OS and browser settings.

- Selecting a file and then clicking the **Download** button (down arrow) opens a standard **Save As** dialog, which allows you to save the selected file to a directory on either your local hard drive or any network storage that you have access to.

- Selecting a directory or file and then clicking the **Delete** button (trash can) deletes the selected directory or file. Deleting a directory also deletes all of the sub-directories and files within that directory, if any.

CAUTION: Do not rename or delete a directory or file that is in use, as this could cause job failures and other errors. there is no undo function when deleting a directory or file.

FS Mounts

The filesystem mount feature allows the automatic addition of NFS v3 or v4 volumes or mounts to virtual nodes/containers. This allows virtual nodes/containers to directly access NFS shares as if they were local directories. You can use this feature to provide common files across all of the virtual nodes/containers of a given tenant, such as a common configuration file that will be used by each of the virtual nodes/containers in the Marketing tenant. This eliminates the need to manually copy common files to individual virtual nodes/containers.

All virtual nodes/containers include a root directory called `/bd-fs-mnt`. If one or more filesystems have been mounted, then this directory will contain the mounted filesystems. Each mounted filesystem in this

directory will have the same name as the **Mount Name** that was assigned when creating the FS mount (see Creating a New FS Mount).

Filesystems are mounted on a per-tenant basis, meaning that a given filesystem mount will be applied to each of the virtual nodes/containers in the tenant where that filesystem was created. For example, if you create a filesystem mount in the Marketing tenant, then each of the virtual nodes/containers created in the Marketing tenant will include that filesystem mount. Tenant Administrator users can create, modify, and delete filesystem mounts. Tenant Member and Platform Administrator users may view filesystem mounts but cannot modify them.

A filesystem may be mounted as either:

- **Read Only:** Users can view (read) objects in the filesystem but cannot create, modify, or delete objects.
- **Read/Write:** Users can view, create, modify, and/or delete objects.

FSmount is backed by a POSIX-based filesystem, such as the HPE Ezmeral Data Fabric POSIX client or NFS server. When HPE Ezmeral Runtime Enterprise is configured with HPE Ezmeral Data Fabric storage as its tenant storage, then FSmount points to HPE Ezmeral Data Fabric POSIX clients by default.

Inside every container:

- When a new filesystem is mounted, the **Name** property will be populated in the `/bd-fs-mnt` directory.
- The contents of the NFS share will be accessible in either read only or read/write fashion, depending on the settings provided when creating the mount.
- Users will not be able to write files to or create new folders in `/bd-fs-mnt`.

See the following articles for additional information:

- The FS Mounts Screen
- Creating a New FS Mount
- Editing an Existing FS Mount
- Deleting an FS Mount

The FS Mounts Screen

NOTE: This feature is not available for imported Kubernetes clusters. See Importing an External Kubernetes Cluster for additional information.

Selecting **FS Mounts** in the main menu opens the **FS Mounts** screen. The information and functions on this screen will vary depending on your role:

- **Members & Platform Administrator:** The **FS Mounts** screen provides read-only information about filesystem mounts. See Member View.
- **Tenant/Project Administrators:** The **FS Mounts** screen provides information about filesystem mounts and allows you to create and delete filesystem mounts. See Tenant Administrator View.

Member View

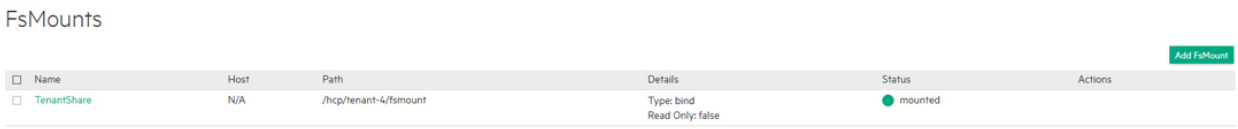
The **FS Mounts** screen for Members and Platform Administrators appears as shown in the following image.

This screen displays the following information and is read-only; you cannot edit any of these parameters:

- **Name:** Name of the filesystem mount. Clicking the name of a filesystem mount opens the **FS Mount Browser** screen. This screen functions identically to the **DataTap Browser** screen. See The DataTap Browser Screen.
- **Host:** Hostname or IP address of the file system host.
- **Path:** Path to the filesystem mount.
- **Details:** Type of filesystem mount and whether Read Only access is enabled (**true**) or disabled (**false**).
- **Status:** Status of the filesystem mount. This can be one of the following:
  - **Mounting:** The filesystem mount is being brought up on one or more of the virtual host(s) in the tenant.
  - **Mounted:** The filesystem mount has been successfully brought up on all of the virtual nodes/containers in the tenant.
  - **Altering:** The filesystem mount is being updated to reflect new settings.
  - **Errors:** The filesystem mount has failed to come up on one or more of the virtual nodes/containers in the tenant. Platform Administrator users can view errors from individual hosts by hovering the mouse cursor over the **Errors** status indicator.
  - **Unmounting:** The filesystem mount is being unmounted from the virtual nodes/containers in the tenant.

Tenant/Project Administrator View

The **FS Mounts** screen for Tenant/Project Administrators appears as shown in the following image.



This screen contains the following buttons:

- **Add FS Mount:** Clicking this button opens the **Add FS Mount** popup. See Creating a New FS Mount.

  NOTE: Filesystem mounts cannot be created for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

- **Delete:** Clicking this button deletes the selected filesystem mount(s) from the tenant. See Deleting an FS Mount.

The table on this screen contains the following information and functions:

- **Name:** Name of the filesystem mount. Clicking the name of a filesystem mount opens the **FS Mount Browser** screen. This screen functions identically to the **DataTap Browser** screen. See The DataTap Browser Screen.
- **Host:** Hostname or IP address of the file system host.
- **Path:** Path to the filesystem mount.
- **Details:** Type of filesystem mount and whether Read Only access is enabled (**true**) or disabled (**false**).

- **Status:** Status of the filesystem mount. This can be one of the following:
  - **Mounting:** The filesystem mount is being brought up on one or more of the virtual host(s) in the tenant.
  - **Mounted:** The filesystem mount has been successfully brought up on all of the virtual nodes in the tenant.
  - **Errors:** The filesystem mount has failed to come up on one or more of the virtual node(s) in the tenant.
  - **Unmounting:** The filesystem mount is being unmounted from the virtual node(s) in the tenant.
- **Actions:** The following actions are available for each filesystem mount, except the default **TenantStorage** mount:
  - **Edit:** Clicking the **Edit** icon (pencil) in the **Actions** column opens the **Edit FS Mount** popup. Editing a filesystem mount that is in use by a running cluster may cause file access errors within that cluster. See Editing an Existing FS Mount.
  - **Delete:** Clicking the **Delete** icon (trash can) in the **Actions** column deletes the filesystem mount from the tenant. See Deleting an FS Mount.

Creating a New FS Mount

NOTE: Filesystem mounts cannot be created for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

Clicking the **Add FS Mounts** button in the **FS Mounts** screen opens the **Add FS Mount** popup.

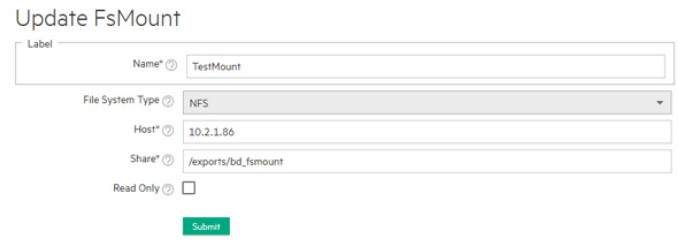
To create a new filesystem mount:

1. Enter the following information in the appropriate fields:
   - **Name:** Name of the filesystem mount. This name must contain only letters, numbers, and/or dashes and must be longer than two characters.
   - **Host:** Enter either the hostname or IP address of the file system host in the **Host** field.
   - **Share:** Enter the name of the path to the NFS export on the NFS server in the **Share** field.
   - **Read Only:** Check this check box to allow the virtual nodes in the tenant to be able to access objects in the filesystem mount but be unable to add, modify, or delete those objects. Clearing this check box allows the virtual nodes to add, modify, and remove objects in the filesystem mount.
2. Click **Submit** to finish creating the filesystem mount, or **Cancel** to exit without creating the filesystem mount.

Editing an Existing FS Mount

NOTE: Filesystem mounts cannot be created for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

In the **FS Mounts** screen, clicking the **Edit** icon (pencil) in the **Actions** column of the table opens the **Update FS Mount** screen for the selected filesystem mount.



To edit a filesystem mount, you may modify some or all of the following:

- **Name:** Name of the filesystem mount. This name must contain only letters, numbers, and/or dashes and must be longer than two characters.
- **Host:** Hostname or IP address of the file system host.
- **Share:** Name of the share.
- **Read Only:** Check this check box to allow the virtual nodes in the tenant to be able to access objects in the filesystem mount but be unable to add, modify, or delete those objects. Clearing this check box allows the virtual nodes to add, modify, and remove objects in the filesystem mount.

When you have finished modifying the parameters for the filesystem mount, click **Submit** to modify that filesystem mount.

Deleting an FS Mount

Tenant Administrators have the ability to delete filesystem mounts. To delete one or more filesystem mount(s):

1. Open the **FS Mounts** screen.
2. Either:
   - Select one or more filesystem mount(s) by checking the appropriate check box(es) in the table, and then click the **Delete** button.
   - Click the **Delete** icon (trash can) for a specific filesystem mount.
3. A popup warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without deleting the filesystem mount.

CAUTION: DELETING A FILESYSTEM MOUNT THAT IS BEING USED BY ONE OR MORE VIRTUAL NODE(S) MAY CAUSE FILE ACCESS ERRORS WITHIN THE AFFECTED VIRTUAL NODE(S).

NOTE: Deleting a filesystem mount does not affect your data. If you accidentally delete a filesystem mount, simply create a new one that points to the same location with the same name.

General Kubernetes Tutorials

This section contains general Kubernetes tutorials and examples.

For Kubeflow tutorials, see Kubeflow Tutorials on page 218.

Kubernetes Cluster Usage Examples

This article presents the following three sample Kubernetes cluster usage examples:

- Hello World

- WordPress with a Persistent Volume

Example 1: Hello World

This example is based on the Hello-World sample scenario, available here (link opens an external web site in a new browser tab/window).

Begin by creating the hello-world service manifest YAML file with HPE Ezmeral Runtime Enterprise annotation.

```
# kubectl apply -f https://k8s.io/examples/service/access/
hello-application.yaml
deployment.apps/hello-world created
# kubectl get deployments hello-world
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-world   2/2     2            2           36s
```

The contents of `cr-hello-world-app-service-epic-lb.yaml` are:

```
apiVersion: v1
kind: Service
metadata:
  name: hello-world-service-epic-lb
  labels:
    hpecp.hpe.com/hpecp-internal-gateway: "true"
spec:
  selector:
    run: load-balancer-example
  ports:
  - name: http-hello
    protocol: TCP
    port: 8080
    targetPort: 8080
  type: NodePort
```

NOTE: The `label` generates a service port on the Gateway host.

```
# kubectl create -f ./cr-hello-world-app-service-epic-lb.yaml
service/hello-world-service-epic-lb created
# kubectl describe services
Name:                     hello-world-service-epic-lb
Namespace:                default
Labels:                   hpecp.hpe.com/hpecp-internal-gateway: true
Annotations:              hpecp-internal-gateway/8080:
mip.storage.enterprise.net:10003  - Note the Gateway host IP address.
Selector:                 run=load-balancer-example
Type:                     NodePort
IP:                       10.96.60.29
Port:                     http-hello  8080/TCP
TargetPort:               8080/TCP
NodePort:                 http  31996/TCP
Endpoints:                10.244.1.5:8080,10.244.2.4:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
# curl http://mip.storage.enterprise.net:10003
Hello Kubernetes!
```

NOTE: If the web interface is configured for SSL access, then replace the `http://` in the cURL command above with `https://` .

If you cannot perform the mapping and receive Error 409 when executing the command `kubectl -n <namespace> logs kubedirector-<port_number>`, be sure that HPE Ezmeral Runtime Enterprise is not in Lockdown mode. See Lockdown Mode on page 916.

Example 2: PHP Guestbook Application with Redis

The following example is based on the PHP Guestbook sample scenario described here (link opens an external web site in a new browser tab/window).

Begin by launching the Redis services.

```
# kubectl apply -f https://kubernetes.io/examples/application/guestbook/
redis-master-deployment.yaml
deployment.apps/redis-master created
# kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
redis-master-7db7f6579f-s5llz   1/1     Running   0          79s
# kubectl logs -f -c master redis-master-7db7f6579f-s5llz
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 2.8.19 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in stand alone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 1
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'
[1] 28 Nov 03:08:51.748 # Server started, Redis version 2.8.19
[1] 28 Nov 03:08:51.749 # WARNING: The TCP backlog setting of 511 cannot be
enforced because /proc/sys/net/core/somaxconn is set to the lower value of
128.
[1] 28 Nov 03:08:51.749 * The server is now ready to accept connections on
port 6379
<CTRL-C>
# kubectl apply -f https://kubernetes.io/examples/application/guestbook/
redis-master-service.yaml
service/redis-master created
# kubectl get service
NAME                          TYPE        CLUSTER-IP     EXTERNAL-IP
PORT(S)          AGE
kubernetes                    ClusterIP   10.96.0.1      <none>
443/TCP          5h21m
redis-master                  ClusterIP   10.96.79.194   <none>
6379/TCP         41s
# kubectl apply -f https://kubernetes.io/examples/application/guestbook/
redis-slave-deployment.yaml
deployment.apps/redis-slave created
# kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
redis-master-545d695785-w2827   1/1     Running   0          12m
redis-slave-546fc99d45-5ffm2    1/1     Running   0          29s
redis-slave-546fc99d45-766rt    1/1     Running   0          29s
# kubectl apply -f https://kubernetes.io/examples/application/guestbook/
redis-slave-service.yaml
service/redis-slave created
```

```
# kubectl get services
NAME                          TYPE        CLUSTER-IP     EXTERNAL-IP
PORT(S)          AGE
kubernetes                    ClusterIP   10.96.0.1      <none>
443/TCP          5h26m
redis-master                  ClusterIP   10.96.79.194   <none>
6379/TCP         5m16s
redis-slave                   ClusterIP   10.96.55.128   <none>
6379/TCP         42s
```

Next, set up the Guestbook front-end service.

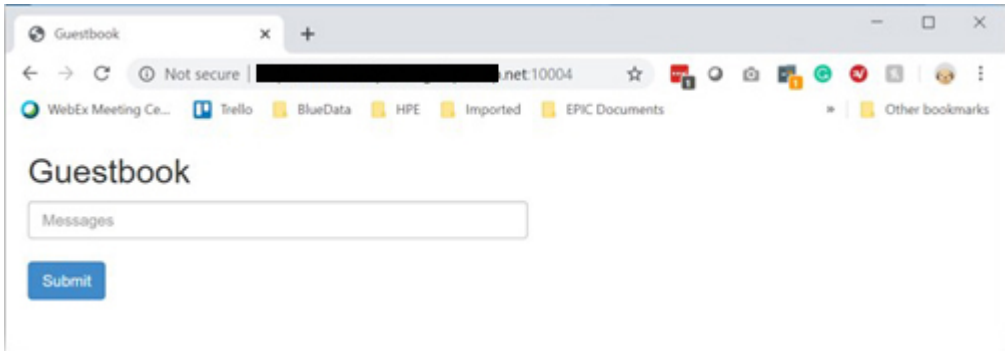
```
# kubectl apply -f https://kubernetes.io/examples/application/guestbook/
frontend-deployment.yaml
deployment.apps/frontend created
# kubectl get pods
NAME                            READY   STATUS              RESTARTS   AGE
frontend-678d98b8f7-754zv       0/1     ContainerCreating   0          40s
frontend-678d98b8f7-g5jtf       0/1     ContainerCreating   0          40s
frontend-678d98b8f7-l6xw9       0/1     ContainerCreating   0          40s
redis-master-545d695785-w2827   1/1     Running             0          18m
redis-slave-546fc99d45-5ffm2    1/1     Running             0          6m6s
redis-slave-546fc99d45-766rt    1/1     Running             0          6m6s
# kubectl get pods -l app=guestbook -l tier=frontend
NAME                         READY   STATUS    RESTARTS   AGE
frontend-678d98b8f7-754zv    1/1     Running   0          2m47s
frontend-678d98b8f7-g5jtf    1/1     Running   0          2m47s
frontend-678d98b8f7-l6xw9    1/1     Running   0          2m47s
# kubectl apply -f https://kubernetes.io/examples/application/guestbook/
frontend-service.yaml
service/frontend created
# kubectl get services
NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP
PORT(S)          AGE
frontend                      NodePort    10.96.165.194   <none>
80:31809/TCP     2m44s
kubernetes                    ClusterIP   10.96.0.1       <none>
443/TCP          5h36m
redis-master                  ClusterIP   10.96.79.194    <none>
6379/TCP         15m
redis-slave                   ClusterIP   10.96.55.128    <none>
6379/TCP         10m
```

Label the service so that the front-end NodePort service will be exposed via the Gateway host. This step is not necessary if the service was created in the namespace of a tenant that has the **Map Services To Gateway** option enabled. See Creating a New Kubernetes Tenant or Project and Editing an Existing Kubernetes Tenant or Project.

```
# kubectl label svc frontend hpecp.hpe.com/hpecp-internal-gateway=true
service/frontend labeled
# kubectl describe services frontend
Name:                     frontend
Namespace:                default
Labels:                   app=guestbook
  hpecp.hpe.com/hpecp-internal-gateway=true
  tier=frontend
Annotations:              hpecp-internal-gateway/80:
mip.storage.enterprise.net:10004 - Note the URL.
Selector:                 app=guestbook,tier=frontend
Type:                     NodePort
IP:                       10.96.165.194
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
```

```
NodePort:                  <unset>  31809/TCP
Endpoints:                 10.244.1.6:80,10.244.1.7:80,10.244.2.7:80
Session Affinity:          None
External Traffic Policy:   Cluster
Events:
  Type    Reason   Age   From          Message
  ----    ------   ----  ----          -------
  Normal  Service  38s   kubedirector  Created HPECP K8S service
```

Finally, the connection to the service using your browser. In this case, the port does not have an "http-" name prefix and the Gateway host is not doing SSL termination. You can therefore navigate to `http://<url_described_above>`.



Example 3: WordPress with Persistent Volume

The following example is based on the WordPress and MySQL with Persistent Volume described here (link opens an external web site in a new browser tab/window).

MySQL and WordPress each require a Persistent Volume to store data. Their Persistent Volume Claims will be created at the deployment step. HPE Ezmeral Data Fabric is used as the default persistent volume.

Begin by adding a Secret generator in `kustomization.yaml` by executing the following command, being sure to replace `YOUR_PASSWORD` with the password you want to use.

```
# mkdir wordpress
# cd wordpress
#
secretGenerator:
  - name: mysql-pass
      literals:
  - password=YOUR_PASSWORD
EOF
```

Next, use either of the following methods to download the following two YAML manifest files for the MySQL and WordPress services, respectively (links open an external website in a new browser tab/window):

- https://kubernetes.io/examples/application/wordpress/mysql-deployment.yaml
- https://kubernetes.io/examples/application/wordpress/wordpress-deployment.yaml

```
# curl -kO https://kubernetes.io/examples/application/wordpress/
mysql-deployment.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1238  100  1238    0     0   1430      0 --:--:-- --:--:-- --:--:--  1429
# curl -kO https://kubernetes.io/examples/application/wordpress/
wordpress-deployment.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
```

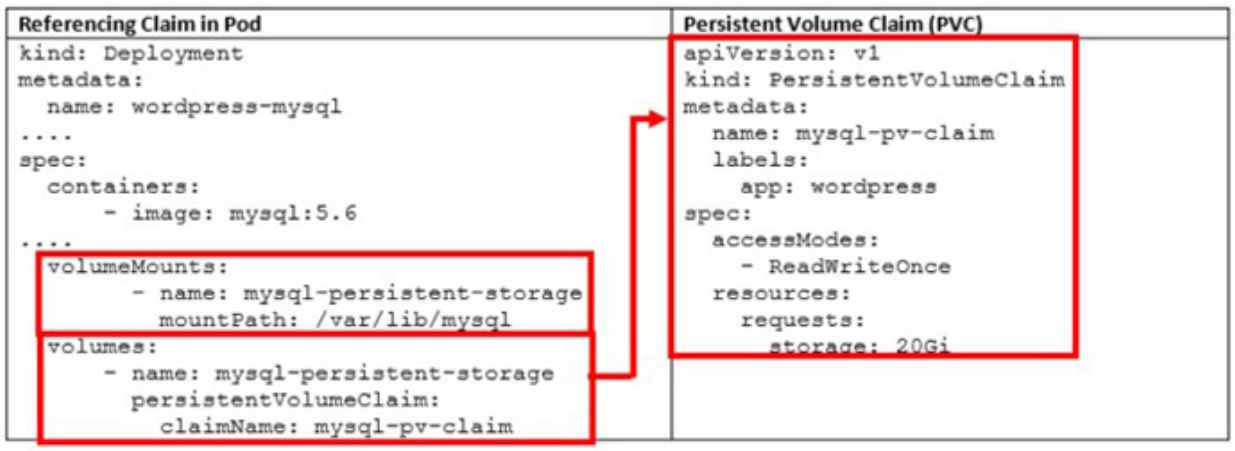
```
                                 Dload  Upload   Total   Spent    Left  Speed
100  1323  100  1323    0     0   1441      0 --:--:-- --:--:-- --:--:--  1441
# ls -al
total 9
drwxr-xr-x    1 leedavid UsersGrp         0 Nov 28 16:50 .
drwx------    1 leedavid UsersGrp         0 Nov 28 16:46 ..
-rw-r--r--    1 leedavid UsersGrp       137 Nov 28 16:49 kustomization.yaml
-rw-r--r--    1 leedavid UsersGrp      1238 Nov 28 16:47 mysql-deployment.yaml
-rw-r--r--    1 leedavid UsersGrp      1323 Nov 28 16:50
wordpress-deployment.yaml
```

If you installed HPE Ezmeral Runtime Enterprise with tenant storage, then HPE Ezmeral Data Fabric will already be registered as the default Storage Class in this namespace.

```
# kubectl get StorageClass
NAME                PROVISIONER        AGE
default (default)   com.mapr.csi-kdf   39h
# kubectl describe StorageClass
Name:                   default
IsDefaultClass:         Yes
Annotations:            storageclass.kubernetes.io/is-default-class=true
Provisioner:            com.mapr.csi-kdf
Parameters:
cldbHosts=192.168.20.131:7222,cluster=epic.mapr.cluster,csi.storage.k8s.io/
provisioner-secret-name=mapr-user-secret,csi.storage.k8s.io/
provisioner-secret-namespace=mapr-csi,csiNodePublishSecretName=mapr-ticket-sec
ret,csiNodePublishSecretNamespace=mapr-csi,mountPrefix=/
mapr-csi,namePrefix=k8s-1-,platinum=true,restServers=192.168.20.131:8443,secur
ityType=secure
AllowVolumeExpansion:  <unset>
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     Immediate
Events:                <none>
```

In these two manifest files, both the WordPress service and MySQL are requesting a persistent volume (PV):

- MySQL Deployment:



- WordPress Deployment:

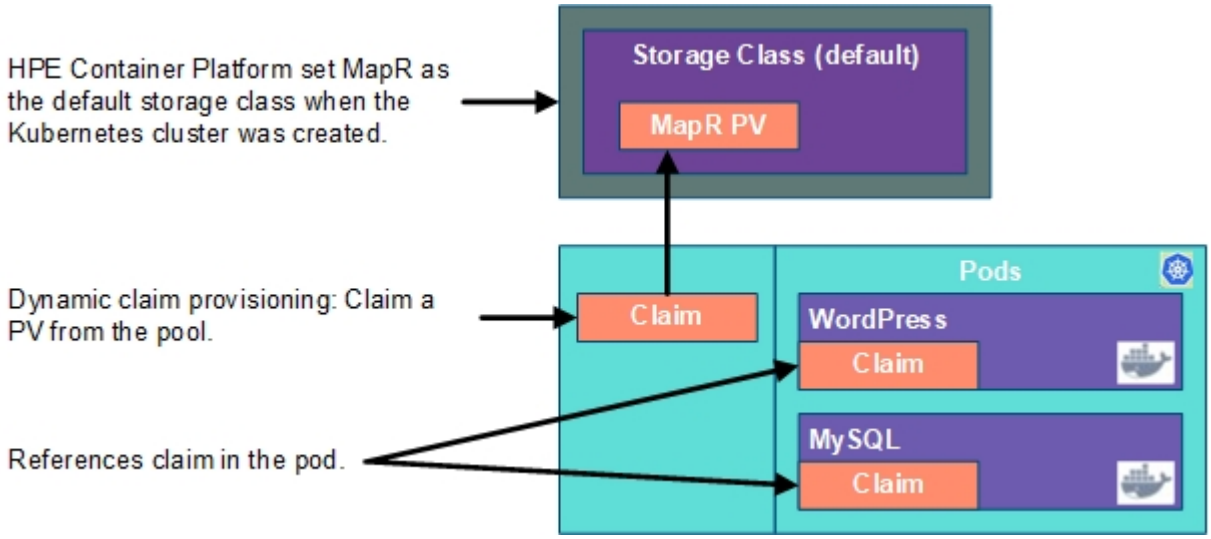
Referencing Claim in Pod

```
kind: Deployment
metadata:
  name: wordpress:4.8-apache
....
spec:
  containers:
      - image: mysql:5.6
....
  volumeMounts:
        - name: wordpress-persistent-storage
          mountPath: /var/www/html
  volumes:
      - name: wordpress-persistent-storage
        persistentVolumeClaim:
          claimName: wp-pv-claim
```

Persistent Volume Claim (PVC)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

Neither pod makes any explicit request for a specific storageClassName. Hence, they will use the default HPE Ezmeral Data Fabric StorageClass.



NodePort Service

Edit the WordPress manifest YAML to use the NodePort service instead of LoadBalancer service. This needs to be done for port mapping to occur.

```
# vi wordpress-deployment.yaml
# cat wordpress-deployment.yaml
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: NodePort - Ensure this is set to NodePort.
---
apiVersion: v1
kind: PersistentVolumeClaim
```

```
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: frontend
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: frontend
    spec:
      containers:
      - image: wordpress:4.8-apache
        name: wordpress
        env:
        - name: WORDPRESS_DB_HOST
          value: wordpress-mysql
        - name: WORDPRESS_DB_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-pass
              key: password
        ports:
        - containerPort: 80
          name: wordpress
        volumeMounts:
        - name: wordpress-persistent-storage
          mountPath: /var/www/html
      volumes:
      - name: wordpress-persistent-storage
        persistentVolumeClaim:
          claimName: wp-pv-claim
```

Continue by adding these two manifests to the `kustomization.yaml` file.

```
# cat <<EOF >>./kustomization.yaml
resources:
  - mysql-deployment.yaml
  - wordpress-deployment.yaml
EOF
```

The `kustomization.yaml` contains all of the resources for deploying a WordPress site and a MySQL database. You can apply the directory, and then verify both the HPE Ezmeral Data Fabric volumes and the services, as follows:

```
# kubectl apply --kustomize ./
secret/mysql-pass-9tt65k5fgm created
service/wordpress-mysql created
service/wordpress created
deployment.apps/wordpress-mysql created
deployment.apps/wordpress created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
```

Confirm that PVC is using the HPE Ezmeral Data Fabric StorageClass (see highlighted text below).

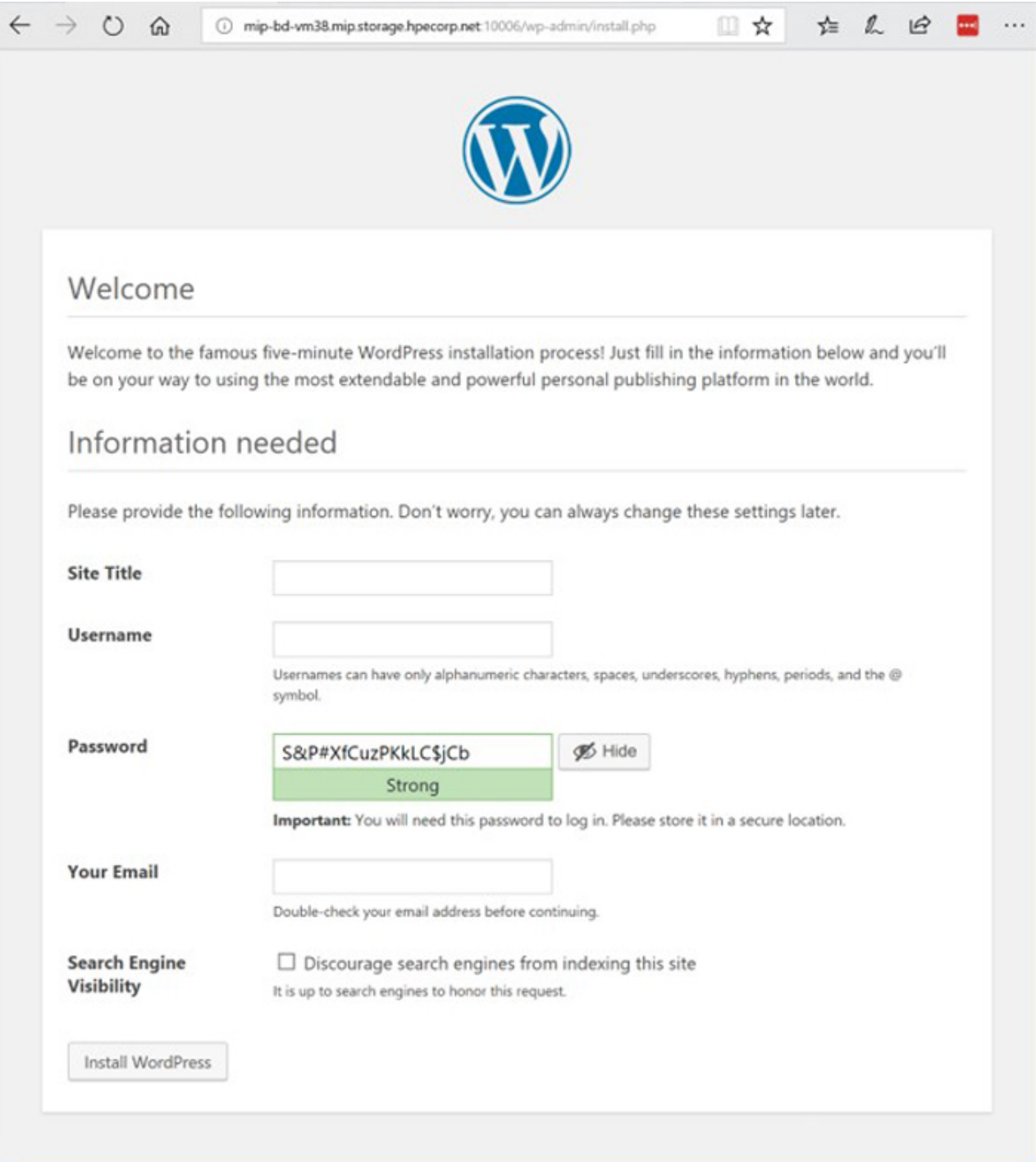
```
# kubectl get pvc
NAME             STATUS   VOLUME
CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mysql-pv-claim   Bound    mapr-pv-16f97a33-b8dd-488a-b6db-1d94a84286e2
20Gi       RWO            default        48s
wp-pv-claim      Bound    mapr-pv-896b3504-e9ba-4593-b9a0-88a9ece392b5
20Gi       RWO            default        48s
# kubectl get pv
NAME                                             CAPACITY  ACCESS MODES
RECLAIM POLICY  STATUS  CLAIM                       STORAGECLASS  REASON  AGE
mapr-pv-32850109-ef66-42db-9522-b563fbc01eae   10Gi      RWO
Delete          Bound   bdwebterm/pvc-kd-977sb-0  default               41h
mapr-pv-a24b1733-39db-40d2-bdaf-0be7c22ed83b   10Gi      RWO
Delete          Bound   bdwebterm/pvc-kd-nbwhn-0  default               31h
mapr-pv-dbf96aed-dafd-47b7-87d4-7d343f182d8b   20Gi      RWO
Delete          Bound   default/mysql-pv-claim    default               69s
mapr-pv-e3c1db71-2865-425c-971e-c01466e9d295   20Gi      RWO
Delete          Bound   default/wp-pv-claim       default               69s
mapr-pv-ed5f1be3-9be2-4470-83cf-67f9b31e9dbf   10Gi      RWO
Delete          Bound   bdwebterm/pvc-kd-dl26j-0  default               32h
```

Label the WordPress service so that the front-end NodePort service will be exposed via the Gateway host. This step is not necessary if the service was created in the namespace of a tenant that has the **Map Services To Gateway** option enabled. See Creating a New Kubernetes Tenant or Project and Editing an Existing Kubernetes Tenant or Project.

```
# kubectl get services
NAME                         TYPE        CLUSTER-IP     EXTERNAL-IP
PORT(S)          AGE
kubernetes                   ClusterIP   10.96.0.1      <none>
443/TCP          26h
wordpress                    NodePort    10.96.98.248   <none>
80:30996/TCP     24s
wordpress-mysql              ClusterIP   None           <none>
3306/TCP         24s
# kubectl label svc wordpress hpecp.hpe.com/hpecp-internal-gateway=true
service/wordpress labeled
# kubectl describe service wordpress
Name:                     wordpress
Namespace:                default
Labels:                   app=wordpress
  hpecp.hpe.com/hpecp-internal-gateway=true
Annotations:              hpecp-internal-gateway/80:
mip.storage.enterprise.net:10006
Selector:                 app=wordpress,tier=frontend
Type:                     NodePort
IP:                       10.96.98.248
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30996/TCP
Endpoints:                10.244.2.11:80
Session Affinity:         None
```

```
External Traffic Policy:  Cluster
Events:
  Type    Reason   Age   From          Message
  ----    ------   ----  ----          -------
  Normal  Service  26s   kubedirector  Created HPECP K8S service
```

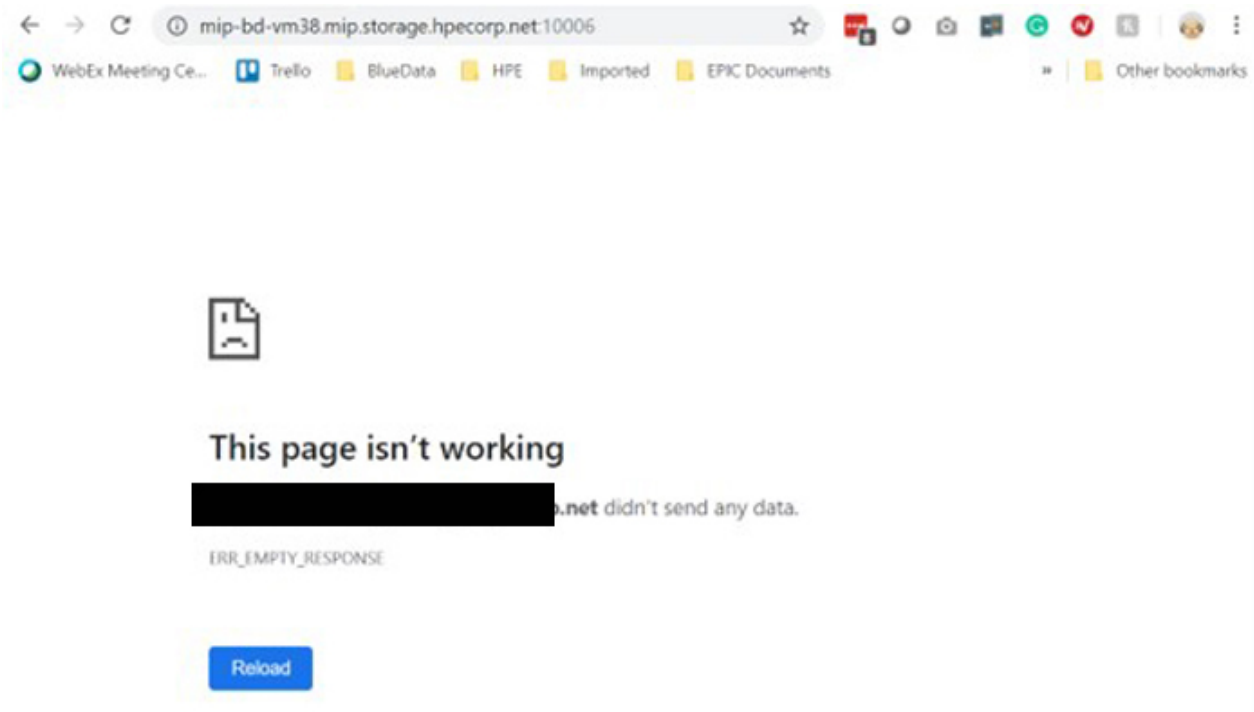
Copy the IP address and port number (see highlighted text above) to your browser. You should see set-up page similar to the following screenshot:



Destroy the application deployments (e.g. pods) and restart the deployments, making sure to preserve the WordPress application information and still preserved.

```
# kubectl delete deployment wordpress
deployment.extensions "wordpress" deleted
# kubectl delete deployment wordpress-mysql
deployment.extensions "wordpress-mysql" deleted
# kubectl get pods
No resources found.
# kubectl get deployments
No resources found.
```

The service is gone, as expected.

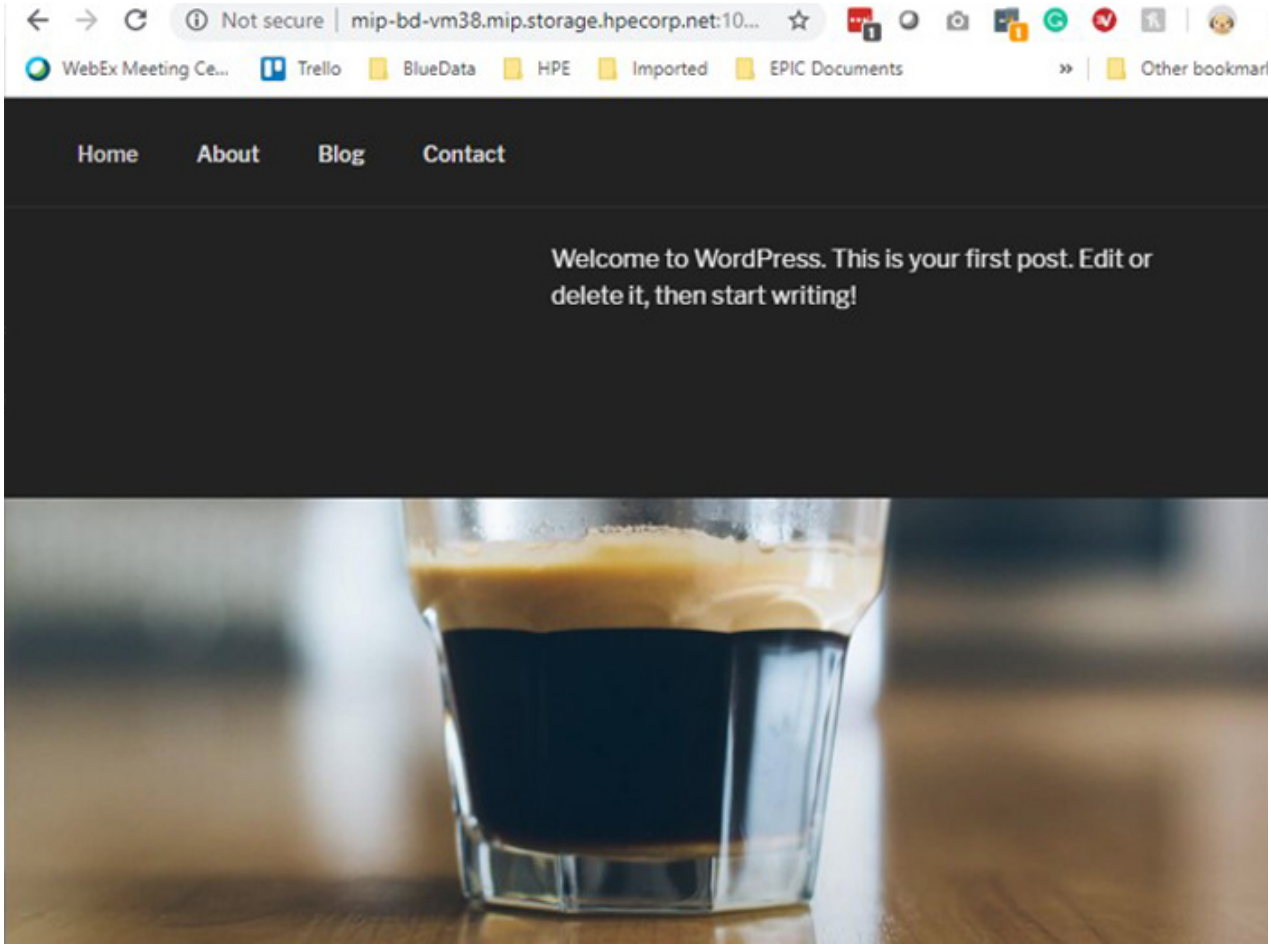


Reapply the same deployment, and reconnect to persistent storage.

```
# kubectl apply -k ./
secret/mysql-pass-9tt65k5fgm unchanged
service/wordpress-mysql unchanged
service/wordpress unchanged
deployment.apps/wordpress-mysql created
deployment.apps/wordpress created
persistentvolumeclaim/mysql-pv-claim unchanged
persistentvolumeclaim/wp-pv-claim unchanged
# kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
wordpress-594759d7f6-jdnvp        1/1     Running   0          27s
wordpress-mysql-847b7b996d-dwf6s  1/1     Running   0          28s
# kubectl describe service wordpress
Name:                     wordpress
Namespace:                default
Labels:                   app=wordpress
                          Hpecp.hpe.com/hpecp-internal-gateway=true
Annotations:              hpecp-internal-gateway/80:
mip.storage.enterprise.net:10006
Selector:                 app=wordpress,tier=frontend
Type:                     NodePort
IP:                       10.96.35.129
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31589/TCP
```

```
Endpoints:                10.244.1.18:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason   Age   From          Message
  ----    ------   ----  ----          -------
  Normal  Service  12m   kubedirector  Created HPECP K8S service
```

The WordPress service is restored.



Finally, you will need to delete the entire deployment in order to free up all of the resources, including the persistent storage.

```
# kubectl delete -k ./
secret "mysql-pass-9tt65k5fgm" deleted
service "wordpress-mysql" deleted
service "wordpress" deleted
deployment.apps "wordpress-mysql" deleted
deployment.apps "wordpress" deleted
persistentvolumeclaim "mysql-pv-claim" deleted
persistentvolumeclaim "wp-pv-claim" deleted
```

Sample YAML Reference Programs

A traditional YAML file has 4 main key-value pairs:

- `apiVersion` - Defines the API version of the `kind` used in the YAML file.
- `Kind` - Kind of Kubernetes object being created. Kubernetes supports many different types of objects or kinds, such as (but not limited to) `Pod`, `Service`, `Deployment`, and `Daemonset`.

- `Metadata` - Object metadata of the object, such as name and labels. Labels are identifiers that facilitate filtering or selecting the correct object from multiple similar objects.
- `Spec` - This key will have many things under it that are closely with the type of Kind/Object, such as (but not limited to) Containers, Volumes, NodePorts, or templates.

Sample 1: HTTPD as a Pod (Single YAML)

The following YAML script creates a Kubernetes pod object with a single container that uses CentOS with installed `httpd` package. This container also exposes a port that can be used from within the cluster to communicate with the container.

```
apiVersion: v1
kind: Pod
metadata:
  name: pod1
  labels:
    layers: single
    sample: httpd
spec:
  containers:
    - name: c1
      image: centos/httpd
      ports:
      - containerPort: 80
```

Project 2: HTTPS as a Service with NodePort to Expose the Endpoint

The following YAML script creates a Kubernetes NodePort service object, which forwards the container ports to the external network. NodePort objects do not include a container key; it links to a pod object based on the selector key. (All pods link to service objects via selector keys). This object includes the `ports` key and can have the `targetPort` and `nodePort` sub-keys.

- The `Port` field is required. This is the port where the service object is listening.
- The `targetPort` defaults to 80 unless specified otherwise. This is the service object output port.
- The `NodePort` defaults to a random port number greater than 30000 unless specified otherwise. This is the forwarded part of the key port.

```
apiVersion: v1
kind: Service
metadata:
  name: svc1
  labels:
    layers: single
    sample: svc-httpd
spec:
  type: NodePort
  selector:
    layers: single
    sample: httpd
  ports:
    - name: httpd
      port: 80
```

Project 3: HTTPD with NodePort and VolumeMount

This is a bigger YAML script that has two objects/kinds (`Pod` and `Service`) separated by `---`. The Pod object adds the following keywords:

- `volumes` - Used for a different type of storage facility. In this example, it mounts a directory on the host to a location on the container provided by the `volumeMounts` key.
- `volumeMounts` - Location where to mount the storage directory.

```
apiVersion: v1
kind: Pod
metadata:
  name: pod2
  labels:
    layers: single
    sample: httpd2
spec:
  containers:
  - name: c1
    image: centos/httpd
    ports:
    - containerPort: 80
    volumeMounts:
      - name: indexfile
        mountPath: /var/www/html
  volumes:
    - name: indexfile
      hostPath:
        path: /tmp
        type: DirectoryOrCreate

---
apiVersion: v1
kind: Service
metadata:
  name: svc2
  labels:
    layers: single
    sample: svc-httpd2
spec:
  type: NodePort
  selector:
    layers: single
    sample: httpd2
  ports:
    - name: httpd
      port: 80
```

Project 4: HTTPD with a PVC and FS Mount

This YAML file defines three kinds: `Pod`, `Service`, and `persistentVolumeClaim`. `persistentVolumeClaim` attaches a persistent volume to the container. If this is not defined, then a default persistent volume will be used. This is different than the volumes key used in the previous example, in that it used a `hostPath` driver while this example uses a `persistentVolumeClaim` object. The metadata defines the new label `hpecp.hpe.com/fsmount`. This is exclusive to an HPE Ezmeral Runtime Enterprise Kubernetes cluster where this label mounts a default FS Mount on the Pod.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
```

```
  name: pvc-sample
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi

---
apiVersion: v1
kind: Pod
metadata:
  name: pod3
  labels:
    layers: single
    sample: httpd3
    hpecp.hpe.com/fsmount: <tenant namespace name>
spec:
  containers:
  - name: c1
    image: bluedata/centos7
    ports:
    - containerPort: 80
    volumeMounts:
      - name: pvcloc
        mountPath: /mnt
  volumes:
    - name: pvcloc
      persistentVolumeClaim:
        claimName: pvc-sample

---
apiVersion: v1
kind: Service
metadata:
  name: svc3
  labels:
    layers: single
    sample: svc-httpd3
spec:
  type: NodePort
  selector:
    layers: single
    sample: httpd3
  ports:
    - name: httpd
      port: 80
```

Tutorial: Using Helm to Deploy Redis

Describes how to use the Helm package manager for Kubernetes to deploy Redis. Helm simplifies discovering and deploying services to a Kubernetes cluster.

This article contains the following sections:

Step 1: Install Helm

NOTE: If you are using the Kubernetes Web Terminal in HPE Ezmeral Runtime Enterprise, Helm is already installed. Skip to Step 2: Search For Chart on page 387.

Helm is a single binary that manages deploying Charts to Kubernetes (link opens an external website in a new browser tab/window). A chart is a packaged unit of Kubernetes software.

Execute the following commands to install Helm:

```
curl -LO https://storage.googleapis.com/kubernetes-helm/
helm-v2.8.2-linux-amd64.tar.gz
tar -xvf helm-v2.8.2-linux-amd64.tar.gz
mv linux-amd64/helm/usr/local/bin/
```

Once installed, initialize and then update the local cache to sync the latest available packages with the environment by executing the following commands:

```
helm init
helm repo update
```

Execute the following command to add the repo:

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
```

Step 2: Search For Chart

You can now start deploying software. You can use the search command. For example, you need to find a Redis chart in order to deploy Redis. You can search for and then inspect Redis by executing the following commands:

```
helm search repo redis
helm inspect stable/redis
```

Step 3: Deploy Redis

Execute the following command to deploy the chart to your Kubernetes cluster:

```
$ helm install my-release bitnami/redis
```

Helm launches the required pods. You can view all packages by executing the following command:

```
helm ls
```

If you receive an error that Helm could not find a ready Tiller pod, this means that Helm is still deploying. Wait a few moments for the Tiller container image to finish downloading.

Step 4: See Results

Helm deploys all the pods, replication controllers, and services. Use kubectl to display what was deployed:

```
kubectl get all
```

The pod remains in a `Pending` state until the container image is downloaded and a persistent volume is available.

```
kubectl apply -f pv .yaml
```

Enable write permissions to Redis by executing the following command:

```
chmod 777 - R /mnt/ data *
```

When the pod status changes to `Running`, the Redis cluster is now running on top of Kubernetes.

If desired, you can give a Helm chart a friendly name by executing a command such as:

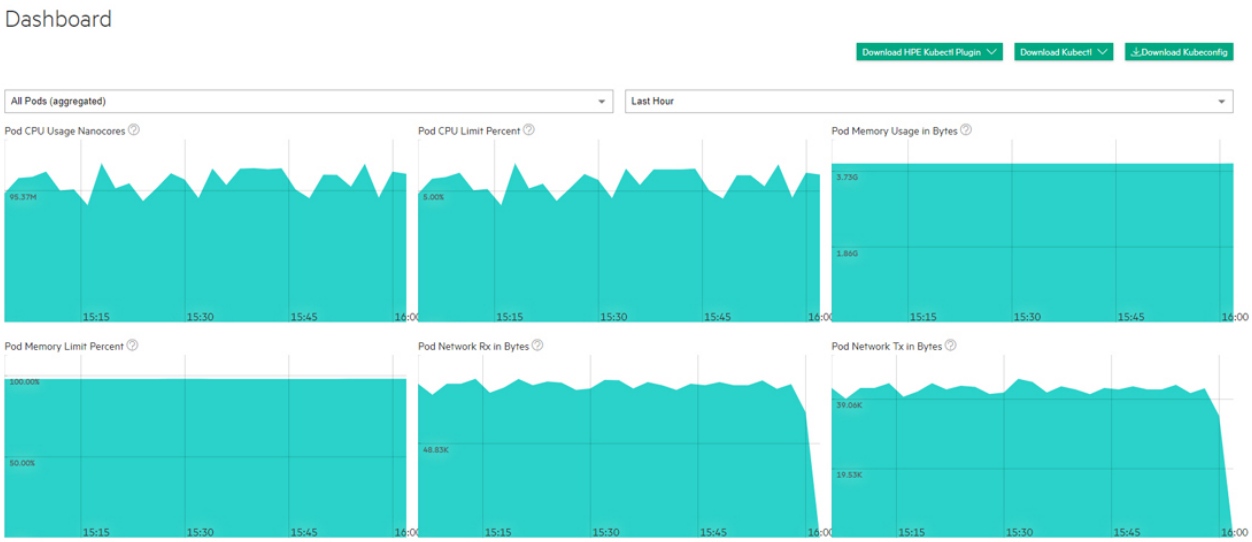
```
helm install --name my-release stable/redis
```

Tenant/Project Administration

The topics in this section describe information and tasks that Kubernetes Tenant/Project Administrators can perform in HPE Ezmeral Runtime Enterprise. .

Dashboard - Kubernetes Tenant/Project Administrator

HPE Ezmeral Runtime Enterprise users who are logged into a Kubernetes tenant/project with the Tenant/Project Administrator role can access the Kubernetes Tenant/Project Administrator **Dashboard** screen by selecting **Dashboard** in the main menu.



The top of this screen has three buttons that allows you to download the plugins that you need to access Kubernetes pods within a cluster. The buttons are:

- **Download HPE Kubectl Plugin:** Downloads the HPE `kubectl` plug-in. Please click here for more information on `kubectl` plug-ins (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl, below.

- **Download Kubectl:** Downloads the generic binary for the `kubectl` command line tool for controlling a Kubernetes cluster. Please click here for more information (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl, below, and Using the HPE Kubectl Plugin.

  NOTE: You may see a warning that `kubectl-hpecp` cannot be opened because the publisher cannot be verified. You may safely ignore this warning and proceed with the installation.

- **Download Kubeconfig:** Downloads the `kubeconfig` file that configures access to Kubernetes when used in conjunction with either the `kubectl` command line tool or other clients. Please click here for more information (link opens an external website in a new browser tab/window).

The top of this screen has two pull-down menus that allow you to filter the data by pod and time frame. You may also choose to view information for all applications or only for KubeDirector applications by moving the **Filter KubeDirector Applications** slider. Hovering your mouse over the graphs displays a popup with additional information. The following charts are available:

- **Pod CPU Use Nanocores:** Number of CPU nanocores in use.
- **Pod CPU Limit Percent:** Percentage of maximum number of pods that are currently running inside the current cluster.
- **Pod Memory Usage in Bytes:** Bytes of memory being used.
- **Pod Memory Limit Percent:** Percentage of memory limit being used.
- **Pod Network Rx in Bytes:** Bytes received over the network.
- **Pod Network Tx in Bytes:** Bytes transmitted over the network.
- **GPU Utilization (percent):** If GPUs are present, displays aggregate GPU utilization in percent.
- **GPU Memory Usage:** If GPUs are present, displays aggregate GPU memory usage in percent.

NOTE: Please see Downloading Kubernetes Usage Details for information about how to download detailed usage and uptime information in comma-delimited (.csv) format.

Installing Kubectl

To install Kubectl on your local system:

1. Download either of the Kubectl plugins:
   - If you are on a Windows system, then this download will be an .exe file.
   - If you are on a UNIX system, then you will need to execute one of the following commands:
     - **HPE Kubectl:** `chmod +x kubectl-hpecp`
     - **Generic Kubectl:**`chmod +x kubectl`
2. Place the Kubectl executable into a folder that is on your system's PATH.
3. Execute the command `kubectl hpecp refresh {HPE Ezmeral Runtime Enterprise controller/gateway ip address}`. If HTTPS is not enabled, then add the argument `--insecure=true`.

Toolbar & Main Menu - Tenant or Project Administrator

Describes the toolbar and navigation sidebar available to users with Kubernetes Tenant/Project Administrator access rights in HPE Ezmeral Runtime Enterprise.

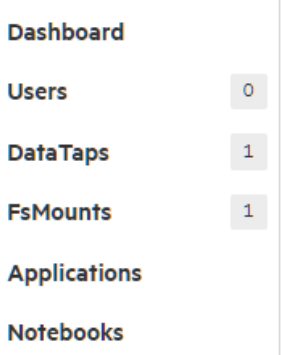
This article describes the UI items for Kubernetes Tenant Administrators and ML Ops Project Administrators:

Toolbar

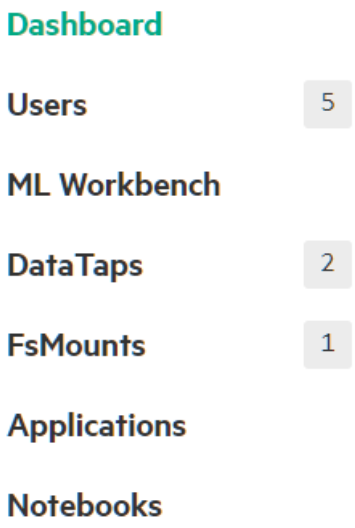
The layout of the Toolbar is the same as described in Navigating the GUI on page 143. For information about the content of the **Quick Access** menu for Tenant and Project Administrators, see Quick Access Menu -Tenant or Project Administrator on page 391.

Main Menu - Tenant or Project Administrator

The Kubernetes Tenant Administrator main menu for tenants that are not ML projects appears as shown in the following image:



The Kubernetes Tenant/Project Administrator main menu for tenants that are ML Ops projects is the same as the main menu for tenants that are not ML Ops Projects, except for the addition of the ML Workbench item, as shown in the following image:



Dashboard	Opens the Kubernetes **Dashboard** screen. See Dashboard - Kubernetes Tenant/Project Administrator on page 388.
Users	Opens the **Users** screen. Tenant Administrators and Project Administrators can assign HPE Ezmeral Runtime Enterprise users to a role in the tenant or project and can revoke the user access to the tenant or project.
ML Workbench	Opens the HPE Ezmeral Runtime Enterprise new UI in a separate browser tab or window, and displays the **Overview** tab of **Project Details** screen of of this project.
DataTaps	Opens the **DataTaps** screen, which enables users to upload and download files. Tenant Administrators and Project Administrators can view the connected storage service details, and can create, edit, and delete DataTaps.

FS Mounts	Opens the **FS Mounts** screen, which enables users to upload and download files. Tenant Administrators and Project Administrators can view the connected storage service details, and can create, edit, and delete FS Mounts.
Applications	Opens the **Kubernetes Applications** screen, which enables you to launch applications within Kubernetes pods and access service endpoints and virtual endpoints.
Notebooks	Opens the **Notebooks** screen, from which you can launch notebook servers and view notebook endpoints.

Quick Access Menu -Tenant or Project Administrator

For tenant or project administrators, the following items appear in the **Quick Access** menu:

Assign User	Opens the **Users Assignment** screen, which enables you to grant or revoke roles within this tenant or project to users.
User Info	Opens the **Current User Information** dialog, which lists your role, current project, and username.
User Guide	Opens this *User and Administrator Guide*.
Privacy	Opens the **Hewlett Packard Enterprise Privacy Statement** web page in a new browser tab or window.
Version	Displays version and build information about the HPE Ezmeral Runtime Enterprise deployment.
Ezmeral Runtime Enterprise New UI	Opens the home page of the HPE Ezmeral Runtime Enterprise new UI in a new browser tab or window. The interface that is displayed is the primary interface you use to access machine learning (ML Ops) projects, and tenants that use analytics applications such as Spark.

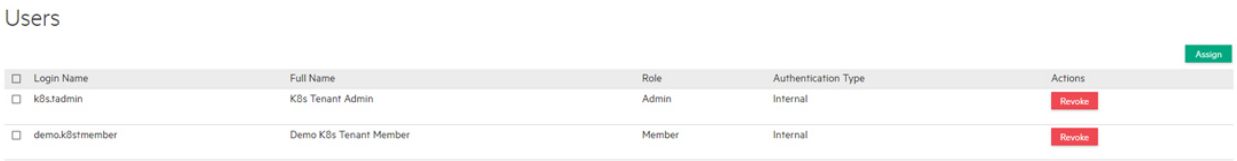
Related reference

Users and Roles on page 130

Viewing and Assigning Kubernetes Tenant Users

This topic describes associating users with roles in Kubernetes tenants in HPE Ezmeral Runtime Enterprise.

Selecting **Users** in the main menu opens the **Users** screen, which displays the users who are assigned to the current Kubernetes tenant.



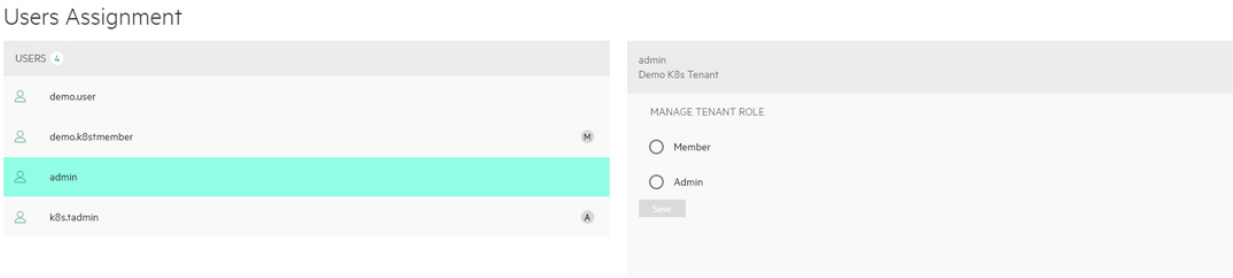
The top of the screen includes the **Assign** button. Clicking this button opens the **Users Assignment** screen, which allows you to assign users to the current Kubernetes tenant. See Assigning User Roles, below.

This screen displays the following information for each user who has a role in the current Kubernetes tenant:

- **Login Name:** Username of the user.
- **Full Name:** Full name of the user.
- **Role:** Role of the user within the current Kubernetes tenant. This will be either **Member** or **Admin**. See Users and Roles.
- **Actions**: Clicking the **Revoke** button for a user revokes their role from the current Kubernetes tenant. This does not affect any other roles the user may have in HPE Ezmeral Runtime Enterprise.

Assigning User Roles

Clicking the **Assign** button in the **Users** screen opens the **Users Assignments** screen, which allows you to assign roles in the current Kubernetes tenant to users.



To assign a role to a user:

1. Select the user to whom you want to assign a role in the **USERS** column.
2. Check the appropriate **MANAGE TENANT ROLE** radio button to assign the desired role.
   - Checking the **Member** radio button makes the selected user a Member of the current tenant.
   - Checking the **Admin** radio button makes the selected user a Tenant Administrator of the current Kubernetes tenant.
3. Click **Save**.

You may repeat this process for each additional user you want to assign.

DataTaps for Tenant/Project Administrators

The topics in this section tasks and information related to DataTaps for Kubernetes Tenant/Project Administrators in HPE Ezmeral Runtime Enterprise.

About DataTaps

DataTaps expand access to shared data by specifying a named path to a specified storage resource. Applications running within virtual clusters that can use the HDFS filesystem protocols can then access paths within that resource using that name, and DataTap implements Hadoop File System API. This allows you to run jobs using your existing data systems without the need to make time-consuming copies or transfers of your data. Tenant/Project Administrator users can quickly and easily build, edit, and remove DataTaps using the **DataTaps** screen, as described in The DataTaps Screen (Admin). Tenant Member users can access DataTaps by name.

Each DataTap requires the following properties to be configured, depending on the type of storage being connected to (MapR, HDFS, HDFS with Kerberos, or NFS):

- **Name:** A unique name for each DataTap. This name may contain letters (A-Z or a-z), digits (0-9), and hyphens (-), but may not contain spaces. You can use the name of a valid DataTap to compose DataTap URIs that you pass to applications as arguments. Each such URI maps to some path on the storage system that the DataTap points to. The path indicated by a URI might or might not exist at the time you start a job, depending on what the application wants to do with that path. Sometimes the path must indicate a directory or file that already exists, because the application intends to use it as input. Sometimes, the path must not currently exist, because the application expects to create it. The semantics of these paths are entirely application- dependent, and are identical to their behavior when running the application on a physical Hadoop or Spark platform.
- **Description:** Brief description of the DataTap, such as the type of data or the purpose of the DataTap.
- **Type:** Type of file system used by the shared storage resource associated with the DataTap (**MAPR**, **HDFS**, or **NFS**). This is completely transparent to the end job or other process using the DataTap.

The following fields depend on the DataTap type:

- MapR
- HDFS
- NFS on page 395
- GCS on page 395

MapR

NOTE: All of the links to MapR articles in this section will open in a new browser tab/window.

A MapR DataTap is configured as follows:

- **Cluster Name:** Name of the MapR cluster. See the MapR articles Creating the Cluster and Creating a Volume articles.
- **CLDB Hosts:** DNS name or address of the container location database of a MapR cluster. See the MapR article Viewing CLDB Information.
- **Port:** Port for the namenode service on the host used to access the MapR file system. See the MapR article Specifying Ports.
- **Mount Path:** Complete path to the directory containing the data within the specified MapR file system. You can leave this field blank if you intend the DataTap to point at the root of the MapR cluster. See the MapR articles Viewing Volume Details and Creating a Volume.
- **MapR Secure:** Checking this check box if MapR cluster is secured. When the MapR cluster is secured, all network connections require authentication, and moving data is protected with wire-level encryption. MapR allows applying direct security protection for data as it comes into and out of the platform without requiring an external security manager server or a particular security plug-in for each ecosystem component. The security semantics are applied automatically on data being retrieved or stored by any ecosystem component, application, or users. See the MapR article Security.
- **Ticket Source:** Select the ticket source. This will be one of the following:
  - **Upload Ticket File:** This is enabled when Ticket source is selected as **Use Existing File**.
  - **Use the existing one:** To use the existing ticket details.

- **Ticket file:** This will be one of the following:
  - When **Upload Ticket File** is selected, **Browse** button is enabled to select the tiket file.
  - When **Use the Existing One** is selected, it is the name of the existing ticket file.
- **Enable Impersonation:** When you enable impersonation, when a user signs into the container and creates a file in the MapR cluster through the DataTap connection, ownership of that file is assigned to that user. If the user does not exist in the MapR cluster, then the connection between the DataTap and the MapR cluster is rejected. Typically, administrators ensure that the same users exist in both the container and the MapR cluster by configuring both the container and the MapR cluster with the same AD/LDAP settings.
- **Select Ticket Type:** Select the ticket type. This will be one of the following:
  - **User:** Grants access to individual users with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service**: Accesses services running on client nodes with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service (with impersonation):** Accesses services running on client nodes to run jobs on behalf of any user. The ticket cannot be used to impersonate the `root` or `mapr` users.
  - **Tenant:** Allows tenant users to access tenant volumes in a multi-tenant environment. The ticket can impersonate any user.
- **Ticket User:** Username to be included in the ticket for authentication.
- **MapR Tenant Volume:** Indicates whether or not the mount path is a MapR tenant volume. See the MapR article Setting Up a Tenant.
- **Enable Passthrough**: Select this box to enable Passthrough mode.

See the following examples for additional information:

- Sample MAPR DataTap - No Impersonation
- Sample MAPR DataTap - Impersonation

HDFS

An HDFS DataTap is configured as follows:

- **Host:** DNS name or IP address of the server providing access to the storage resource. For example, this could be the host running the namenode service of an HDFS cluster.
- **Standby NameNode:** DNS name or IP address of a standby namenode host that an HDFS DataTap will try to reach if it cannot contact the primary host. This field is optional; when used, it provides high-availability access to the specified HFDS DataTap.
- **Port:** For HDFS DataTaps, this is the port for the namenode server on the host used to access the HDFS file system.
- **Path:** Complete path to the directory containing the data within the specified HDFS file system. You can leave this field blank if you intend the DataTap to point at the root of the specified file system.
- **Kerberos parameters:** If the HDFS DataTap has Kerberos enabled, then you will need to specify additional parameters. HPE Ezmeral Runtime Enterprise supports two modes of user access/ authentication.

  - Proxy mode permits a “proxy user” to be configured to have access to the remote HDFS cluster. Individual users are granted access to the remote HDFS cluster by the proxy user configuration. Mixing and matching distributions is permitted between the compute Hadoop cluster and the remote HDFS.
  - Passthrough mode passes the credentials of the current user to the remote HDFS cluster for authentication.
- HDFS file systems configured with TDE encryption as well as cross-realm Kerberos authentication are supported. See HDFS DataTap TDE Configuration and HDFS DataTap Cross-Realm Kerberos Authentication for additional configuration instructions.

NFS

NOTE: This option is not available for Kubernetes tenants.

An NFS DataTap is configured as follows:

- **Host:** DNS name or IP address of the server providing access to the storage resource.
- **Share:**This is the exported share on the selected host.
- **Path:** Complete path to the directory containing the data within the specified NFS share. You can leave this field blank if you intend the DataTap to point at the root of the specified share.

GCS

An GCS DataTap is configured as follows:

- **Bucket Name:** Specify the bucket name for GCS.
- **Credential File Source:** This will be one of the following:
  - When **Upload Ticket File:** is selected, **Browse** button is enabled to select in the **Credential File**. The credential file is a JSON file that contains the service account key.
  - When **Use the Existing One:** is selected, enter the name of the previously uploaded credential file. The credetial file is a JSON file that contains the service account key.
- **Proxy:** This is optional. Specify http proxy to access GCS.
- **Mount Path:**Enter a path within the bucket that will serve as the starting pointfor the DataTap. If the path is not specified, the starting point will default to the bucket.

Using a DataTap

The storage pointed to by a DataTap can be accessed via a URI that includes the name of the DataTap.

A DataTap points to the top of the “path” configured for the given DataTap. The URI has the following form:

```
dtap://datatap_name/
```

In this example, `datatap_name` is the name of the DataTap that you wish to use. You can access files and directories further in the hierarchy by appending path components to the URI:

```
dtap://datatap_name/some_subdirectory/another_subdirectory/some_file
```

For example, the URI `dtap://mydatatapr/home/mydirectory` means that the data is located within the `/home/mydirectory` directory in the storage that the DataTap named `mydatatap` points to.

DataTaps exist on a per-tenant basis. This means that a DataTap created for Tenant A cannot be used by Tenant B. You may, however, create a DataTap for Tenant B with the exact same properties as its counterpart for Tenant A, thus allowing both tenants to access the same storage resource. Further, multiple jobs within a tenant may use a given DataTap simultaneously. While such sharing can be useful, be aware that the same cautions and restrictions apply to these use cases as for other types of shared storage: multiple jobs modifying files at the same location may lead to file access errors and/or unexpected job results.

Users who have a Tenant Administrator role can view and modify detailed DataTap information. Members can only view general DataTap information and are unable to create, edit, or remove a DataTap.

CAUTION: Data conflicts can occur if more than one DataTap points to a location being used by multiple jobs at once.

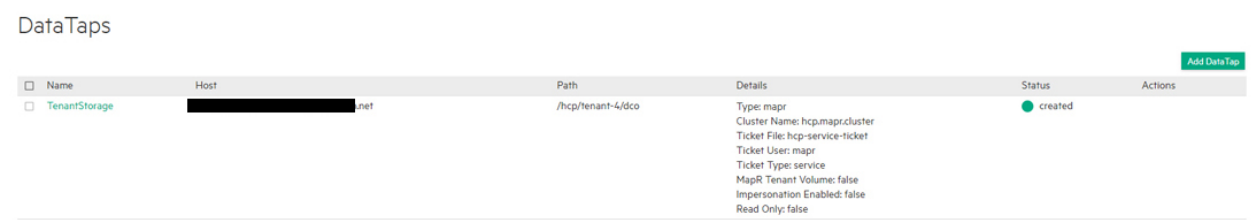
CAUTION: Editing or deleting a DataTap while it is being used by one or more running jobs can cause errors in the affected jobs.

More information

Troubleshooting DataTap Issues on page 944

The DataTaps Screen (Admin)

Selecting **DataTaps** in the main menu opens the **DataTaps** screen. The information and functions on this screen will vary depending on your role. For Tenant Administrators, the **DataTaps** screen for Tenant Administrators appears as shown in the following image.



This screen contains the following buttons:

- **Create:** Clicking this button opens the **Create New DataTap** screen. See Creating a New DataTap.
- **Delete:** Clicking this button deletes the selected DataTap(s) from the tenant. See Deleting a DataTap.

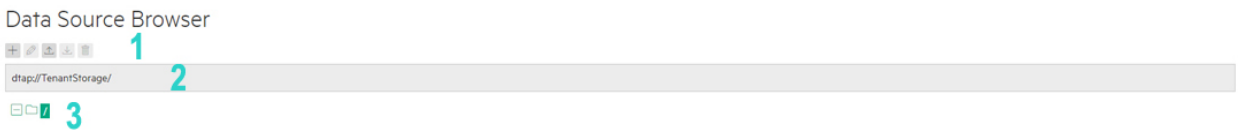
The table on this screen contains the following information and functions:

- **Name:** Name of the DataTap. Clicking a name in this column opens the **DataTap Browser** screen for the selected DataTap. See The DataTap Browser Screen.
- **Description:** Brief description of the DataTap.
- **Host:** DNS name or IP address of the service providing access to the shared storage resource associated with the DataTap.
- **Path:** Location of the root directory of the DataTap. This field is blank if the DataTap points to the root of the specified share/volume/file system.
- **Details:** Detailed DataTap information, such as:
  - **Type:** Type of file system used by the storage resource (**MAPR**, **HDFS**, or **NFS**).
  - **Additional Info:** Whether (**True**) or not (**False**) this DataTap is read-only. This column will also display **Kerberos Protected** for an HDFS DataTap with Kerberos protection enabled.

- **Actions:** The following actions are available for each DataTap (except the default **TenantStorage** DataTap):
  - **Edit:** Clicking the **Edit** icon (pencil) in the **Actions** column opens the **Edit DataTap** screen. Editing a DataTap that is in use by a running job may cause file access errors within that job. See Editing an Existing DataTap. You cannot edit the **TenantStorage** DataTap.
  - **Delete:** Clicking the **Delete** icon (trash can) in the **Actions** column deletes the DataTap from the tenant. See Deleting a DataTap.

The DataTap Browser Screen

In the **DataTaps** screen, clicking the name of a DataTap opens the **DataTap Browser** screen for the selected DataTap.



This screen contains the following information and functions:

- **File/Directory Buttons (1):** These buttons allow you to create and delete directories and files, upload files, and rename files and directories. See Uploading and Downloading Files.
- **DataTap URI (2):** This field provides the full path to the currently-selected directory or file.
- **Directory listing (3):** This list presents a hierarchical view of the directories and files that can be accessed by the selected DataTap. The **File/Directory** buttons are enabled or disabled depending on your selections in this listing.
  - Clicking an item in this list selects that item and makes additional functions available. See Uploading and Downloading Files.
  - Clicking an **Expand** icon (+) next to a collapsed directory expands that directory to reveal any subdirectories and/or files within that directory.
  - Clicking a **Collapse** icon (-) next to an expanded directory collapses that directory to hide any subdirectories and/or files within that directory.

Uploading and Downloading Files

The **Directory Listing** area of the **DataTap Browser** screen contains an expandable tree view of the directories underneath the root directory of the selected DataTap.

NOTE: Various **File/Directory** buttons will become available depending on your directory/file selection. This image shows all five buttons enabled for documentation purposes, but this will not happen during actual DataTap use.

In this view:

- Clicking a plus sign (**+**) next to a directory expands that directory to display the file(s) and sub-directories (if any) under the selected directory.

- Clicking a minus sign (**-**) next to a directory collapses the view of the file(s) and sub-directories (if any) under the selected directory.

When you are browsing locations within a locally-shared storage service created at deployment install time, the **File/Directory** buttons allow you to add, rename, and remove files and directories. For any other DataTap, the **DataTap Browser** screen will allow you to view the file/directory structure and select paths for various UI purposes. In this case, you will need to upload/download files and/or create/remove directories from outside the web interface using some native client appropriate for the storage service. For certain operations (like creating a directory), it may also be useful to access the DataTap from within a virtual node and then manually perform `hadoop fs` operations on it.

From left to right, the **File/Directory** buttons are:

- Selecting a directory and then clicking the **Create Directory** button (plus sign) opens the **Create new directory under /directory** window, where **/directory** is the name of the currently selected directory. Entering a name in the field and then clicking **OK** creates a new sub-directory and closes the window.

- Selecting a directory or file and then clicking the **Rename** button (pencil) opens the **Rename item** window, where **item** is the name of the currently selected directory or file. Entering a name in the field and then clicking **OK** renames the selected directory or file.

- Selecting a directory and then clicking the **Upload** button (up arrow) opens a standard **Upload** dialog, which allows you to locate, select, and upload a file to the selected directory. The dialog appearance will vary based on your OS and browser settings.

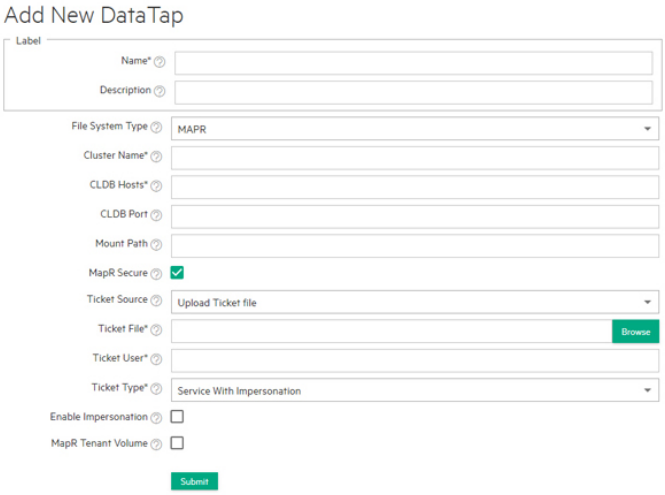
- Selecting a file and then clicking the **Download** button (down arrow) opens a standard **Save As** dialog, which allows you to save the selected file to a directory on either your local hard drive or any network storage that you have access to.

- Selecting a directory or file and then clicking the **Delete** button (trash can) deletes the selected directory or file. Deleting a directory also deletes all of the sub-directories and files within that directory, if any.

CAUTION: Do not rename or delete a directory or file that is in use, as this could cause job failures and other errors. there is no undo function when deleting a directory or file.

Creating a New DataTap

Tenant Administrators can create DataTaps. Clicking the **Create** button in the **DataTaps** screen opens the **Create New DataTap** screen.



DataTaps are created on a per-tenant basis. This means that a DataTap created in Tenant A is not available to Tenant B. You may, however, choose to create DataTaps in different tenants that point to the same storage path; in this situation, jobs in different tenants can access the same storage simultaneously. Also, multiple jobs within a tenant may use a given DataTap simultaneously. While such sharing can be useful, be aware that the same cautions and restrictions apply to these use cases as for other types of shared storage: multiple jobs modifying files at the same location may lead to file access errors and/or unexpected job results.

CAUTION: Creating multiple DataTaps to the same directory can lead to conflicts and potential data loss.

NOTE:

This article contains generic instructions for creating a DataTap. Please see the following for more specific examples:

- Sample MAPR DataTap - No Impersonation
- Sample MAPR DataTap - Impersonation

To create a DataTap:

1. Please see About DataTaps on page 122 for important limitations on where you can create DataTaps.
2. Enter a name for the DataTap in the **Name** field. This name may contain letters (A-Z or a-z), digits (0-9), and hyphens (-), but may not contain spaces.
3. Enter a brief description for the DataTap in the **Description** field.
4. You can make a DataTap read only by checking the **Read Only** check box. Clearing this check box allows read/write access.
5. Select the file system type using the **Select Type** pull-down menu. The available options are:
   - **MAPR:** See MAPR Parameters, below.
   - **HDFS:** See HDFS Parameters, below.
   - **NFS:** See NFS Parameters, below. This option is not available for Kubernetes tenants.
   - **GCS:** See GCS Parameters on page 402, below.

6. Review the entries you made in Steps 1-6 to make sure they are accurate.

When you have finished modifying the parameters for the DataTap, click **Submit** to create the new DataTap.

NOTE: If you need to configure wire encryption and/or Transparent Data Encryption (TDE), then please see HDFS DataTap Wire Encryption and/or HDFS DataTap TDE Configuration, as appropriate.

MAPR Parameters

If you selected **MAPR** in Step 5, above, then enter the following parameters:

- **Cluster Name:** Name of the MapR cluster. See the MapR articles Creating the Cluster and Creating a Volume articles.
- **CLDB Hosts:** DNS name or address of the container location database of a MapR cluster. See the MapR article Viewing CLDB Information.
- **Port:** Port for the CLDB server used to access the MapR file system. See the MapR article Specifying Ports.
- **Mount Path:** Complete path to the directory containing the data within the specified MapR file system. You can leave this field blank if you intend the DataTap to point at the root of the MapR cluster. See the MapR articles Viewing Volume Details and Creating a Volume.
- **MapR Secure:** Checking this check box if MapR cluster is secured. When the MapR cluster is secured, all network connections require authentication, and all moving data is protected with wire-level encryption. MapR allows applying direct security protection for data as it comes into and out of the platform without requiring an external security manager server or a particular security plug-in for each ecosystem component. The security semantics are applied automatically on data being retrieved or stored by any ecosystem component, application, or users. See the MapR article Security.
- **Ticket:** Enter the complete path to the MapR ticket. MapR uses tickets for authentication. Tickets contain keys that are used to authenticate users and MapR servers. In addition, certificates are used to implement server authentication. Every user who wants to access a secured cluster must have a MapR ticket. Tickets are encrypted to protect their contents. See the MapR articles Tickets and How Tickets Work.
-
  - **Ticket Source:** Select the ticket source. This will be one of the following:
    - **Upload Ticket File:** This is enabled when Ticket source is selected as **Use Existing File**.
    - **Use the existing one:** To use the existing ticket details.
  - **Ticket file:** This will be one of the following:
    - When **Upload Ticket File** is selected, **Browse** button is enabled to select the tiket file.
    - When **Use the Existing One** is selected, it is the name of the existing ticket file.
  - **Enable Impersonation:** Enable user impersonation. To enable user impersonation, user authentication, such as AD/LDAP should be configured at the MapR cluster side.
  - **Select Ticket Type:** Select the ticket type. This will be one of the following:
    - **User:** Grants access to individual users with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.

  - **Service**: Accesses services running on client nodes with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service (with impersonation):** Accesses services running on client nodes to run jobs on behalf of any user. The ticket cannot be used to impersonate the `root` or `mapr` users.
  - **Tenant:** Allows tenant users to access tenant volumes in a multi-tenant environment. The ticket can impersonate any user.
- **Ticket User:** Username to be included in the ticket for authentication.
- **MapR Tenant Volume:** Indicates whether or not the mount path is a MapR tenant volume. See the MapR article Setting Up a Tenant.
- **Enable Passthrough**: Check this check box to enable Passthrough mode.

See the following examples for additional information:

- Sample MAPR DataTap - No Impersonation
- Sample MAPR DataTap - Impersonation

HDFS Parameters

If you selected **HDFS** in Step 5, above, then enter the following parameters:

- **Host:** DNS name or IP address of the server providing access to the storage resource. For example, this could be the host running the namenode service of an HDFS cluster.
- **Standby NameNode:** DNS name or IP address of a standby namenode host that an HDFS DataTap will try to reach if it cannot contact the primary host. This field is optional; when used, it provides high-availability access to the specified HFDS DataTap.
- **Port:** For HDFS DataTaps, this is the port for the namenode server on the host used to access the HDFS file system.
- **Path:** Complete path to the directory containing the data within the specified HDFS file system. You can leave this field blank if you intend the DataTap to point at the root of the specified file system.
- **Kerberos parameters:** If the HDFS DataTap has Kerberos enabled, then you will need to specify additional parameters. HPE Ezmeral Runtime Enterprise supports two modes of user access/ authentication.
  - Proxy mode permits a “proxy user” to be configured to have access to the remote HDFS cluster. Individual users are granted access to the remote HDFS cluster by the proxy user configuration. Mixing and matching distributions is permitted between the compute Hadoop cluster and the remote HDFS.
  - Passthrough mode passes the credentials of the current user to the remote HDFS cluster for authentication.
- HDFS file systems configured with TDE encryption as well as cross-realm Kerberos authentication are supported. See HDFS DataTap TDE Configuration and HDFS DataTap Cross-Realm Kerberos Authentication for additional configuration instructions.

Continue from Step 6, above, after entering the HDFS parameters.

NFS Parameters

NOTE: This option is not available for Kubernetes tenants.

If you selected **NFS** in Step 5, above, then enter the following parameters:

- **Host:** DNS name or IP address of the server providing access to the storage resource.
- **Share:**This is the exported share on the selected host.
- **Path:** Complete path to the directory containing the data within the specified NFS share. You can leave this field blank if you intend the DataTap to point at the root of the specified share.

Also, be sure to configure the storage device to allow access from each host and each Controller and Worker that will using this DataTap.

Continue from Step 6, above, after entering the NFS parameters.

GCS Parameters

An GCS DataTap is configured as follows:

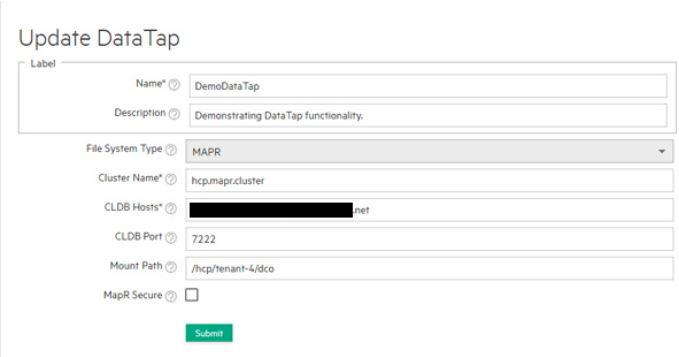
- **Bucket Name:** Specify the bucket name for GCS.
- **Credential File Source:** This will be one of the following:
  - When **Upload Ticket File:** is selected, **Browse** button is enabled to select in the **Credential File**. The credential file is a JSON file that contains the service account key.
  - When **Use the Existing One:** is selected, enter the name of the previously uploaded credential file. The credetial file is a JSON file that contains the service account key.
- **Proxy:** This is optional. Specify http proxy to access GCS.
- **Mount Path:**Enter a path within the bucket that will serve as the starting pointfor the DataTap. If the path is not specified, the starting point will default to the bucket.

More information

Troubleshooting DataTap Issues on page 944

Editing an Existing DataTap

Tenant Administrators have the ability to edit DataTaps. In the **DataTaps** screen, clicking the **Edit** icon (pencil) in the **Actions** column of the table opens the **Update DataTap** screen for the selected DataTap.



NOTE: You cannot edit a DataTap if Lockdown mode is enabled. See Lockdown Mode on page 916.

Please see About DataTaps for important limitations on the directories that a DataTap can point to.

To edit a DataTap, you may modify some or all of the following:

- **Name:** Rename the DataTap by entering a new name in the **Name** field. This name may contain letters (A-Z or a-z), digits (0-9), and hyphens (-), but may not contain spaces.

- **Description:** Update the description of the DataTap by providing a new description in the **Description** field.
- Select the storage device type using the **Select Type** pull-down menu.
  - **MAPR:** See MAPR Parameters, below.
  - **HDFS:** See HDFS Parameters, below.
  - **NFS:**See NFS Parameters, below. NFS is not available for Kubernetes tenants.

When you have finished modifying the parameters for the DataTap, click **Submit** to modify that DataTap.

CAUTION:

Editing a DataTap that is being used by a currently running job can cause file access errors within that job.

NOTE: If you need to configure wire encryption and/or Transparent Data Encryption (TDE), then please see HDFS DataTap Wire Encryption and/or HDFS DataTap TDE Configuration, as appropriate.

MAPR Parameters

If you selected **MAPR** as the DataTap type, then enter the following parameters::

- **Cluster Name:** Name of the MapR cluster. See the MapR articles Creating the Cluster and Creating a Volume articles.
- **CLDB Hosts:** DNS name or address of the container location database of a MapR cluster. See the MapR article Viewing CLDB Information.
- **Port:** Port for the CLDB server used to access the MapR file system. See the MapR article Specifying Ports.
- **Mount Path:** Complete path to the directory containing the data within the specified MapR file system. You can leave this field blank if you intend the DataTap to point at the root of the MapR cluster. See the MapR articles Viewing Volume Details and Creating a Volume.
- **MapR Secure:** Checking this check box if MapR cluster is secured. When the MapR cluster is secured, all network connections require authentication, and all moving data is protected with wire-level encryption. MapR allows applying direct security protection for data as it comes into and out of the platform without requiring an external security manager server or a particular security plug-in for each ecosystem component. The security semantics are applied automatically on data being retrieved or stored by any ecosystem component, application, or users. See the MapR article Security.
- **Ticket:** Enter the complete path to the MapR ticket. MapR uses tickets for authentication. Tickets contain keys that are used to authenticate users and MapR servers. In addition, certificates are used to implement server authentication. Every user who wants to access a secure cluster must have a MapR ticket. Tickets are encrypted to protect their contents. See the MapR articles Tickets and How Tickets Work.
- **Ticket Type:** Select the ticket type. This will be one of the following:
  - **User:** Grants access to individual users with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service**: Accesses services running on client nodes with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.

  - **Service (with impersonation):** Accesses services running on client nodes to run jobs on behalf of any user. The ticket cannot be used to impersonate the `root` or `mapr` users.
  - **Tenant:** Allows tenant users to access tenant volumes in a multi-tenant environment. The ticket can impersonate any user.
- **Ticket User:** Username to be included in the ticket for authentication.
- **MapR Tenant Volume:** Indicates whether or not the mount path is a MapR tenant volume. See the MapR article Setting Up a Tenant.
- **Enable Impersonation:** Enable user impersonation.

HDFS Parameters

If you selected **HDFS** as the DataTap type, then enter the following parameters:

- **Host:** Enter either the hostname or IP address of the HDFS NameNode in the **Host** field.
- **Standby NameNode Host:** Enter the hostname or IP address of the HDFS standby NameNode, if any, in the **Standby NameNode Host** field.
- **Port:** Enter the NameNode port number in the **Port** field. Leave blank to use the default HDFS NameNode port.
- **Path:** Enter the HDFS directory under the share to use for the DataTap in the **Path** field. You may also click the **Browse** button to open an explorer window to navigate to the desired directory. You can leave this field blank if you intend the DataTap to point the root of the specified file system.
- **Kerberos Protection:** You can enable or disable Kerberos protection for the selected DataTap by checking or clearing the **Kerberos Protected** check box, as appropriate. See HDFS DataTap Kerberos Security.
- **Username:** If needed, you can enter a valid username for accessing the HDFS.

NFS Parameters

If you selected **NFS** as the DataTap type, then enter the following parameters:

- **Host:** Enter either the hostname or IP address of the file system host in the **Host** field.
- **Share:** Enter the name of the share in the **Share** field.
- **Path:** This field specifies where the top of the DataTap's file system is rooted. For manually created DataTaps, this field must either be empty, or it must point to an existing subdirectory of the indicated storage system. For an automatically created tenant default DataTap, then HPE Ezmeral Runtime Enterprise will automatically create the indicated subdirectory if necessary, whenever any writes are done to that DataTap. Either enter the directory under the share to use for the DataTap in the **Path** field (click the **Browse** button to open an explorer window to navigate to the desired directory, if desired), or leave this field blank to point the DataTap to point the root of the specified share.

Also, be sure to configure the storage device to allow access from each host and each Controller and Worker that will using this DataTap.

Deleting a DataTap

Tenant Administrators have the ability to delete DataTaps. To delete one or more DataTap(s):

1. Open the **DataTaps** screen.
2. Either:

- Select one or more DataTap(s) by checking the appropriate check box(es) in the table, and then click the **Delete** button.
- Click the **Delete** icon (trash can) for a specific DataTap.

3. A popup warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without deleting the DataTap.

CAUTION: DELETING A DATATAP THAT IS BEING USED BY A CURRENTLY-RUNNING JOB MAY CAUSE FILE ACCESS ERRORS WITHIN THAT JOB.

NOTE: Deleting a DataTap does not affect your data. If you accidentally delete a DataTap, simply create a new one that points to the same location.

NOTE: You cannot delete the **TenantStorage** DataTap.

DataTap Tensorflow Support

Tensorflow images support DataTaps by:

1. Placing the shared library in the `/opt/bludata/` directory.
2. Installing and configuring the necessary Hadoop package.
3. Configuring required environment variables.

After creating a virtual cluster with a Tensorflow image, log in to one of the virtual nodes/containers in that cluster, and then verify basic I/O functionality by executing the following commands in a Python shell:

```
import tensorflow as tf

import os


#check CXX11_ABI_FLAG

from tensorflow.python.framework.versionsimport CXX11_ABI_FLAG

CXX11_ABI_FLAG


#load bdfs shared library

bdfs_file_system_library= os.path.join("/opt/
bluedata","libbdfs_file_system_shared_r1_13.so")

tf.load_library(bdfs_file_system_library)


#write to a test file

with tf.gfile.Open("dtap://TenantStorage/tensorflow/dtap.txt", 'w') as f:

    f.write("This is the dtaptest file")


#read from the test file

with tf.gfile.Open("dtap://TenantStorage/tensorflow/dtap.txt", 'r') as f:

    content = f.read()
```

```
# show the connect of the file

Content
```

Accessing DataTaps in Kubernetes Pods
Describes the generic process for configuring Kubernetes pods to access DataTaps, including considerations and steps for Hadoop 2.x and Hadoop 3.x applications.

About this task

The `hpecp-agent` observes pod creation. If the pod includes the `hpecp.hpe.com/dtap` label, the following occurs:

- `hpecp-agent` adds a sidecar container that implements the DataTaps. The `hpecp-agent` creates an `emptyDir` volume named `dtap-shared-vol`. This volume is mounted to the `/opt/bdfs` directory of the sidecar container and the application container.
- On startup, based on the appropriate Hadoop version, the sidecar container prepares the appropriate `bluedata-dtap.jar` file in the `/opt/bdfs` directory.
- The `/opt/bdfs` directory in the sidecar DataTap container and in the application container mounts from the same volume `dtap-shared-vol`. Thus, the application container can also directly access the `bluedata-dtap.jar` in the `/opt/bdfs` directory.

The following procedure is a generic example only.

- KubeDirector applications included with HPE Ezmeral Runtime Enterprise are preconfigured to be able to access DataTaps, and you need only set the pod label. See Accessing DataTaps in KubeDirector Applications.
- Spark Operator applications must be configured for DataTap access as described in Tutorial: Spark Configuration and Execution on Kubernetes.
- If a pod has the label `hpecp.hpe.com/dtap: hadoop2` or `hpecp.hpe.com/dtap: hadoop3`, the DataTap sidecar container runs until the pod is deleted. In some scenarios—such as when a user submits a Spark Operator application—the application container exits automatically after the application is completed. If the DataTap sidecar container still runs after the application container exits, the pod is unable to enter a completed status. Because the pod does not enter the completed state, the pod continues to use resources instead of those resources being released for use by other pods.

  To ensure that the DataTap sidecar container also exits automatically after the application container exits, use one of the following labels:

  - If the application is Hadoop 2.x, add the label:

    ```
    hpecp.hpe.com/dtap: hadoop2-job
    ```

  - If the application is Hadoop 2.x, add the label:

    ```
    hpecp.hpe.com/dtap: hadoop3-job
    ```

Procedure

1. Add one of the following sets of labels to the YAML file of the pod:

- If the application is Hadoop 2.x, add the following labels:

```
hpecp.hpe.com/dtap: hadoop2
hpecp.hpe.com/dtap: hadoop2-job
```

- If the application is Hadoop 2.x, add the following labels:

```
hpecp.hpe.com/dtap: hadoop3
hpecp.hpe.com/dtap: hadoop3-job
```

2. In the application container, add `bluedata-dtap.jar` to the `classpath`, and then modify the Hadoop `core-site.xml` file.

   The following example adds the `fs.dtap.impl`, `fs.AbstractFileSystem.dtap.impl`, and `fs.dtap.impl.disable.cache` to the `core-site.xml` file:

```
            fs.dtap.impl
            com.bluedata.hadoop.bdfs.Bdfs


            fs.AbstractFileSystem.dtap.impl
            com.bluedata.hadoop.bdfs.BdAbstractFS


            fs.dtap.impl.disable.cache
            false
```

Launching Kubernetes Pods to Access DataTaps

This section provides a sample YAML file called `demo.yaml` that includes an HDFS client. Note the `DataTap` label, as described in Accessing DataTaps in Kubernetes Pods:

```
apiVersion: v1
kind: Pod
metadata:
  name: demo
  namespace: k8s
  labels:
     hpecp.hpe.com/dtap: hadoop2
spec:
  containers:
  - name: app
    image: docker.io/xoxoxoxoxoxo/app:1.0
    resources:
      limits:
        cpu: "500m"
        memory: "4Gi"
      requests:
        cpu: "500m"
        memory: "4Gi"
```

The pod information is as follows after successful deployment of the `demo.yaml` file:

```
[root@intel-s02 ~]# kubectl -n k8s describe pod demo
Name:         demo
Namespace:    k8s
Priority:     0
Node:         hostname.enterprise.net/10.50.50.50
Start Time:   Mon, 20 Jul 2020 17:18:16 -0700
```

```
Labels:       hpecp.hpe.com/dtap=hadoop2
Annotations:  cni.projectcalico.org/podIP: 10.192.1.15/32
              hpecp.hpe.com/dtap-status: injected
Status:       Running
IP:           10.192.1.15
IPs:
  IP:  10.192.1.15
Containers:
  app:
    Container ID:   docker://
7c0df2c39b74643f52dc68be0752142af80b386f0f79f2f258a46ec4ead41649
    Image:          docker.io/xoxoxoxoxoxo/app:1.0
    Image ID:       docker-pullable://xoxoxoxoxoxo/
app@sha256:9ac6291b7116c083e293c56887dbaf682102a43f121eb1914f1ab0f7d6cae36e
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Mon, 20 Jul 2020 17:18:17 -0700
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:     500m
      memory:  4Gi
    Requests:
      cpu:        500m
      memory:     4Gi
    Environment:  <none>
    Mounts:
      /opt/bdfs from dtap-shared-vol (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from
default-token-jlnzg (ro)
  dtap:
    Container ID:   docker://
c7c45beedfaf8e08fe3f9b894f4cb276a341bcbd8f8e73d0c30645abe4314dcf
    Image:          bluedata/hpecp-dtap:1.63
    Image ID:       docker-pullable://bluedata/
hpecp-dtap@sha256:a2cde114efb257e457bbd839391f1706c5a28a2b467bfe6e57fa9fcc4f
14267b
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Mon, 20 Jul 2020 17:18:17 -0700
    Ready:          True
    Restart Count:  0
    Limits:
      cpu:     200m
      memory:  409Mi
    Requests:
      cpu:     200m
      memory:  409Mi
    Environment:
      K8S_DTAP_SHARED_MEMORY_SIZE:  153
      HADOOP_VERSION:               hadoop_version_2
    Mounts:
      /etc/bluedata/dtap from secret-vol-dtap (ro)
      /opt/bdfs from dtap-shared-vol (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from
default-token-jlnzg (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
```

```
Volumes:
  default-token-jlnzg:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-jlnzg
    Optional:    false
  dtap-shared-vol:
    Type:       EmptyDir (a temporary directory that shares a pod's
lifetime)
    Medium:     Memory
    SizeLimit:  <unset>
  secret-vol-dtap:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  dtap
    Optional:    false
QoS Class:       Guaranteed
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:          <none>
```

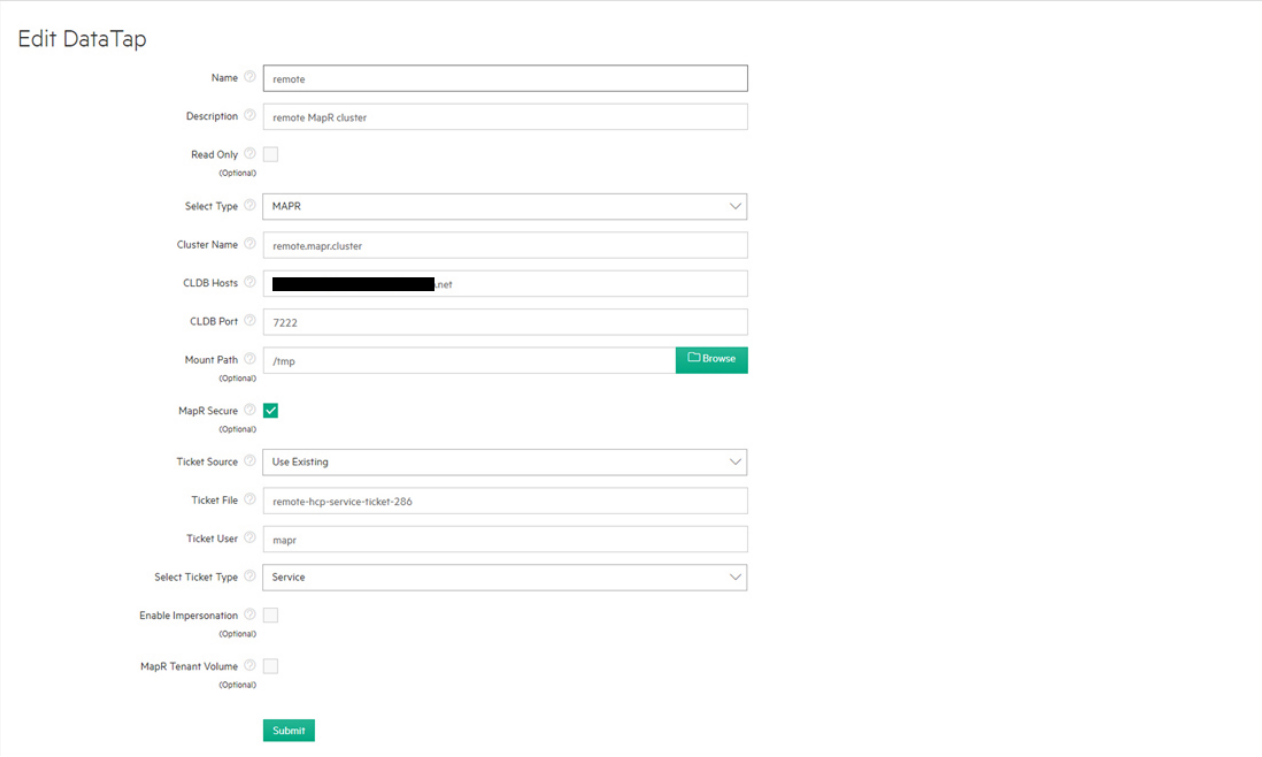
After the pod is ready, you can log in into the app container to either run HDFS commands or perform I/O operation on the tenant's DataTaps.

Accessing DataTaps in KubeDirector Applications

To access a DataTap from a KubeDirector application, add the Hadoop pod labels to the corresponding roles in the application's YAML file, as described in Accessing DataTaps in Kubernetes Pods.

Sample MAPR DataTap - No Impersonation

This image shows a sample MAPR DataTap without impersonation.

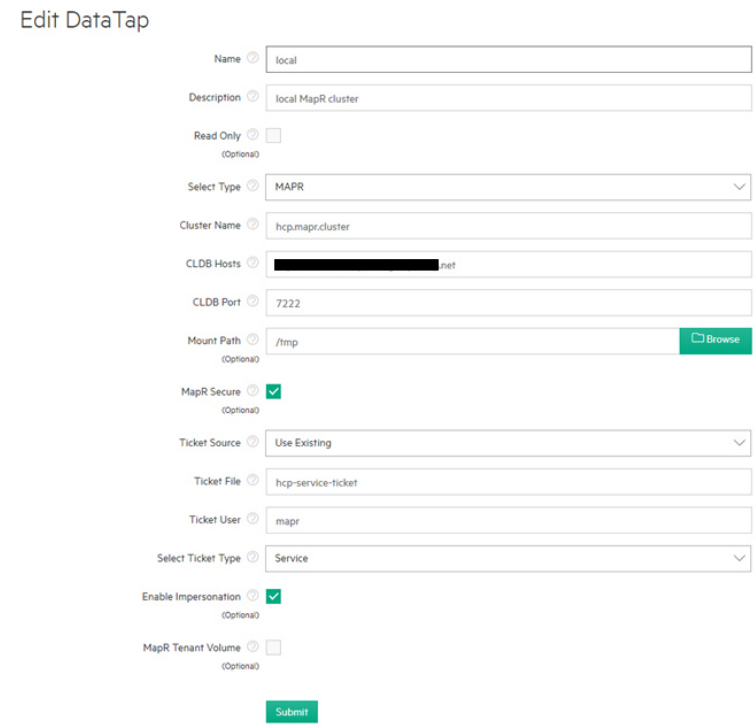


This sample DataTap Enable does not have the **Enable Impersonation** option enabled. In this case:

- If a real user logs in to the container and creates a new file against a MapR DataTap without impersonation, then the file owner will be the ticket user. For example, if the real user `testuser` logs in to a container and creates a new file by executing the command `Hadoop fs -put ./testfile dtap://remote/`, then the actual file owner of `testfile` will be `mapr` in the MapR cluster.
- If a real user logs in to the container to list the files against the MapR DataTap without impersonation, then the file owner will be the currently-logged-in user. In this case, the DataTap purposely shows the currently-logged-in user as the file owner to shield the information of the ticket user. For example, if the real user `testuser` logs in to the container to list the directory by executing the command `Hadoop fs -ls dtap://remote/`, then the output will show that the owner of the sub-directory and files is `testuser`.

Sample MAPR DataTap - Impersonation

This image shows a sample MAPR DataTap with impersonation.



This sample DataTap Enable has the **Enable Impersonation** option enabled. The following conditions need to be met in order to support impersonation:

The ticket should support impersonation. For example, if the ticket user is either `mapr` or `root`, then the ticket can be used for impersonation, and the ticket type `servicewithimpersonation` can support impersonation.

The real user should exist in the MapR cluster. If the real user does not exist in the MapR cluster, then the connection between the DataTap and the MapR cluster will be rejected. Generally, the container and the MapR cluster should be configured with the same AD/LDAP settings.

When the real user logs in to the container to create a new file against the MapR DataTap with impersonation, then the owner of file will be the real user. For example:

- If the real user `testuser` logs in to the container to create a new file by executing the command `Hadoop fs -put ./testfile dtap://local/`, then the actual file owner of `testfile` will be `testuser` in the MapR cluster.

- If the real user logs in to the container to list the files against a MapR DataTap with impersonation, then the owner of file will be the actual owner of the file.

HDFS DataTap Cross-Realm Kerberos Authentication

NOTE: This article only applies to HDFS DataTaps.

Cross-realm Kerberos authentication allows the users of one Kerberos realm to access services that reside inside a different Kerberos realm. To do this, both realms must share a key for the same principal, and both keys must share the same version number. For example, to allow a user in `REALM_A` to access `REALM_B`, then both realms must share a key for a principal named `krbtgt/REALM_B@REALM_A`. This key is unidirectional; for a user in `REALM_B` to access `REALM_A`, both realms must share a key for `krbtgt/REALM_A@REALM_B`.

Most of the responsibilities of the remote KDC server can be offloaded to a local KDC that Kerberizes the compute clusters within a tenant, while the DataTap uses a KDC server specific to the enterprise datalake. The users of the cluster come from the existing enterprise central KDC. Assuming that the enterprise has a network DNS name of `ENTERPRISE.COM`, the three Kerberos realms could be named as follows:

- **KDC Realm CORP.ENTERPRISE.COM:** This central KDC realm manages the users who run jobs in the Hadoop compute clusters. For example, `user@CORP.ENTERPRISE.COM`.
- **KDC Realm CP.ENTERPRISE.COM:** This local KDC realm Kerberizes the Hadoop compute clusters.
- **KDC Realm: DATALAKE.ENTERPRISE.COM:** This KDC Kerberizes the remote HDFS file system accessed via DataTap. For example, `dtap://remotedata`.

In this example, the user `user@CORP.ENTERPRISE.COM` can run jobs in the compute cluster that belongs to the `CP.ENTERPRISE.COM` realm, and jobs can access data residing in `dtap://remotedata` in the `DATALAKE.ENTERPRISE.COM` realm. This scenario requires a one-way Kerberos trust relationship between realms `CORP.ENTERPRISE.COM` and `DATALAKE.ENTERPRISE.COM` and `CP.ENTERPRISE.COM`, as well as a one-way trust relationship between realm `DATALAKE.ENTERPRISE.COM` and realm `CP.ENTERPRISE.COM`. More specifically:

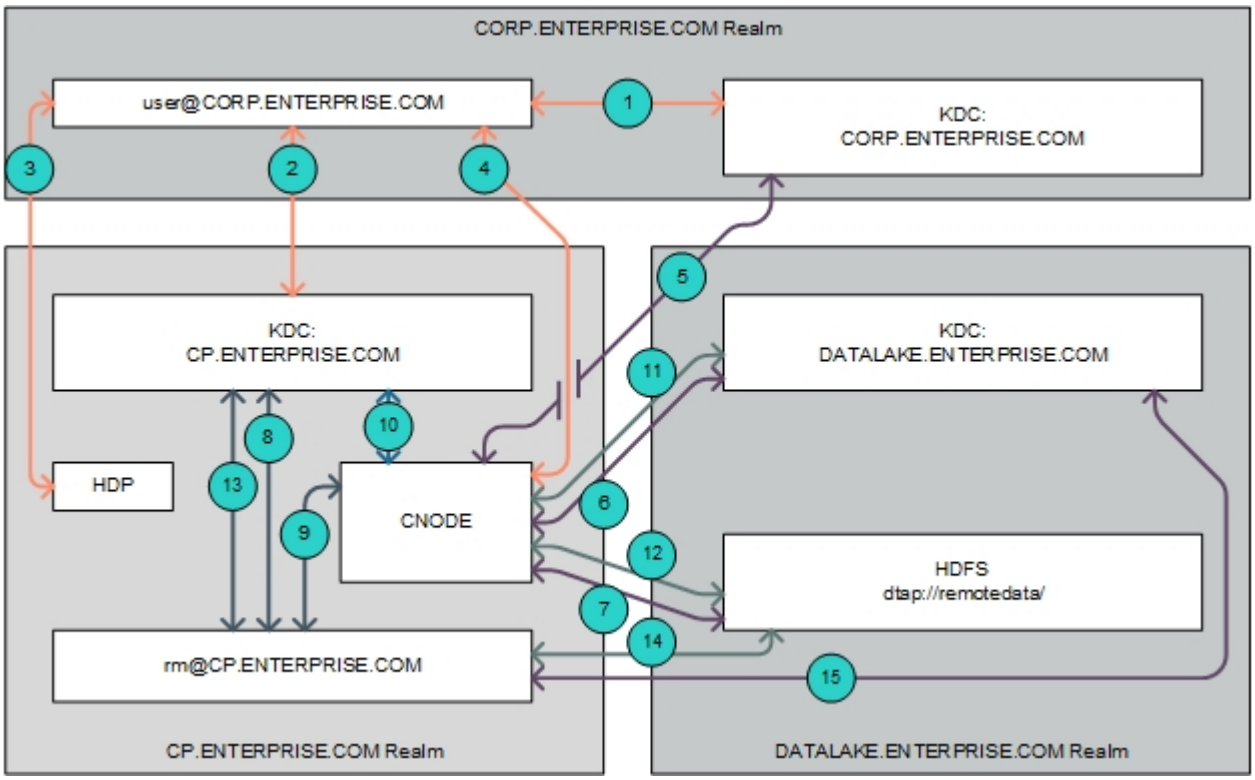
- **CP.ENTERPRISE.COM trusts CORP.ENTERPRISE.COM:** The user `user@CORP.ENTERPRISE.COM` needs to be able to access services within the compute cluster in order to perform tasks such as submitting jobs to the compute cluster YARN Resource Manager and writing the job history to the local HDFS.
- **DATALAKE.ENTERPRISE.COM trusts CORP.ENTERPRISE.COM:** The user `user@CORP.ENTERPRISE.COM` needs to be able to access `dtap://remotedata/` from the compute cluster.
- **DATALAKE.ENTERPRISE.COM trusts CP.ENTERPRISE.COM:** When the user `user@CORP.ENTERPRISE.COM` accesses dtap://remotedata/ from the compute cluster to run jobs, the `YARN/rm` user of the compute cluster (`CP.ENTERPRISE.COM`) also needs to be able to access `dtap://remotedata/` to get partition information and to renew the HDFS delegation token.

This article describes the following:

- **Access:** See Accessing a Passthrough DataTap with Cross-Realm Authentication.
- **Authentication:** See One-Way Cross-Realm Authentication.
- **Using Ambari to configure cross-realm authentication:** See Using Ambari to Configure /etc/krb5.conf.
- **Troubleshooting:** See Debugging.

Accessing a Passthrough DataTap with Cross-Realm Authentication

This diagram displays the high-level authentication flow for accessing a passthrough DataTap with cross-realm authentication:



In this example, the user `user@CORP.ENTERPRISE.COM` submits a job to the cluster that is Kerberized by the KDC realm `CP.ENTERPRISE.COM` and accesses data stored on a remote HDFS file system Kerberized by the KDC realm `DATALAKE.ENTERPRISE.COM` using the following flow (numbers correspond to the callouts in the preceding diagram):

1. The user `user@CORP.ENTERPRISE.COM` wants to send a job to a service in KDC realm `CP.ENTERPRISE.COM`. This realm is different from the one that the user belongs to. Normally, this is not allowed. However, since there is a trust relationship where realm `CP.ENTERPRISE.COM` trusts realm `CORP.ENTERPRISE.COM`, the user `user@CORP.ENTERPRISE.COM` is able to request a temporary service ticket from its home realm (`CORP.ENTERPRISE.COM`) that will be valid when submitted to the TGS (Ticket Granting Service) of the foreign realm, `CP.ENTERPRISE.COM`.

2. The user `user@CORP.ENTERPRISE.COM` submits the temporary service ticket issued by the `CORP.ENTERPRISE.COM` realm to the Ticket Granting Service (TGS) of the `CP.ENTERPRISE.COM` realm.

3. The user `user@CORP.ENTERPRISE.COM` then submits this service ticket to the YARN service in the HDP compute cluster in order to run the job.

4. When the job that user `user@CORP.ENTERPRISE.COM` submitted needs to get data from the remote HDFS, the DataTap forwards the user's TGT to the deployment CNODE service. The CNODE service finds that the realm of the TGT for `user@CORP.ENTERPRISER.COM` is `CORP.ENTERPRISE.COM` and not the same as KDC realm `DATALAKE.ENTERPRISE.COM`, which is the one used by the HDFS file system configured for the DataTap.

5. The CNODE service obtains a temporary service ticket from the `CORP.ENTERPRISE.COM` KDC server.

6. Since there is a trust relationship where realm `DATALAKE.ENTERPRISE.COM` trusts realm `CORP.ENTERPRISE.COM`, the CNODE service then obtains a service ticket from realm `DATALAKE.ENTERPRISE.COMCORP.ENTERPRISE.COM` server.

7. The CNODE service uses the `DATALAKE.ENTERRPRISE.COM` service ticket to authenticate with the NameNode service of the remote HDFS file system and access the data as user `user@CORP.ENTERPRISE.COM`.

8. While running the job submitted by `user@CORP.ENTERPRISE.COM` to the cluster, the YARN Resource Manager service will need to access the remote HDFS file system in order to get partition information. The Resource Manager service runs with principal `rm@CP.ENTERPRISE.COM`. In order to access the remote HDFS, it will need to obtain a Ticket-Granting Ticket (TGT) from the local `CP.ENTERPRISE.COM` KDC server.

9. The user `rm@CP.ENTERPRIE.COM` then accesses the DataTap. The DataTap forwards the user's TGT to the deployment CNODE service. The CNODE service finds that the realm of the TGT for `rm@CP.ENTERPRISE.COM` is not the same as KDC realm `DATALAKE.ENTERPRISE.COM`, which is the one used by the HDFS service configured for the DataTap.

10. The CNODE service obtains a temporary service ticket from the `CP.ENTERPRISE.COM` KDC server.

11. Since there is a trust relationship where realm `DATALAKE.ENTERPRISE.COM` trusts realm `CP.ENTERPRISE.COM`, the CNODE service then obtains a service ticket from realm `DATALAKE.ENTERPRIE.COM`, the KDC protecting access to the remote HDFS file system based on the temporary service ticket that was issued by the realm `CP.ENTERPRISE.COM` server.

12. The CNODE service uses the service ticket to authenticate with the NameNode Service of the remote HDFS file system and access the data as user `rm@CP.ENTERPRISE.COM`.

13. When the Resource Manager it does not use the CNODE service when it needs to renew an HDFS delegation token. Instead, the `rm@CP.ENTERPRISE.COM` user requests a temporary service ticket for the `DATALAKE.ENTERPRISE.COM` realm from the local (`CP.ENTERPRISE.COM`) KDC server. Since there is a trust relationship between realms `CP.ENTERPRISE.COM` and `DATALAKE.ENTERPRISE.COM`, the `CP.ENTERPRISE.COM` KDC server is able to issue the temporary service ticket.

14. The `rm@CP.ENTERPRISE.COM` user submits the temporary service ticket to the KDC server of the `DATALAKE.ENTERPRISE.COM` realm and gets a service ticket for the NameNode service of the remote HDFS file system.

15. The `rm@CP.ENTERPRISE.COM` user can then use the service ticket to renew the HDFS delegation token with the NameNode service of the remote HDFS.

One-Way Cross-Realm Authentication

Allowing the user `user@CORP.ENTERPRISE.COM` to run jobs on the Hadoop compute cluster requires configuring one-way cross-realm authentication between the realms `CORP.ENTERPRISE.COM` and `CP.ENTERPRISE.COM`. Further, allowing the user `user@CORP.ENTERPRISER.COM` to use the DataTap `dtap://remotedata` within the Hadoop compute cluster requires configuring one-way cross-realm authentication between the realms `CORP.ENTERPRISE.COM` and `DATALAKE.ENTERPRISE.COM`, and between the realms `CP.ENTERPRISE.COM` and `DATALAKE.ENTERPRISE.COM`. In other words, the realm `DATALAKE.ENTERPRISE.COM` trusts the realms `CP.ENTERPRISE.COM` and `CORP.ENTERPRISE.COM`, and the realm `CP.ENTERPRISE.COM` trusts realm `CORP.ENTERPRISE.COM`.

To enable these one-way cross-realm trust relationships, you will need to configure the following:

- **KDC:** See Step 1: KDC Configuration.

- **Host:** See Step 2: Host Configuration.
- **DataTap:** See Step 3: Remote DataTap Configuration.
- **Cluster:** See Step 4: Hadoop Compute Cluster Configuration.

See Using Ambari to Configure /etc/krb5.conf for information on using the Ambari interface to configure cross-realm authentication.

Step 1: KDC Configuration

Configure the KDCs as follows:

1. On the KDC server for realm DATALAKE.ENTERPRISE.COM add the following two principals:

```
krbtgt/DATALAKE.ENTERPRISE.COM@CORP.ENTERPRISE.COM
        krbtgt/DATALAKE.ENTERPRISE.COM@CP.ENTERPRISE.COM
```

2. On the KDC server for realm CP.ENTERPRISE.COM, add the following two principals:

```
krbtgt/DATALAKE.ENTERPRISE.COM@CP.ENTERPRISE.COM
        krbtgt/CP.ENTERPRISE.COM@CORP.ENTERPRISE.COM
```

3. On the KDC server for realm CORP.ENTERPRISE.COM, add the following two principals:

```
krbtgt/DATALAKE.ENTERPRISE.COM@CORP.ENTERPRISE.COM
        krbtgt/CP.ENTERPRISE.COM@CORP.ENTERPRISE.COM
```

Step 2: Host Configuration

On the host(s) where the CNODE service is running, modify the `[realms]` and `[domain_realm]` sections of the `/etc/bluedata/krb5.conf` file to add the `CORP.ENTERPRISE.COM`, `CP.ENTERPRISE.COMDATALAKE.ENTERPRISE.COM` realms. For example:

```
[root@yav-028 ~]# !cat
cat /etc/bluedata/krb5.conf
[logging]
  default = FILE:/var/log/krb5/krb5libs.log
  kdc = FILE:/var/log/krb5kdc.log
  admin_server = FILE:/var/log/kadmind.log

[libdefaults]
  default_realm = CP.ENTERPRISE.COM
  dns_loookup_realm = false
  dns_lookup_kdc = false
  ticket_lifetime = 24h
  renew_lifetime = 7d
  forwardable = true

[realms]
  CP.ENTERPRISE.COM = {
    kdc = kerberos.cp.enterprise.com
  }
  CORP.ENTERPRISE.COM = {
    kdc = kerberos.corp.enterprise.com
  }
  DATALAKE.ENTERPRISE.COM = {
    kdc = kerberos.datalake.enterprise.com
  }
```

```
                        [domain_realm]
                          .cp.enterprise.com = CP.ENTERPRISE.COM
                          .datalake.enterprise.com = DATALAKE.ENTERPRISE.COM
                          .corp.enterprise.com = CORP.ENTERPRISE.COM
                        [root@yav-28 ~]#
```

Step 3: Remote DataTap Configuration

On the remote HDFS NameNode service pointed to by the DataTap `dtap://remotedata/`, append the `auth_to_local` on the Hadoop cluster as follows:

```
RULE:[1:$1@$0](ambari-qa-hdp@epic.ENTERPRISE.COM)s/.*/ambari-qa/
RULE:[1:$1@$0](hbase-hdp@epic.ENTERPRISE.COM)s/.*/hbase/
RULE:[1:$1@$0](hdfs-hdp@epic.ENTERPRISE.COM)s/.*/hdfs/
RULE:[1:$1@$0](.*@CP.ENTERPRISE.COM)s/@.*//
RULE:[2:$1@$0](dn@CP.ENTERPRISE.COM)s/.*/hdfs/
RULE:[2:$1@$0](hbase@CP.ENTERPRISE.COM)s/.*/hbase/
RULE:[2:$1@$0](hive@CP.ENTERPRISE.COM)s/.*/hive/
RULE:[2:$1@$0](jhs@CP.ENTERPRISE.COM)s/.*/mapred/
RULE:[2:$1@$0](nm@CP.ENTERPRISE.COM)s/.*/yarn/
RULE:[2:$1@$0](nn@CP.ENTERPRISE.COM)s/.*/hdfs/
RULE:[2:$1@$0](rm@CP.ENTERPRISE.COM)s/.*/yarn/
RULE:[2:$1@$0](yarn@CP.ENTERPRISE.COM)s/.*/yarn/
RULE:[1:$1@$0](.*@CORP.ENTERPRISE.COM)s/@.*//
DEFAULT
```

Step 4: Hadoop Compute Cluster Configuration

To configure the Hadoop compute cluster:

1. Modify the `/etc/krb5.conf` file on each of the virtual nodes, as follows:

   ```
   [realms]
     CP.ENTERPRISE.COM = {
       kdc = kerberos.cp.enterprise.com
     }
     CORP.ENTERPRISE.COM = {
       kdc = kerberos.corp.enterprise.com
       }
      DATALAKE.ENTERPRISE.COM = {
       kdc=kerberos.datalake.enterprise.com
     }

   [domain_realm]
     .cp.enterprise.com = CP.ENTERPRISE.COM
     .datalake.enterprise.com = DATALAKE.ENTERPRISE.COM
     .corp.enterprise.com = CORP.ENTERPRISE.COM
   ```

2. Users in the realm `CORP.ENTERPRISE.COM` also need access to the HDFS file system in the Hadoop compute cluster. Enable this by adding the following rule to the `hadoop.security.auth_to_local` configuration file:

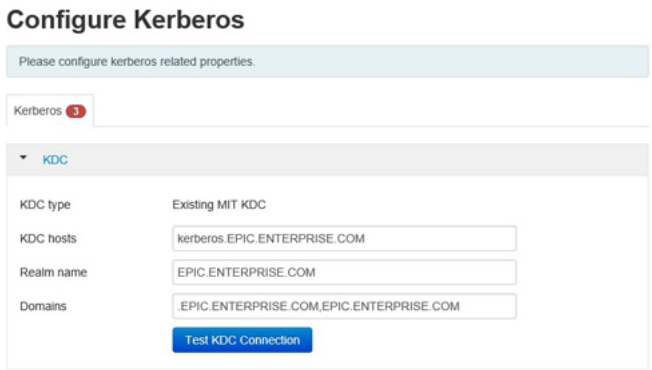
   ```
   RULE:[1:$1@$0](.*@CORP.ENTERPRISE.COM)s/@.*//
   ```

3. Restart the Hadoop services once you have finished making these changes. Do not restart the Kerberos service, because Ambari will overwrite the modified `/etc/krb5.conf` file with the original version when it finds a mismatch.

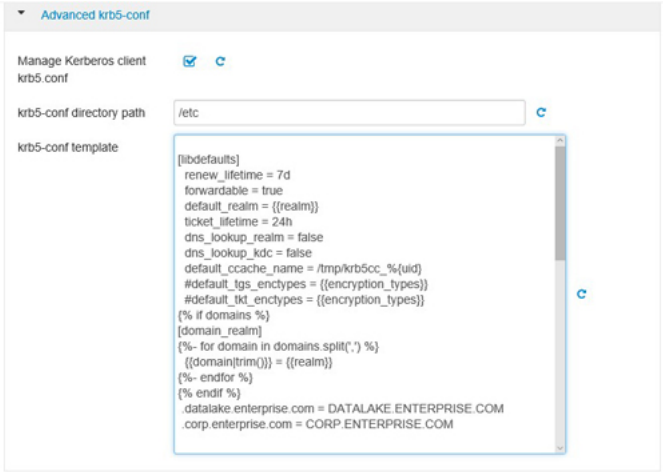
Using Ambari to Configure /etc/krb5.conf

You may modify the `/etc/krb5.conf` file using the Ambari interface by selecting **Admin>Kerberos>Configs**. The advantage of using Ambari to modify the `/etc/krb5.conf` file is that you can freely restart all services.

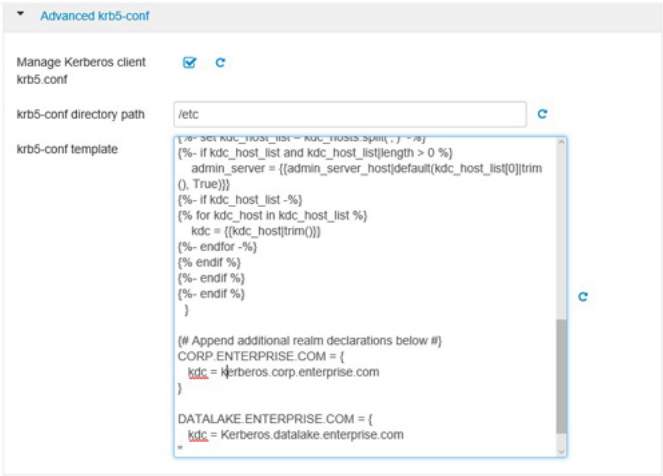
Here, the **Domains** field is used to map server host names to the name of the Kerberos realm:



The **krb5-conf** template field allows you to append additional server host names to the realm name mapping:



You may also append additional realm/KDC declarations in the **krb5-conf template** field.

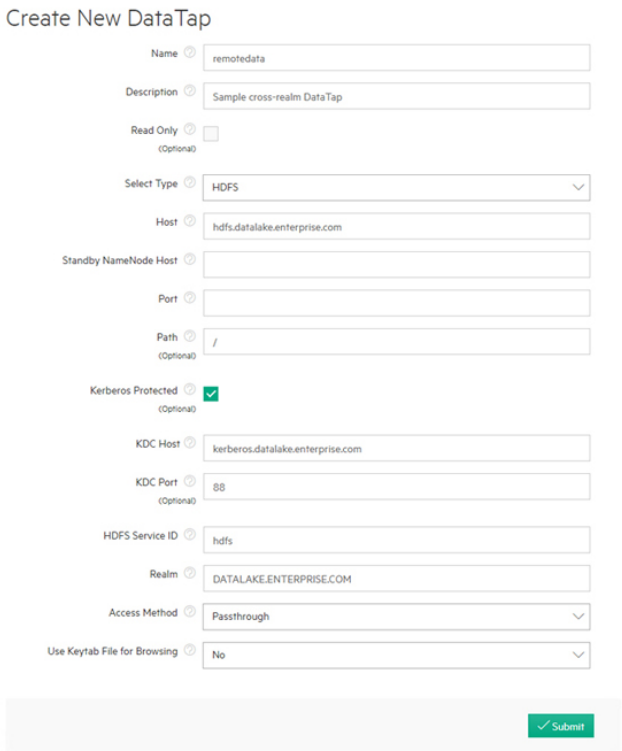


Debugging

If a failure occurs while trying to access the remote HDFS storage resource using the DataTap, you may try accessing the namenode of the remote HDFS storage resource directly.

You may view DataTap configuration using the **Edit DataTap** screen, as described in Editing an Existing DataTap.

This image shows a sample central `dtap://remotedata/` configuration:



To test the configuration, log into any node in the Hadoop compute cluster and execute the `kinit` command to create a KDC session. The user you are logged in as must be able to be authenticated against either the `CORP.ENTERPRISE.COM` or the `CP.ENTERPRISE.COM` KDC realms. Once the `kinit` completes successfully, you should be able to access the namenode of the remote HDFS storage resource directly, without involving the deployment CNODE service, by executing the command `bluedata-1 ~]$ hdfs dfs -ls hdfs://hdfs.datalake.enterprise.com/`.

- If this command completes successfully, then test accessing the namenode of the remote HDFS file system via the deployment CNODE service and DataTap by executing the command `bluedata-1 ~]$ dfs -ls dtap://remotedata/`.
- If either of these commands fails, then there is an error in the KDC/HDP/HDFS configuration that must be resolved.

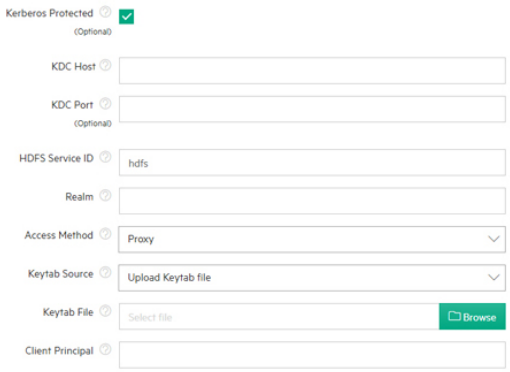
The following commands enable HDFS client debugging. Execute these commands before executing the `hdfs dfs -ls` command in order to log additional output:

```
export HADOOP_ROOT_LOGGER=DEBUG,console
export HADOOP_OPTS="-Dsun.security.krb5.debug=true -
Djavax.net.debug=ssl"
```

HDFS DataTap Kerberos Security

NOTE: This article only applies to HDFS DataTaps.

DataTaps that reference Kerberos-protected HDFS services are supported.



To configure a DataTap for Kerberos-protected HDFS:

1. If you are either adding a new Kerberos-protected DataTap with Proxy access mode for the first time or editing an existing DataTap with Proxy access mode and changing the Kerberos principal name then proceed to Step 2; otherwise, skip to Step 4.

2. Add the unique Kerberos principal name that will be used to register the DataTap (such as `bluedata`) as a super user by adding the following code snippet to the `core-site.xml` file of the remote HDFS:

```
<property>
  <name>hadoop.proxyuser.bluedata.groups
  </name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.bluedata.hosts
  </name>
  <value>*</value>
</property>
```

3. Restart the remote HDFS for the new configuration to take effect.

4. Set the permissions of the base HDFS directory to `777` and then open the web interface for the remaining steps.

5. Check the **Kerberos Protected** check box.

6. Enter the following parameters:

   - **KDC Host:** Name or IP address of the Kerberos hosts. You may enter multiple hosts separated by commas. If you enter more than one host, then the first host in the list as the primary Kerberos host. If the primary host is unreachable, then another host will be used.

   - **KDC Port:** Post used by the Kerberos hosts. Leave this field blank if not known. If you enter a value in this field, then all of the Kerberos hosts must use the same port.

   - **HDFS Service ID:** Name of the service, as defined by your Kerberos administrator. This is optional; if you leave this field blank, then HPE Ezmeral Runtime Enterprise will automatically detect the HDFS service ID.

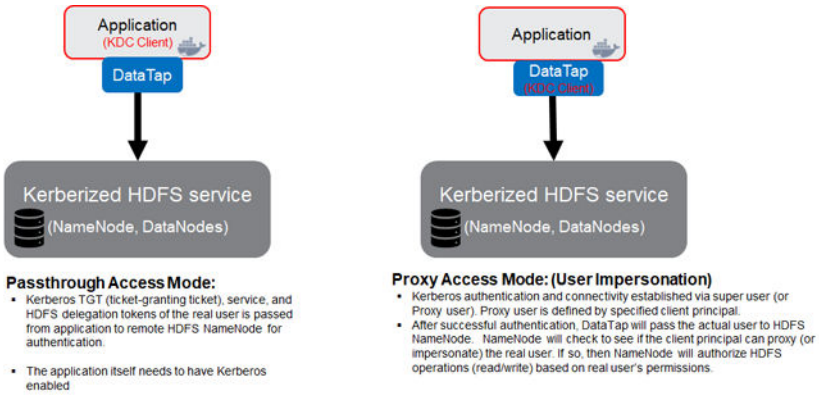
- **Realm:** Name space that helps define access permissions. Obtain this from your Kerberos administrator.

- **Access Method:** Select either **Passthrough** or **Proxy**, as appropriate.

  Selecting **Proxy** passes the specified client principal's credential to the namenode for authentication. In this case, the name of the real user who is accessing the DataTap from within a virtual node is also passed to the namenode for authorization.

  Selecting **Passthrough** passes the credentials of the user who is accessing the DataTap from within a virtual node to the namenode for authentication and authorization. In this case, the virtual cluster needs to be kerberized and the application (kubeadm-dind-cluster (KDC) client) must have Kerberos enabled.

  In both cases, the namenode authorizes the access based on the real user.



  If you select **Proxy**, then Proceed to Step 7.

  If you select **Passthrough**, then skip to Step 8.

7. Enter the following information:

- **Keytab Source:** Use this pull-down menu to select either **Upload Keytab File** or **Use Existing**, as appropriate.

- **Keytab File:** If you need to upload a keytab file, then place this file on your local computer and then click the **Browse** button in the **Keytab File** field to browse to the file and securely upload it . If you need to use a keytab file that was previously uploaded either via the interface or manually, then enter the name of that file in the **Keytab File** field.

- **Client Principal:** This is a unique identity to which Kerberos can assign tickets (such as bluedata). Enter the appropriate value in this field.

Skip to Step 9.

NOTE: Your organization security policies may not allow you to upload keytab files via the web interface. If you need to manually upload keytab files, then place keytab files used for local HDFS tenant storage in the `/srv/bluedata/keytab/site_admin` directory on the Controller node. Keytabs used in DataTap definitions are in subdirectories associated with the tenant ID, such as `/srv/bluedata/keytab/3`.

8. Enter the following information:

- **Use Keytab File for Browsing:** Use this pull-down menu to select either **Yes** or **No**, as appropriate. If you select **Yes**, then enter the following information. The proxy option only applies when users are accessing the DataTap from directly within a virtual node. If you want this DataTap to be available to users who are accessing the web interface, then you will need to select **Yes** and provide all of the following information. In this case, DataTap access will function as a passthrough when the DataTap is accessed from the web interface and will pass individual user credentials when the user is accessing the virtual node directly. If you select **No**, then the DataTap will not be available from within the web interface; skip to Step 9.

- **Keytab Source:** Use this pull-down menu to select either **Upload Keytab File** or **Use Existing**, as appropriate.

- **Keytab File:** If you need to upload a keytab file, then place this file on your local computer and then click the **Browse** button in the **Keytab File** field to browse to the file and securely upload it . If you need to use a keytab file that was previously uploaded either via the interface or manually, then enter the name of that file in the **Keytab File** field.

- **Client Principal:** This is a unique identity to which Kerberos can assign tickets (such as bluedata). Enter the appropriate value in this field.

9. Continue creating or editing the DataTap, as appropriate. See Creating a New DataTap and Editing an Existing DataTap.

NOTE: To disable Kerberos protection, clear the appropriate **Kerberos Protected** check box(es) and then click **Submit**.

NOTE: If you need to configure passthrough DataTap authentication across multiple Kerberos realms, then please see HDFS DataTap Cross-Realm Kerberos Authentication.

HDFS DataTap TDE Configuration

NOTE: This article only applies to HDFS DataTaps.

Transparent Data Encryption (TDE) provides end-to-end data encryption between virtual clusters and HDFS storage resources. This encryption and decryption are transparent, because no changes are required to the application code. Only the virtual cluster can encrypt and decrypt this data; the storage resource never stores nor accesses unencrypted data or the keys required to decrypt that data. This means that data is encrypted both when it is at rest (residing on storage media such as a disk) and in transit (being transmitted across a network). DataTaps handle TDE because encrypting and decrypting data is computationally intensive, and this method will only affect the container that accesses the TDE-enabled DataTap.

A virtual cluster that will use TDE-enabled DataTaps must be Kerberized, because the DataTap uses Kerberos authentication when communicating with the Key Management Service (KMS).

Enabling TDE requires several configuration changes to the remote HDFS storage resource, including:

- Installing and configuring a KMS, including the Access Control List (ACL) and SSL.

- Configuring the remote HDFS storage resource to use the KMS.

- Creating an encryption key and encryption zone on the remote HDFS storage resource.

The instructions in this article assume that the remote HDFS storage resource and KMS have been correctly configured before proceeding to create and configure the DataTap. Please see Sample TDE Configuration, below, for a sample CDH-based HDFS and KMS) configuration.

On the HPE Ezmeral Runtime Enterprise side, enabling and supporting TDE requires the following configuration updates to the virtual cluster itself:

- **KMS URL:** The DataTap and HDFS client use this information to locate the KMS.
- **Truststore:** The DataTap and HDFS client use this information to authenticate with the KMS server because the protocol is based on HTTPS.

NOTE: The DataTap must be configured in passthrough mode (see HDFS DataTap Kerberos Security) in order to enable TDE.

Please see the appropriate section below for instructions on configuring a virtual cluster:

- **CDH clusters:** See TDE Configuration for Cloudera Clusters.
- **HDP clusters:** See TDE Configuration for Hadoop Clusters.

Configuring Cloudera Clusters for TDE

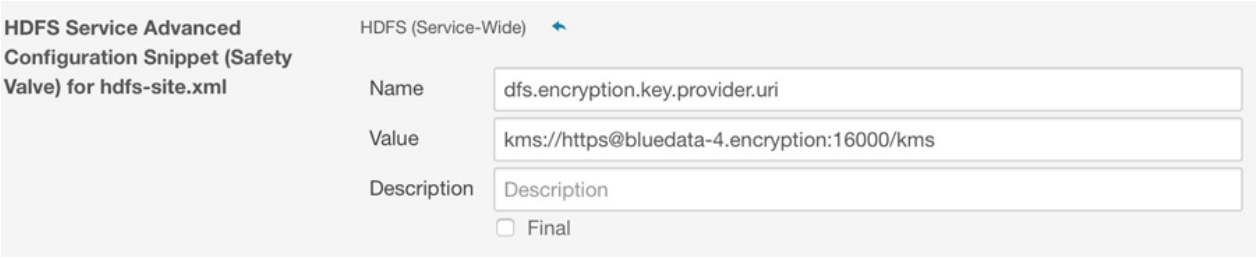
Configuring Cloudera clusters for TDE is a two-phase process:

- **KMS URL:** See Phase 1: Configuring the KMS URL (CDH).
- **Truststore:** See Phase 2: Configuring the Truststore (CDH).

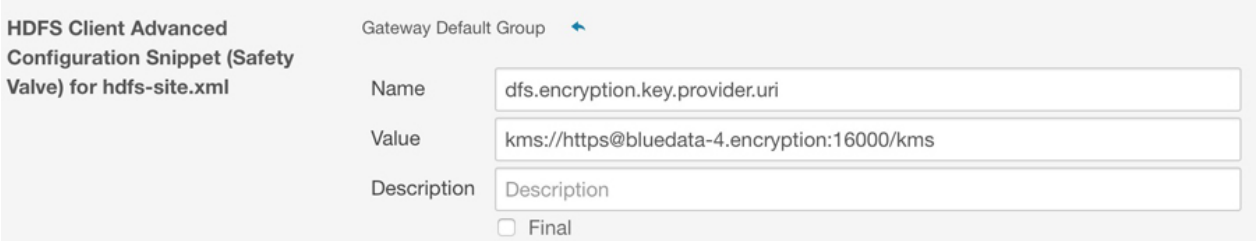
Phase 1: Configuring the KMS URL (CDH)

To configure the KMS URL for a Cloudera virtual cluster:

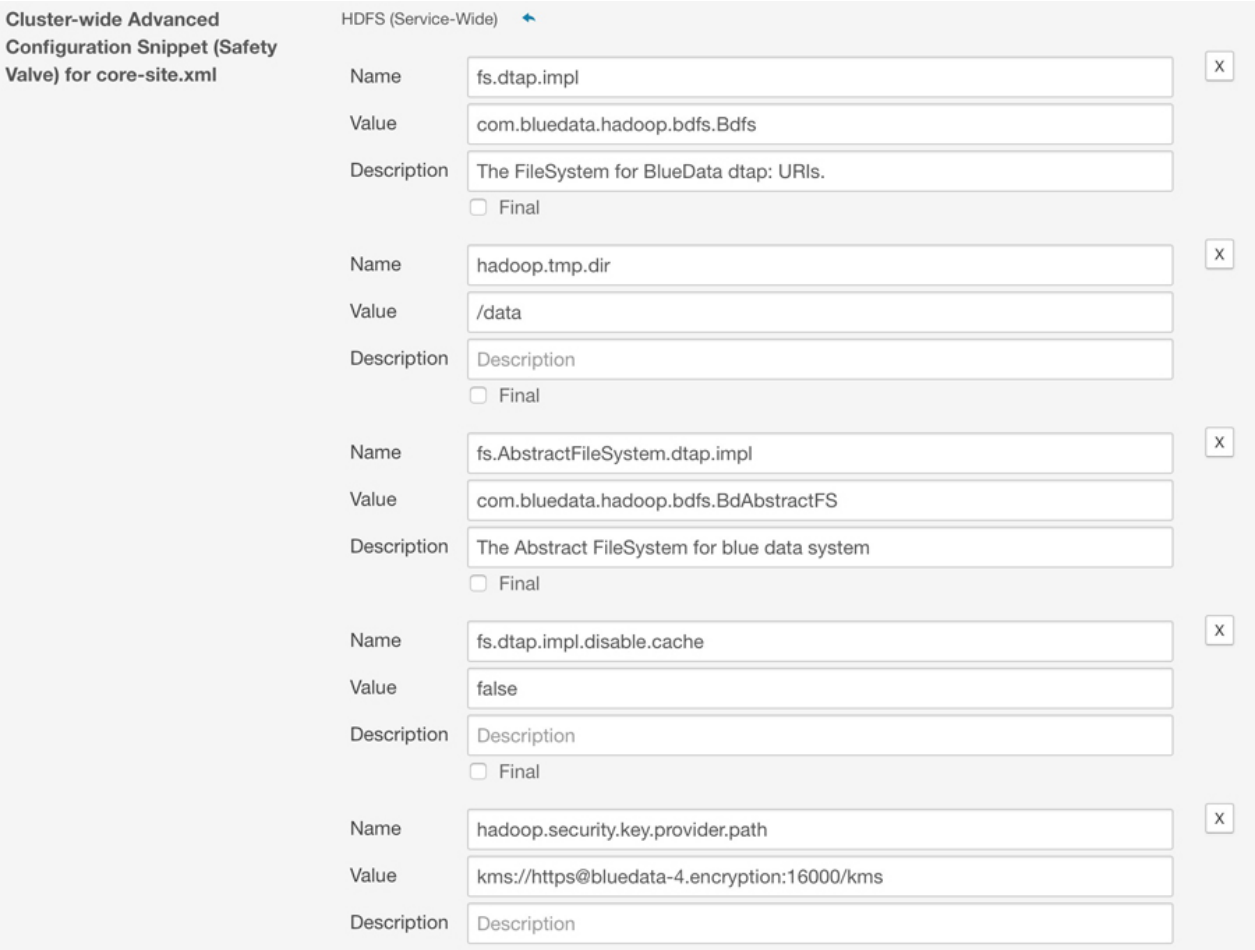
1. In the remote HDFS storage resource, add the `dfs.encryption.key.provider.uri` property to the following:
   - **HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml**.



   - **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml**.



2. In the remote HDFS storage resource, add the `hadoop.security.key.provider.path` property is added to **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml**.



Phase 2: Configuring the Truststore (CDH)

To configure the Truststore for a Cloudera virtual cluster:

1. Verify that the certificate file for the KMS server is ready. This example assumes that the certificate file is named `selfsigned.cer`.

2. Execute the following commands to import the certificate into the truststore:

```
cp /usr/java/jdk1.7.0_67-cloudera/jre/lib/security/cacerts /usr/java/
jdk1.7.0_67-cloudera/jre/lib/security/jssecacerts
/usr/java/jdk1.7.0_67-cloudera/jre/bin/keytool -import -alias
kmshost -file /opt/cloudera/security/jks/selfsigned.cer -keystore
/usr/java/jdk1.7.0_67-cloudera/jre/lib/security/jssecacerts -storepass
changeit
```

3. Copy the truststore file (named `jssecacerts` in this example) to all of the virtual nodes/containers in the HPE Ezmeral Runtime Enterprise virtual cluster. The path to the truststore file must be identical on all nodes.

4. In the remote HDFS storage resource, select **HDFS>Configs>Advanced**.

   The **Advanced** tab appears.

5. Modify the `ssl.client.truststore.location` and `ssl.client.truststore.password` properties.

NOTE: If the `ssl.client.truststore.location` property is not configured for a Cloudera virtual cluster, then the Oracle JDK will search for the `/usr/java/jdk1.7.0_67-cloudera/jre/lib/security/jssecacerts` file by default. This means that you can ignore the `ssl.client.truststore.location` and `ssl.client.truststore.password` properties if you are using this default configuration.

Configuring Hadoop Clusters for TDE

Configuring Hadoop clusters for TDE is a two-phase process:

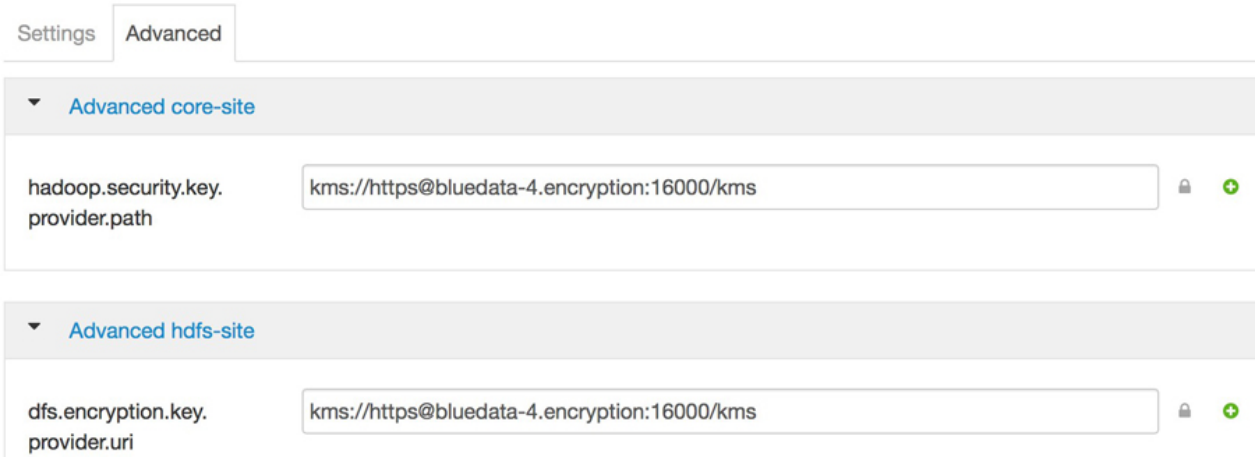
- **KMS URL:** See Phase 1: Configuring the KMS URL (HDP).
- **Truststore:** See Phase 2: Configuring the Truststore (HDP).

Phase 1: Configuring the KMS URL (HDP)

To configure the KMS URL for a Hadoop virtual cluster:

1. In the remote HDFS storage resource, select **HDFS>Configs>Advanced**.

   The **Advanced** tab appears.



2. Modify the `hadoop.security.key.provider.path` property in the **Advanced core-site** section.
3. Modify the `dfs.encryption.key.provider.uri` property in the **Advanced hdfs-site** section.

Phase 2: Configuring the Truststore (HDP)

To configure the Truststore for a Cloudera virtual cluster:

1. Verify that the certificate file for the KMS server is ready. This example assumes that the certificate file is named `selfsigned.cer`.

2. Execute the following commands to import the certificate into the truststore:

```
cp /usr/java/jdk1.7.0_67-cloudera/jre/lib/security/cacerts /usr/java/
jdk1.7.0_67-cloudera/jre/lib/security/jssecacerts
/usr/java/jdk1.7.0_67-cloudera/jre/bin/keytool -import -alias
kmshost -file /opt/cloudera/security/jks/selfsigned.cer -keystore
/usr/java/jdk1.7.0_67-cloudera/jre/lib/security/jssecacerts -storepass
changeit
```

3. Copy the truststore file (named `jssecacerts` in this example) to all of the virtual nodes/containers in the HPE Ezmeral Runtime Enterprise virtual cluster. The path to the truststore file must be identical on all nodes.

4. In the remote HDFS storage resource, select **HDFS>Configs>Advanced**.

   The **Advanced** tab appears.

5. Modify the `ssl.client.truststore.location` and `ssl.client.truststore.password` properties.

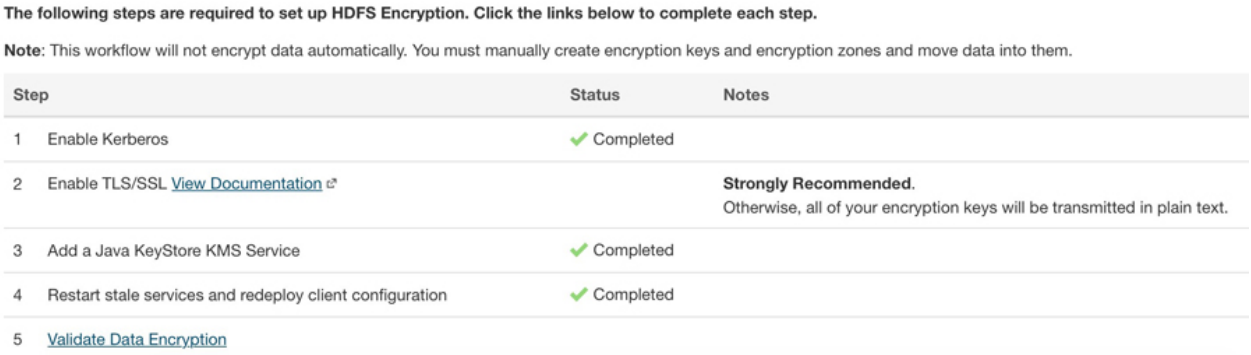
Configuration Example

This example demonstrates how to configure a sample remote HDFS storage device and KMS for use with a Cloudera virtual cluster. To do this:

1. Kerberize all of the virtual nodes/containers in the Cloudera virtual cluster.

2. In the remote HDFS storage resource, select **HDFS>Actions>Set up HDFS Data At Rest Encryption**.

3. Follow the listed steps to enable TDE. The key goal here is to key point is to generate the keystore for the KMS server to enable HTTPS. This example uses a self-signed certificate for simplicity.



The following steps are required to set up HDFS Encryption. Click the links below to complete each step.

Note: This workflow will not encrypt data automatically. You must manually create encryption keys and encryption zones and move data into them.

Step		Status	Notes
1	Enable Kerberos	Completed	
2	Enable TLS/SSL View Documentation		**Strongly Recommended.** Otherwise, all of your encryption keys will be transmitted in plain text.
3	Add a Java KeyStore KMS Service	Completed	
4	Restart stale services and redeploy client configuration	Completed	
5	Validate Data Encryption		

4. Execute the following command on the node that hosts the Java KeyStore KMS service. (This example uses a cn of `bluedata-4.encryption` that should be replaced by the actual FQDN):

```
/usr/java/jdk1.7.0_67-cloudera/jre/bin/keytool -genkeypair -alias
kmshost -keyalg RSA -keysize 2048
        -dname "cn=bluedata-4.encryption, ou=EN, o=BD, l=SC,
st=CA, c=US" -keypass password -keystore kmshost-keystore.jks -storepass
password
```

NOTE: The keypass and storepass must be the same.

5. Copy the generated keystore file to `/opt/cloudera/security/jks/kmshost-keystore.jks`.

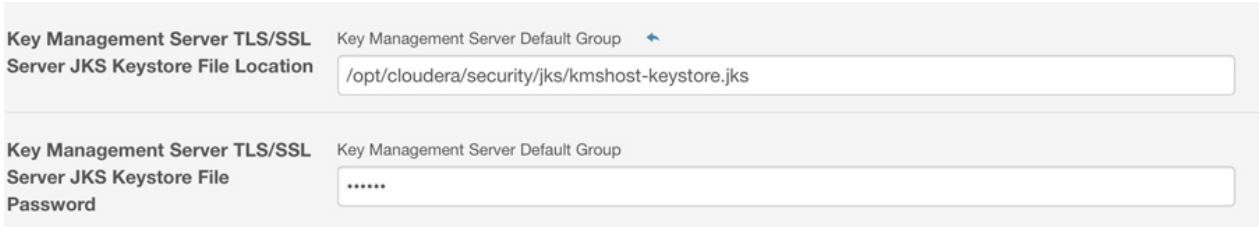
NOTE:

You can execute the following command to export the KMS certificate and then use that certificate generate the KMS client truststore:

```
/usr/java/jdk1.7.0_67-cloudera/jre/bin/keytool -export -alias
kmshost -keystore kmshost-keystore.jks -rfc -file selfsigned.cer
```

.

6. Based on the generated keystore file, configure TLS/SSL as shown here:



This procedure configures the KMS ACL, which will appear similar to the following:

```
<property>
  <name>hadoop.kms.acl.CREATE</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.DELETE</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.ROLLOVER</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.GET</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.acl.GET_KEYS</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.GET_METADATA</name>
  <value>hdfs supergroup</value>
</property>
<property>
```

```
  <name>hadoop.kms.acl.SET_KEY_MATERIAL</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.acl.GENERATE_EEK</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.acl.DECRYPT_EEK</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.blacklist.CREATE</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.blacklist.DELETE</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.blacklist.ROLLOVER</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.blacklist.GET</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.kms.blacklist.GET_KEYS</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.blacklist.SET_KEY_MATERIAL</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.kms.blacklist.DECRYPT_EEK</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>default.key.acl.MANAGEMENT</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.GENERATE_EEK</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.DECRYPT_EEK</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.READ</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.MIGRATE</name>
  <value></value>
</property>
<property>
  <name>whitelist.key.acl.MANAGEMENT</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
```

```
  <name>hadoop.kms.acl.CREATE</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.DELETE</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.ROLLOVER</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.GET</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.acl.GET_KEYS</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>hadoop.kms.acl.GET_METADATA</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.acl.SET_KEY_MATERIAL</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.acl.GENERATE_EEK</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.acl.DECRYPT_EEK</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.blacklist.CREATE</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.blacklist.DELETE</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.blacklist.ROLLOVER</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>hadoop.kms.blacklist.GET</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.kms.blacklist.GET_KEYS</name>
  <value></value>
</property>
<property>
  <name>hadoop.kms.blacklist.SET_KEY_MATERIAL</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.kms.blacklist.DECRYPT_EEK</name>
  <value>hdfs supergroup</value>
</property>
<property>
```

```
  <name>default.key.acl.MANAGEMENT</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.GENERATE_EEK</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.DECRYPT_EEK</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.READ</name>
  <value></value>
</property>
<property>
  <name>default.key.acl.MIGRATE</name>
  <value></value>
</property>
<property>
  <name>whitelist.key.acl.MANAGEMENT</name>
  <value>xou,kishore xou,kishore</value>
</property>
<property>
  <name>whitelist.key.acl.READ</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>whitelist.key.acl.GENERATE_EEK</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>whitelist.key.acl.DECRYPT_EEK</name>
  <value>xou,kishore,yarn,nm xou,kishore,yarn,nm</value>
</property>property>
  <name>whitelist.key.acl.READ</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>whitelist.key.acl.GENERATE_EEK</name>
  <value>hdfs supergroup</value>
</property>
<property>
  <name>whitelist.key.acl.DECRYPT_EEK</name>
  <value>xou,kishore,yarn,nm xou,kishore,yarn,nm</value>
</property>
```

NOTE: In order to run a job, the YARN user of a Cloudera cluster and the NM user of a Hadoop cluster must have the DECRYPT_EEK privilege in order to access files in the encryption zone if the source/destination files of the job are located at the encryption zone.

Validation

To validate the TDE configuration:

1. In the virtual cluster, execute the command `hadoop key list` to verify the communication between the cluster and KMS, as shown here:

```
                         [bluedata@bluedata-1 ~]$ hadoop key list
                         Listing keys for KeyProvider:
KMSClientProvider[https://bluedata-4.encryption:16000/kms/v1/]
                         mykey1
```

2. In the virtual cluster, execute the command `openssl s_client -connect host.fqdn.name:port` to check TLS/SSL negotiation. The output will appear similar to the following if the test is successful:

```
                         [bluedata@bluedata-1 ~]$ openssl s_client -connect
bluedata-4.encryption:16000
                         CONNECTED(00000003)
                         depth=0 C = US, ST = CA, L = SC, O = BD, OU = EN,
CN = bluedata-4.encryption
                         verify error:num=18:self signed certificate
                         verify return:1
                         depth=0 C = US, ST = CA, L = SC, O = BD, OU = EN,
CN = bluedata-4.encryption
                         verify return:1
                         ---
                         Certificate chain
                         0 s:/C=US/ST=CA/L=SC/O=BD/OU=EN/
CN=bluedata-4.encryption
                            i:/C=US/ST=CA/L=SC/O=BD/OU=EN/
CN=bluedata-4.encryption
                         ---
                         Server certificate
                         -----BEGIN CERTIFICATE-----

MIIDYTCCAkmgAwIBAgIEQDnyMzANBgkqhkiG9w0BAQsFADBhMQswCQYDVQQGEwJV
UzELMAkGA1UECBMCQ0ExCzAJBgNVBAcTAlNDMQswCQYDVQQKEwJCRDELMAkGA1UE
CxMCRU4xHjAcBgNVBAMTFWJsdWVkYXRhLTQuZW5jcnlwdGlvbjAeFw0xNzA3MTgx
NzExMDRaFw0xNzEwMTYxNzExMDRaMGExCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJD
QTELMAkGA1UEBxMCU0MxCzAJBgNVBAoTAkJEMQswCQYDVQQLEwJFTjEeMBwGA1UE
AxMVYmx1ZWRhdGEtNC5lbmNyeXB0aW9uMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAivCHpkQfYy88Fg8dLnA5E3JJ4i9R7FRhi6zmNx9k+SI/QZLEERZ2
DJPUtvfvABHsSM9eSUSMGay6yYdwAjvrBaBI1Nwvcl+Sq2q+1kbcFf80F09b3oe0
2Ac3TyOlDVYWkXYquQFjsExMWJD32cgohrmhHzjU/zomxDO1Yltko4s7Bq+2jR9D
w6PLMhno4qgtItqTeUqCqQg/iUdGVbdxWnXIFCZtMxIMZBub6vXsi8s2rnRi8PU5
IgmfO4HCqw84VNgKU5Z5i71wm7ZPJXM6Atb+fd/3TKvuY76dcz+YjSBOmBqn2Brm
IkMYwOtOtXFQs4BHPZPlsPfLHeTBQy+LMwIDAQABoyEwHzAdBgNVHQ4EFgQUK92j
s0W3FVtiB6G2MpKnmVI6mK0wDQYJKoZIhvcNAQELBQADggEBACavBuJ8n033GGjv
oElJ+2FEjEItfci0dY50TCkKTlSJilLpVGOaWgqNAS6sD5qnodOQ5XhQ+smawNF4 XZ1zjhlN/
AzwEInATvIgIICDgxKg30TWI5cJZ+Rr2fErr3SO1EPh8azsVy38UbjB /
TtzrN4VWK+NeYZddGfo5SMyxSMAN2vf6Sn3Cll/spmDQCR9fXqQrNt/McDfm1rK
BASWCAnMe0OQafXR9eYgy1mtSnP5KQc1A2rqK6oZC7tv+qiZtk0jfh4bAlWHgLOt X/
yZRF4f49bdP7NioR9KsMnxc20JjwaDpYdyXK3b4U36/lphksllM4jCiGUvlcXI B/g+k1E=
                         -----END CERTIFICATE-----
                         subject=/C=US/ST=CA/L=SC/O=BD/OU=EN/
CN=bluedata-4.encryption
                         issuer=/C=US/ST=CA/L=SC/O=BD/OU=EN/
CN=bluedata-4.encryption
                         ---
                         No client certificate CA names sent
                         Server Temp Key: ECDH, secp521r1, 521 bits
                         ---
                         SSL handshake has read 1477 bytes and written 497
bytes
                         ---
                         New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-SHA384
                         Server public key is 2048 bit
                         Secure Renegotiation IS supported
                         Compression: NONE
                         Expansion: NONE
                         SSL-Session:
                             Protocol  : TLSv1.2
                             Cipher    : ECDHE-RSA-AES256-SHA384
                             Session-ID:
```

```
598B79F355B64A6106A82E735689E44F570DD6926B41082DDCD9E89B0E8CC49E
                            Session-ID-ctx:
                            Master-Key:
70000BD0F41E60933EACB912446AFD4C2F7A83E43444FEE1D989DB6D446A57B9D860BDAE6CE
31BBAA4A498847C437FDD
                            Key-Arg   : None
                            Krb5 Principal: None
                            PSK identity: None
                            PSK identity hint: None
                            Start Time: 1502312947
                            Timeout   : 300 (sec)
                            Verify return code: 18 (self signed
certificate)
                        ---
                        &H94;C
                        [bluedata@bluedata-1 ~]$
```

3. In the virtual cluster, execute the following commands to output debugging information:

```
export HADOOP_ROOT_LOGGER=DEBUG,console
export HADOOP_OPTS="-Dsun.security.krb5.debug=true -Djavax.net.debug=ssl"
```

HDFS DataTap Wire Encryption

NOTE: This article only applies to HDFS DataTaps.

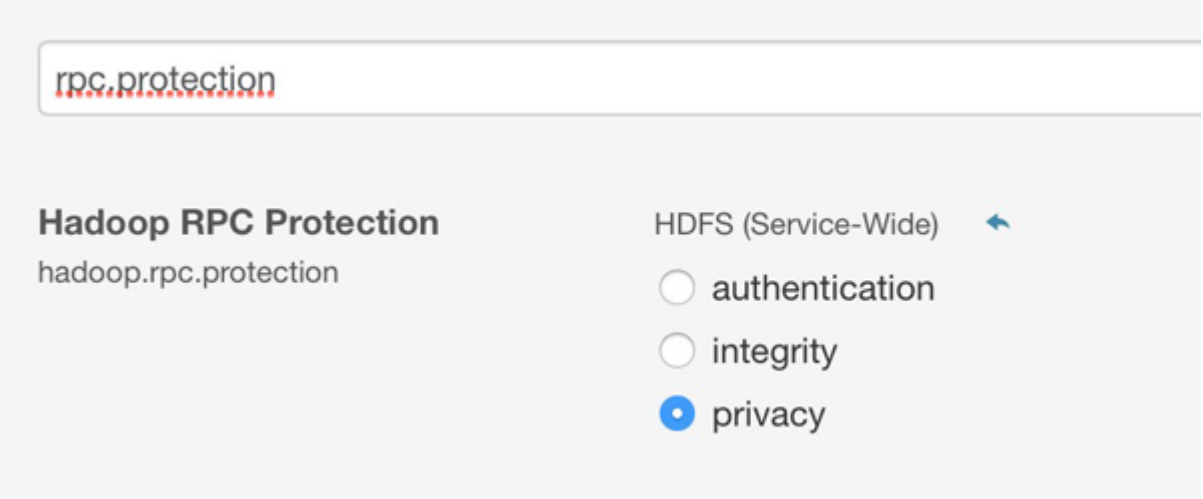
Wire encryption means that the network packets between virtual node and remote HDFS service are encrypted. This includes:

- **RPC encryption:** the RPC messages between the virtual node(s) and the HDFS namenode are encrypted.
- **Data Transfer encryption:** The control message and data between the virtual node(s) and the HDFS data nodes are encrypted.

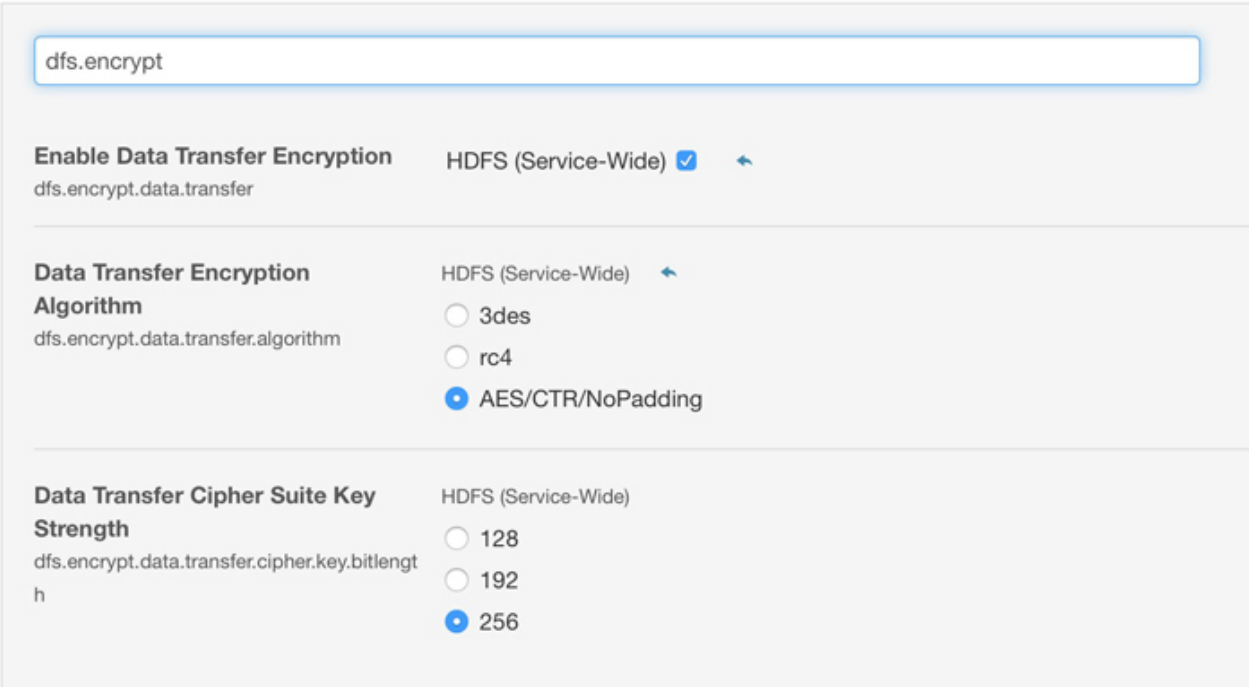
No additional configuration is required to support this feature; however, the HDFS configurations must be modified to enable the wire encryption. Further, the remote HDFS must be Kerberized for security.

To enable wire encryption on a CDH HDFS service:

1. Enable RPC encryption on the remote HDFS service using the CDH Manager interface, as shown here.



2. Enable data transfer encryption on the remote HDFS service using the CDH Manager interface, as shown here.



3. Restart the remote HDFS service.

To enable wire encryption on an HDP HDFS service:

1. In the Ambari interface, enable RPC encryption by selecting **HDFS>Configs>Advanced>Custom core-site**, and then adding `hadoop.rpc.protection = privacy`, as shown here.

2. In the Ambari interface, enable Data Transfer encryption by selecting **HDFS>Configs>Advanced>Custom hdfs-site**, add then adding `dfs.encrypt.data.transfer = true`, as shown here.

3. Restart the remote HDFS service.

NOTE: Currently the `dfs.encrypt.data.transfer.algorithm` supports AES, CTR, or NoPadding, and the `dfs.encrypt.data.transfer,cipher.key.bitlength` can support 128, 192, or 256 bits..

Kubernetes Cluster Administrator Tasks

The topics in this section describe information and tasks that Kubernetes Cluster Administrators can perform in HPE Ezmeral Runtime Enterprise.

Dashboard - Kubernetes Cluster Administrator

Users who are logged into a Kubernetes tenant with the Cluster Administrator role can access the Kubernetes Cluster Administrator **Dashboard** screen by selecting **Dashboard** in the main menu.

The top of this screen has three buttons that allows you to download the plugins that you need to access Kubernetes pods within a cluster. The buttons are:

- **Download HPE Kubectl Plugin:** Downloads the HPE installer for the `kubectl` command line tool for controlling a Kubernetes cluster. For more information about kubectl, see Command line tool (kubectl) (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl, and Using the HPE Kubectl Plugin.

- **Download Kubectl:** Downloads the generic installer for the `kubectl` command line tool for managing a Kubernetes cluster. Please click here for more information (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl.

NOTE: You might see a warning that `kubectl-hpecp` cannot be opened because the publisher cannot be verified. You can safely ignore this warning and proceed with the installation.

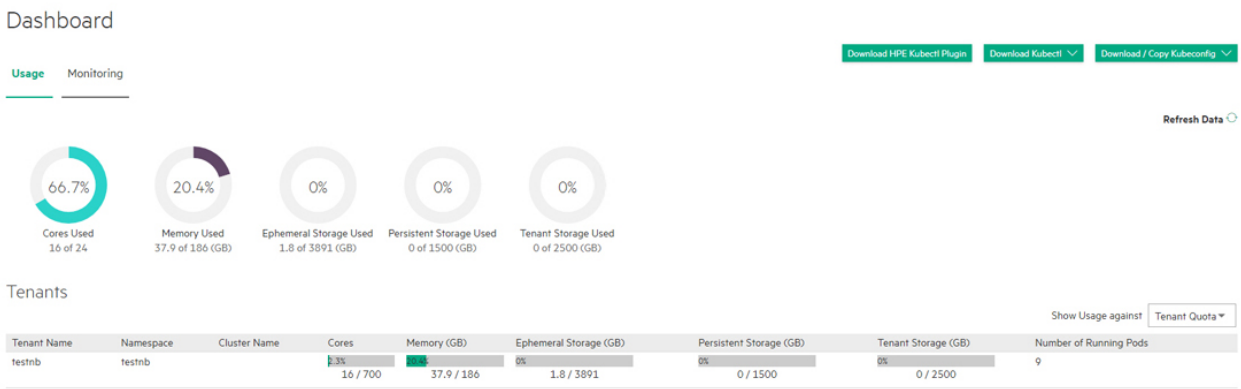
- **Download Kubeconfig:** Downloads the `kubeconfig` file that contains information to configure access to Kubernetes when used in conjunction with either the `kubectl` command line tool or other clients. For more information about the `kubeconfig` file, see Organizing Cluster Access Using kubeconfig Files (link opens an external website in a new browser tab/window).

This screen has the following tabs:

Usage	Displays resource usage on a cluster-wide and tenant-by-tenant basis. See Usage Tab.
Monitoring	Provides detailed resource monitoring. See Monitoring Tab.

Usage Tab

The **Usage** tab displays usage statistics for the current Kubernetes cluster.



The top of the **Usage** tab displays dials showing the following aggregate information for the cur:

- **Cores Used:** Percentage of available virtual CPU cores being used by all of the tenants in the deployment.

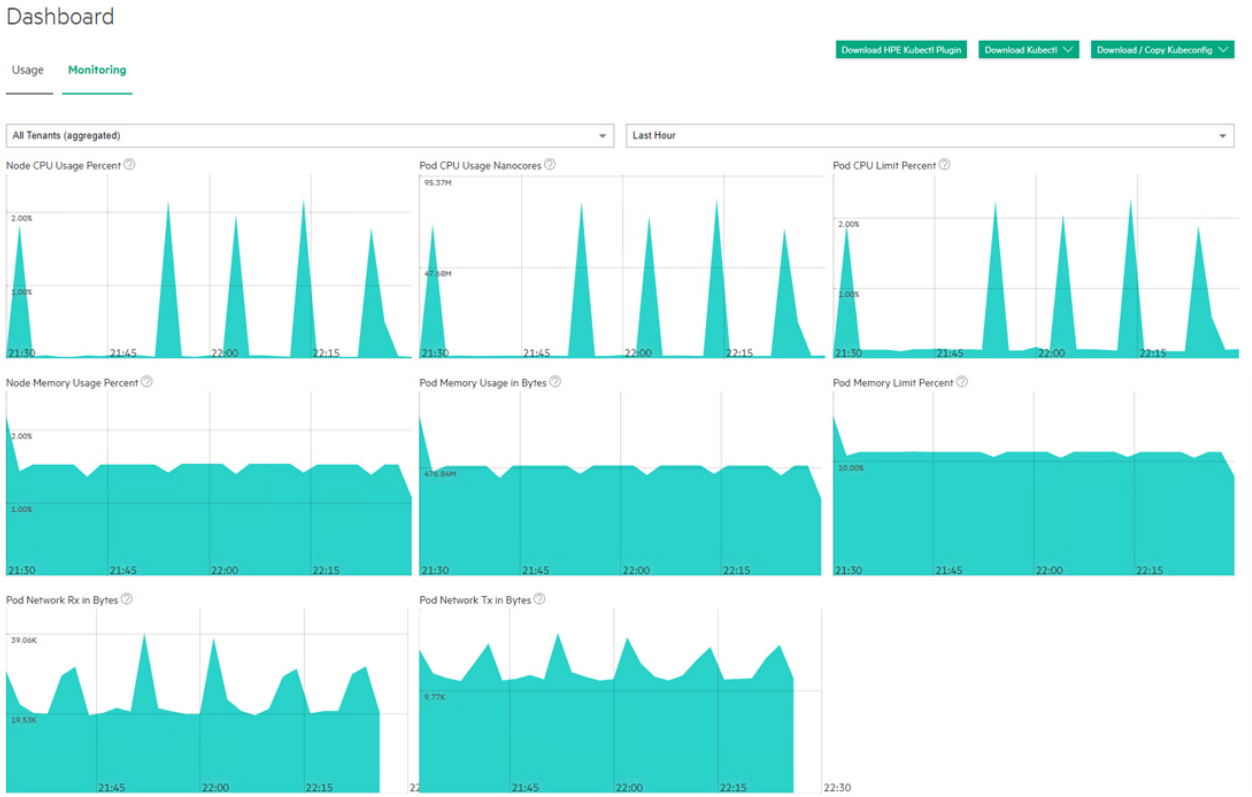
- **Memory Used (GB):** Percentage of available RAM being used by all of the tenants in the deployment.
- **Ephemeral Storage Used (GB):** Percentage of available ephemeral storage used and the total available persistent storage, in GB.
- **Persistent Storage Used (GB):** Percentage of available persistent storage used and the total available persistent storage, in GB.
- **Tenant Storage Used (GB):** Percentage of available tenant storage used and the total available persistent storage, in GB.
- **GPU Utilization (percent):** If GPUs are present, displays aggregate GPU utilization in percent.
- **GPU Memory Usage:** If GPUs are present, displays aggregate GPU memory usage in percent.

The bottom of this tab contains a table that lists all of the tenants within the current Kubernetes cluster. This table displays the **Tenant Name**, **Namespace**, **Cluster Name**, **Cores**, **Memory (GB**), **Ephemeral Storage**, **Persistent Storage**, **Tenant Storage**, and the **Number of Running Pods**. This data can be expressed against either the Tenant Quota or total System Resources, depending on your **Show Usage against** menu selection.

NOTE: For information about how to download detailed usage and uptime information in comma-delimited (.csv) format, see Downloading Kubernetes Usage Details.

Monitoring Tab

The **Monitoring** tab displays resource usage over time.



The top of this screen has three buttons that allows you to download the plugins that you need to access Kubernetes pods within a cluster. The buttons are:

- **Download HPE Kubectl Plugin:** Downloads the HPE installer for the `kubectl` command line tool for controlling a Kubernetes cluster. Please click here for more information (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl, below.
- **Download Kubectl:** Downloads the generic installer for the `kubectl` command line tool for controlling a Kubernetes cluster. Please click here for more information (link opens an external website in a new browser tab/window). You will need to install this application. See Installing Kubectl, below.
- **Download Kubeconfig:** Downloads the `kubeconfig` file that configures access to Kubernetes when used in conjunction with either the `kubectl` command line tool or other clients. Please click here for more information (link opens an external website in a new browser tab/window).

The top of this screen has three pull-down menus that allow you to filter the data by tenant, pod, and time frame. You may also choose to view information for all applications or only for KubeDirector applications by moving the **Filter KubeDirector Applications** slider. Hovering your mouse over the graphs displays a popup with additional information. The following charts are available:

- **Node CPU Usage Percent:** Percentage of available CPU resources in use.
- **Pod CPU Use Nanocores:** Number of CPU nanocores in use.
- **Pod CPU Pod Limit Percent:** Percentage of maximum number of pods that are currently running inside the current cluster.
- **Node Memory Usage Percent:** Percentage of available memory being used.
- **Pod Memory Usage in Bytes:** Bytes of memory being used.
- **Pod Memory Limit Percent:** Percentage of memory limit being used.
- **Pod Network Rx in Bytes:** Bytes received over the network.
- **Pod Network Tx in Bytes:** Bytes transmitted over the network.

NOTE: For information about how to download detailed usage and uptime information in comma-delimited (.csv) format, see Downloading Kubernetes Usage Details.

Installing Kubectl

To install Kubectl on your local system:

1. Download both of the Kubectl plugins by clicking the **HPE Kubectl Plugin** and **Kubectl** buttons:
   - If you are on a Windows system, then these downloads will be .exe files.
   - If you are on a MacOS or UNIX system, then you will need to execute the following commands:

     ```
     chmod +x kubectl-hpecp
     ```

     ```
     chmod +x kubectl
     ```

2. Place both executables into a folder that is on your system's PATH. You can find the folders in your system's PATH by executing the appropriate command:
   - **Windows:** `ECHO %PATH%`
   - **MacOS or Linux:** `echo $PATH`

3. Execute the command `kubectl hpecp refresh {HPE Ezmeral Runtime Enterprise controller/gateway ip address}`. If HTTPS is not enabled, then add the argument `--insecure=true`.

Toolbar & Main Menu - Kubernetes Cluster Administrator

This article describes the UI items for Kubernetes Cluster Administrators.

Toolbar

The layout of the Toolbar is the same as described in Navigating the GUI on page 143. For information about the content of the **Quick Access** menu for Kubernetes Cluster Administrators, see Quick Access Menu - Kubernetes Cluster Administrator on page 435.

Main Menu - Kubernetes Cluster Administrator

The main menu for Kubernetes Cluster Administrators appears as shown in the following image:

For Kubernetes Cluster Administrators, the **Main Menu** includes the following items. For information about performing the tasks associated with the screens you access from the main menu, see Kubernetes Cluster Administrator Tasks on page 432.

Dashboard	Opens the **Kubernetes Dashboard** screen.
Cluster	Opens the **Cluster Details** screen of the current Kubernetes cluster.
Tenants	Displays the number of Kubernetes tenants and opens the **Kubernetes Tenants** screen, which enables you to view information about tenants and projects and assign users to roles in the tenants or projects.
Users	Opens the **Kubernetes Cluster Users** screen, which enables you to view and manage the users assigned to the current Kubernetes cluster.

Quick Access Menu - Kubernetes Cluster Administrator

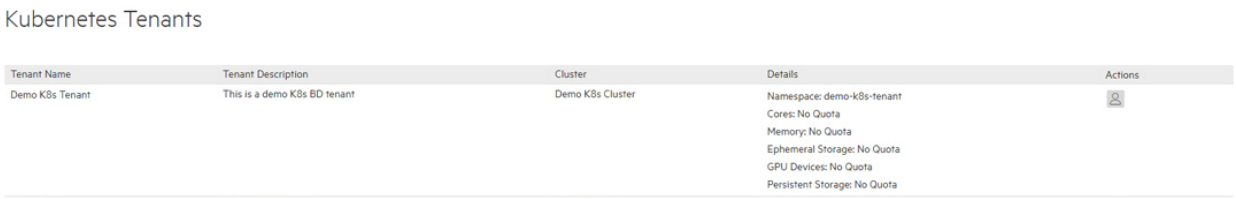
For Kubernetes Cluster Administrators, the following items appear in the **Quick Access** menu:

Create Tenant	Opens the **Create New Tenant** screen, which allows you to create a new tenant or AI/ML project.
Assign User	Opens the **Users Assignment** screen, which enables you to grant roles to users.

	See Viewing and Assigning Kubernetes Cluster Users on page 436.
User Info	Opens the **Current User Information** dialog, which lists your role, current project, and username.
User Guide	Opens this *User and Administrator Guide*.
Privacy	Opens the **Hewlett Packard Enterprise Privacy Statement** web page in a new browser tab or window.
Version	Displays version and build information about the HPE Ezmeral Runtime Enterprise deployment.

Viewing a Kubernetes Tenant or Project

Selecting **Tenants** in the main menu opens the **Kubernetes Tenants** screen, which allows you to view the tenants in the current Kubernetes cluster and assign user roles within this tenant.



The table on this screen contains the following information and functions:

- **Tenant Name:** Name of the tenant.
- **Tenant Description:** Brief description of the tenant.
- **Cluster:** Cluster to which this tenant belongs.
- **Details:** Detailed information about the tenant, including:
  - **Namespace:** Kubernetes tenant namespace.
  - **Quotas:** Cores, Memory, Node Storage, and GPU resource quotas assigned to the tenant compared to the total available resources in the system, such as **`Cores=8/16`**. If the tenant has no quota for a resource, then the display will show the resources being used (such as **`Cores=8`**) and the message **`No Quota`**.
- **Actions:** Clicking the **Users** icon (person) in the **Actions** column opens the **Tenant Users** screen for that tenant, which allows you to either assign and revoke user roles or delete a user. See Viewing and Assigning Kubernetes Tenant Users.

Viewing and Assigning Kubernetes Cluster Users

Selecting **Users** in the main menu opens the **Clusters Users** screen, which displays the users who are assigned to the current Kubernetes tenant.

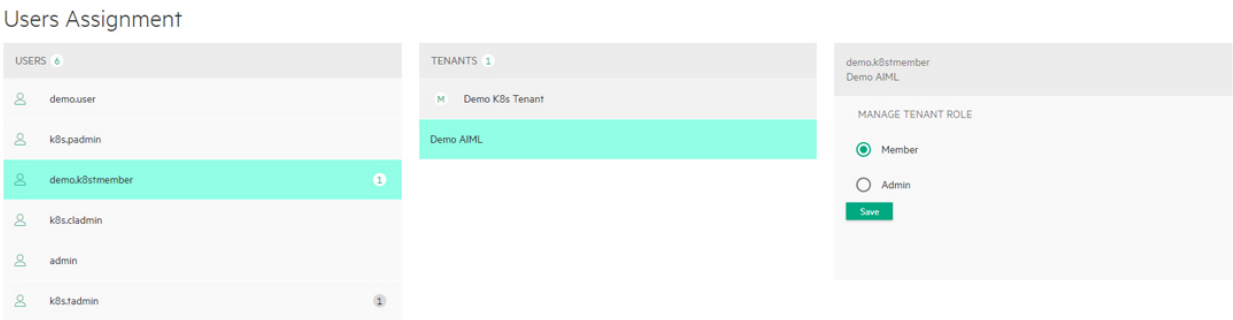
The top of the screen includes the **Assign** button. Clicking this button opens the **Users Assignment** screen, which allows you to assign users to the current Kubernetes tenant. See Assigning User Roles, below.

This screen displays the following information for each user who has a role in the current Kubernetes tenant:

- **Login Name:** Username of the user.
- **Full Name:** Full name of the user.
- **Role:** Role of the user within the current Kubernetes tenant. This will be either **Member** or **Admin**. See Users and Roles.
- **Actions**: Clicking the **Revoke** button for a user revokes their role from the current Kubernetes tenant. This does not affect any other role(s) the user may have.

Assigning User Roles

Clicking the **Assign** button in the **Users** screen opens the **Users Assignments** screen, which allows you to assign roles in the current Kubernetes tenant to users.



To assign a role to a user:

1. Select the user to whom you want to assign a role in the **USERS** column.
2. If the current cluster has more than one tenant, then select the tenant to which you want to assign the user in the **TENANTS** column.
3. Check the appropriate **MANAGE TENANT ROLE** radio button to assign the desired role.
   - Checking the **Member** radio button makes the selected user a Member of the current tenant.
   - Checking the **Admin** radio button makes the selected user a Tenant Administrator of the current Kubernetes tenant.
4. Click **Save**.

You may repeat this process for each additional user you want to assign.

The Kubernetes Cluster Details Screen

Clicking **Cluster** in the main menu opens the **Cluster Details** screen for the current Kubernetes cluster.

The top of this screen contains the **Cluster Operations** pull-down menu, which allows you to:

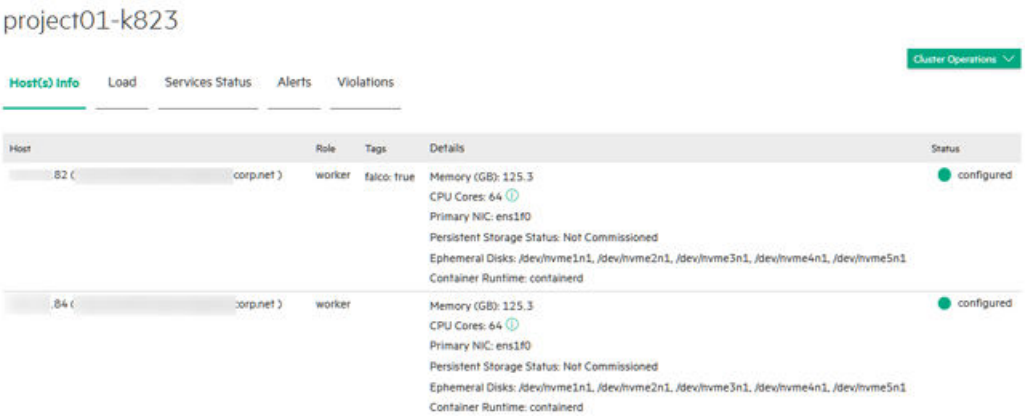
- **Access the Kubernetes dashboard:** See Accessing the Kubernetes Dashboard.
- **Download the Administrator Kubeconfig file:** See Downloading the Admin Kubeconfig.

The following tabs are available:

- **Host(s) Info:** This tab displays information about the hosts in the current cluster. See Host(s) Info Tab.
- **Load:** This tab displays load statistics for on-premises CPU, memory, and network resources within the deployment. See Load Tab.
- **Services Status:** This tab displays the health status for each component service within the deployment for each host. See Services Status Tab.
- **Alerts:** This tab displays any alert messages generated by the system. See Alerts Tab.

Host(s) Info Tab

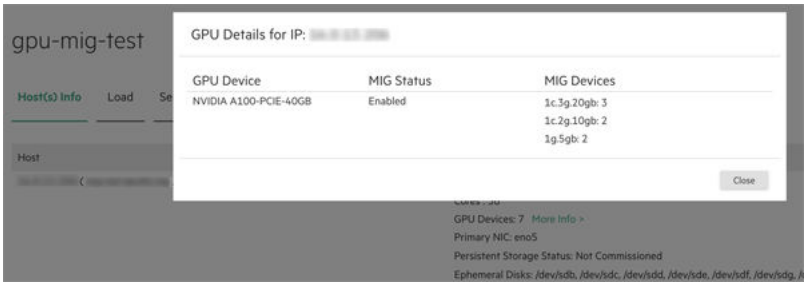
The **Host(s) Info** tab displays information about the hosts in the current Kubernetes cluster.



The table on this page displays the following information for each host in the cluster:

- **Host:** IP address and hostname of the host.
- **Role:** Role of the host (**Master** or **Worker**).
- **Tags:** The tags that have been assigned to the host. For example, HPE Ezmeral Data Fabric hosts have the tag: `Datafabric: Yes`
- **Details:** This column presents the following information:
  - **Memory:** Amount of RAM, in GB.
  - **Cores:** Number of CPU cores.
  - **GPU Devices:** The number of GPU devices.

    If the GPU supports MIG, when you click the **More Info** link, **GPU Details** dialog shows information about the MIG configuration. For example:



If the GPU device does not support MIG, the **GPU Details** dialog lists the GPU devices, but shows `N/A` in **MIG Status** and in **MIG Devices**.

- **Primary NIC:** Name of the primary Network Interface Card.
- **Persistent Storage Status:** Status of the persistent storage services (i.e. HPE Ezmeral Data Fabric FS).
- **Ephemeral Disks:** Path to the ephemeral storage resource on the host.
- **Persistent Disks:** Path to the persistent storage resource on the host.
- **Container Runtime:** If the host is running the Hewlett Packard Enterprise distribution of Kubernetes, the container runtime is `containerd`. If the host is part of a Kubernetes cluster that was created on a previous version of HPE Ezmeral Runtime Enterprise and has not been migrated to use the Hewlett Packard Enterprise distribution of Kubernetes, the container runtime is `Docker`.

- **Status:** This column will say **configured** for all fully-installed Kubernetes hosts. See Step 3: Add the Host(s) and Step 6: Add the Kubernetes Host(s) as Worker(s) for the statuses that may appear during host installation.

Load Tab

The **Load** tab displays a series of dials and charts. Hovering the mouse over a bar opens a popup with more detailed information for the selected time.

This tab shows the following information for the selected time period:

- **Host CPU Utilization Percent:** Current percentage of host CPU utilization across all cluster processes that are currently running for the selected hosts over the selected time period.
- **Host Memory Usage:** Current use of host memory across all cluster processes for the selected hosts over the selected time period.
- **Host Swap Memory Usage:** Amount of swap-file usage over the selected time period, in GB, for the selected hosts over the selected time period.
- **Host System Load:** Overall load placed on each host.

- **Host Network Traffic (Bytes In):** Amount of incoming host network bandwidth being used by the selected hosts over the selected time period.
- **Host Network Traffic (Bytes Out):** Amount of outgoing host network bandwidth being used by the selected hosts over the selected time period.

The following additional information applies to tenants with GPUs enabled:

- **GPU Utilization (percent):** Selecting **All hosts** in the left pull-down menu displays aggregate GPU utilization in percent per host. Selecting an individual host displays per-GPU utilization for that host.
- **GPU Memory Usage:** Selecting **All hosts** in the left pull-down menu displays aggregate GPU memory usage in percent per host. Selecting an individual host displays per-GPU memory usage for that host.

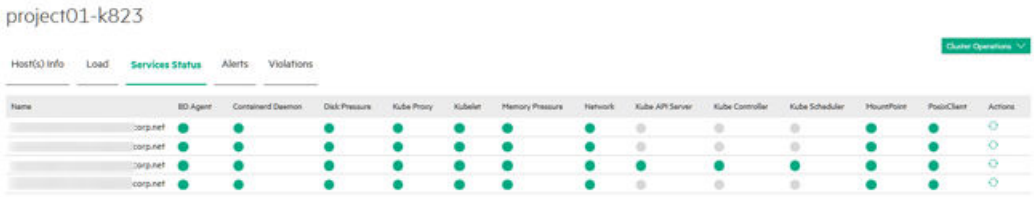
You may select the hosts you want to view and also adjust the time period for which results appear using the pull-down menus at the right side of the **Load** tab. The available options are:

- Last Hour (default)
- 6 Hours
- Day
- Week

Services Status Tab

NOTE: This tab is not available for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

The **Services Status** tab displays the status of services for each host being used for this Kubernetes cluster.



This tab displays information such as (but not necessarily limited to) the following for each host in the deployment:

- **Name:** Name of the host.
- **BD Agent:** Status of the management service, which handles back-end administration tasks.
- **Monitoring Collector:** Status of the monitoring engine that collects performance, usage, and other metrics.
- **Disk Pressure:** Whether the available disk space and inodes on either the node's root filesystem or image filesystem has satisfied an eviction threshold.
- **Containerd Daemon:** Status of the containerd daemon, which creates and manages Kubernetes containers.
- **Kube API Server:** Status of the Kubernetes API server.
- **Kube Controller:** Status of the Kubernetes controller host.

- **Kube Proxy:** Status of the Kubernetes proxy.
- **Kube Scheduler:** Status of the control plane Kubernetes scheduler.
- **Kubelet:** Maintains the pods that are running inside each host.
- **Memory Pressure:** Whether the available host memory has satisfied an eviction threshold.
- **Network:** Kubernetes network status.
- **FileServer:** File server status of the integrated persistent storage.
- **MountPoint:** Mount point status of the integrated persistent storage.
- **PosixClient:** Status of the POSIX Client of the integrated persistent storage.
- **Warden:** Warden status.

The status of a service can be either **OK** (green dot), **CRITICAL** (red dot), or **DISABLED** (intentionally not running; gray dot). Hovering the mouse over the status button opens a popup with additional information. In general:

- The Master host must not display any red dots. If the Master host has one or more errors, then the Kubernetes cluster may not function properly.
- If all of the dots for a Worker host are red, then that host will not be able to provide resources to the cluster. This situation typically occurs because the host has been powered off, has lost network connectivity, or because HPE Ezmeral Runtime Enterprise is not properly installed.
- A Worker host with some red and some green dots may cause some Kubernetes cluster operations to fail, unless the errors are transient conditions caused by the host powering on or regaining network connectivity.

Please generate a support bundle and then contact Hewlett Packard Enterprise Technical Support if a host that is reporting service errors meets all of the following criteria:

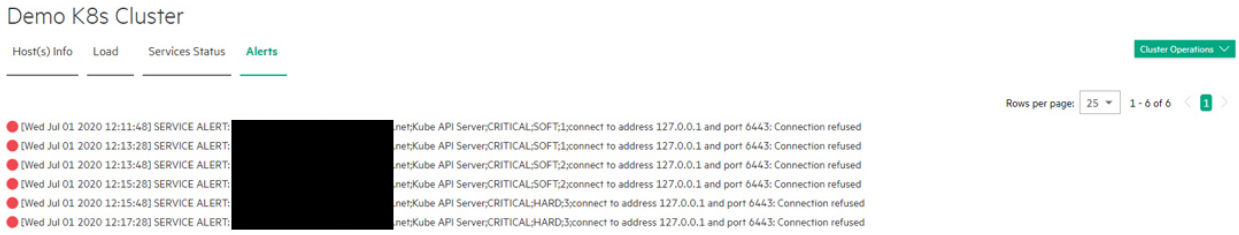
- HPE Ezmeral Runtime Enterprise is completely installed.
- The host is powered on.
- The host has network connectivity.

See The Support/Troubleshooting Screen and Generating a Support Bundle.

Alerts Tab

NOTE: This tab is not available for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

The **Alerts** tab displays any alert messages from the Caching Node, Data Server, and Management services.



The following alerts appear in this tab:

- **Notifications:** Routine messages. A green dot appears next to each routine notification.
- **Error:** A minor error has occurred. A gray dot appears next to each error notification.
- **Warning:** A serious error has occurred. An orange dot appears next to each warning notification.
- **Critical:** A critical error has occurred. A red dot appears next to each critical notification.

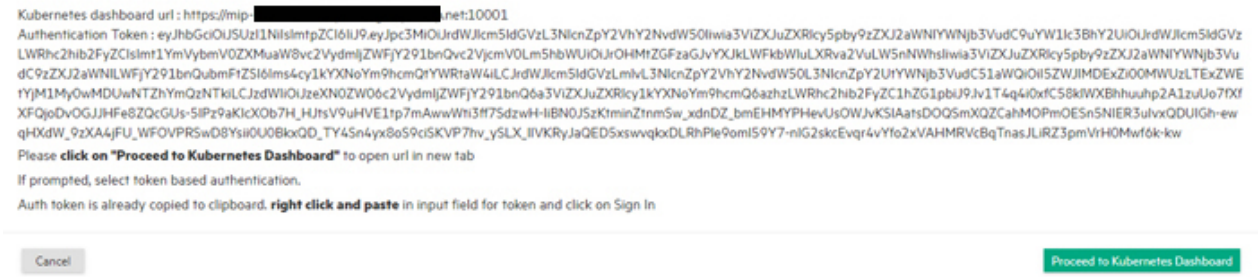
NOTE: The presence of non-routine alerts does not mean that HPE Ezmeral Runtime Enterprise will not function normally.

Accessing the Kubernetes Dashboard

To access the Kubernetes dashboard:

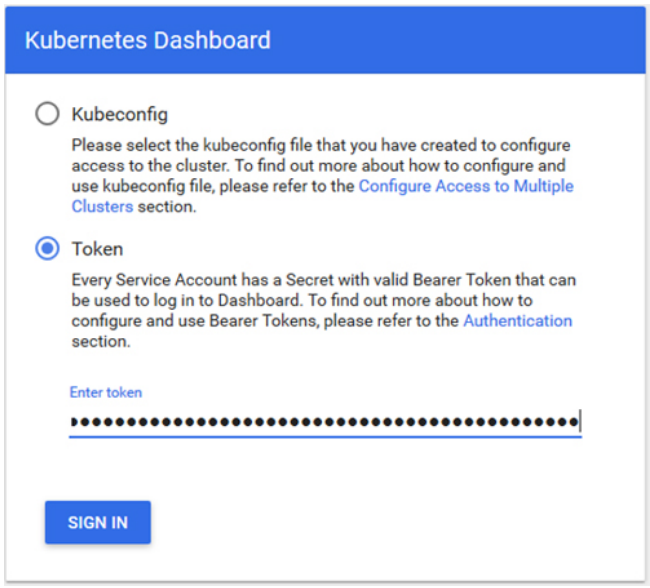
1. Accessing this function varies by your assigned role:
   - If you are a Platform Administrator user, you may click the **Access Kubernetes Dashboard** icon (screen) for the desired cluster in the **Clusters** screen. See The Kubernetes Clusters Screen.
   - Platform and Kubernetes Cluster Administrator users can select **Access Kubernetes Dashboard** from the **Cluster Operations** menu in the **Cluster Details** screen. See Viewing Kubernetes Cluster Details and The Kubernetes Cluster Details Screen.

     A popup appears with the authentication token.



2. Click **Proceed to Kubernetes Dashboard**. This copies the token to your clipboard.

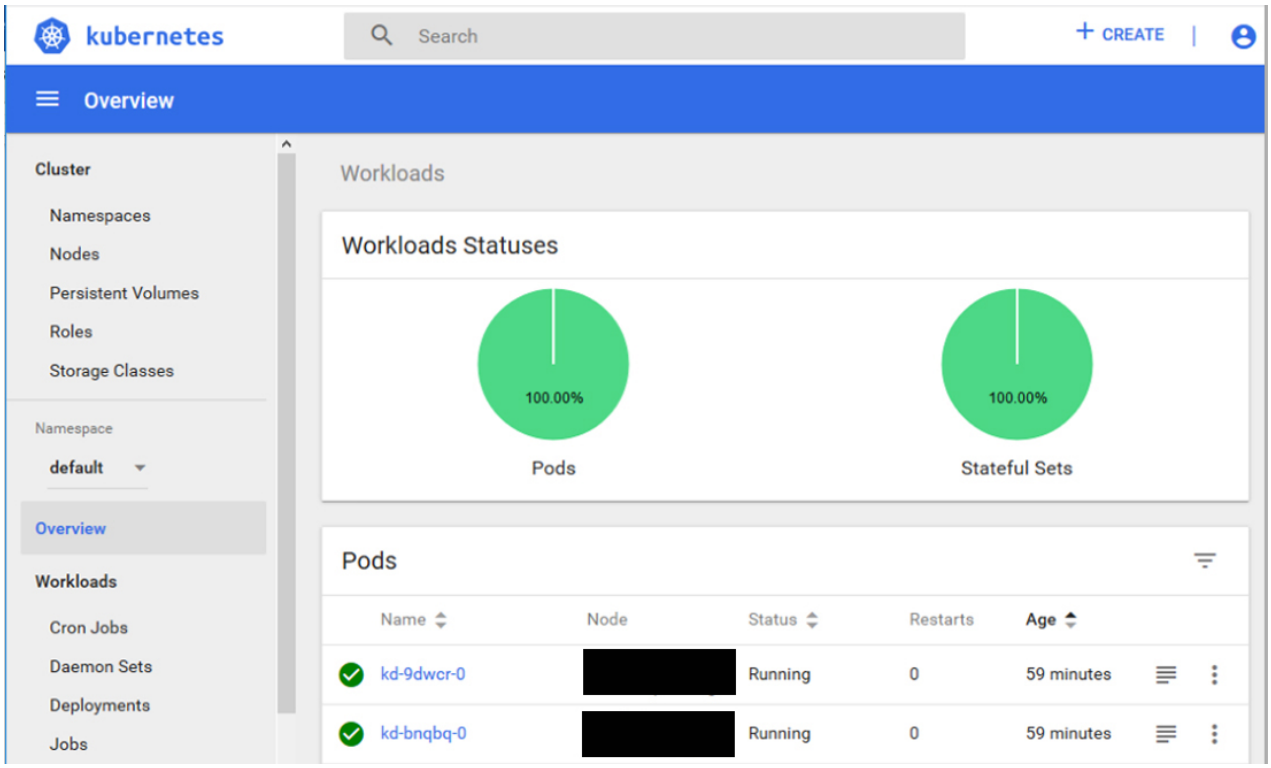
   The **Kubernetes Dashboard** sign-on appears.



3. Check the **Token** radio button, and then paste the token into the **Enter Token** field.

4. Click **Sign In**.

The Kubernetes dashboard appears.



If you are having issue accessing the Kubernetes Dashboard on a subsequent attempt, then:

1. Delete your browser cache and cookies.

2. Restart the browser.

3. Restart the Kubernetes dashboard.

Downloading Admin Kubeconfig

Kubernetes Cluster Adminstrator and Platform Adminstrator users can download the Admin Kubeconfig file for a cluster, as follows:

- **Cluster Administrator:** Select **Download Admin Kubeconfig** from the **Cluster Operations** pull-down menu in the **Kubernetes Cluster Details** screen. See Viewing Kubernetes Cluster Details.

- **Platform Administrator:** Click the **Download Admin Kubeconfig** icon (down arrow) for the desired cluster in the **Kubernetes Clusters** screen. See The Kubernetes Clusters Screen.

The downloaded file will look something like this:

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: <certificate goes here>
    server: https://mip.storage.enterprise.net:10000
  name: k8s-1
contexts:
- context:
    cluster: k8s-1
```

```
    user: kubernetes-admin
  name: kubernetes-admin@k8s-1
current-context: kubernetes-admin@k8s-1
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: <certificate goes here>
    client-key-data: <key goes here>
```

Cluster Kubeconfig

Kubernetes users can download the non-administrative (Member) Kubeconfig file for a cluster by clicking the **Kubeconfig** button in the Kubernetes **Dashboard** screen.

The downloaded file will look something like this:

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: <certificate string goes here>
    server: https://test.mynewdeployment.com:9500
  name: Kubernetes Cluster One
contexts:
- context:
    cluster: Kubernetes Cluster One
    user: HPECP-k8s_member
    namespace: k8s-tenant1
  name: Kubernetes Cluster One-K8S Tenant1-k8s_member
current-context: Kubernetes Cluster One-K8S Tenant1-k8s_member
kind: Config
preferences: {}
users:
- name: HPECP-k8s_member
  user:
    exec:
      command: kubectl
      apiVersion: client.authentication.k8s.io/v1beta1
      args:
      - hpecp
      - authenticate
      - test.mynewdeployment.com:8080
      - --hpecp-user=k8s_member
      - --hpecp-token=/api/v2/session/<UUID goes here>
      - --hpecp-token-expiry=1581033286
      - --insecure=true
      - --insecure-skip-tls-verify=true
```

Kubernetes Certificate Management

By default, all Kubernetes clusters created by HPE Ezmeral Runtime Enterprise have:

- A certificate authority with a 10-year life span.
- Client certificates with a 1-year life span.

CAUTION:

Kubernetes cluster certificates are created with a one-year duration. If the certificates are allowed to expire, the cluster will become unuseable until the certificates are manually re-generated.

To prevent this situation from occurring, about a month prior to the expiration of the certificate, contact Hewlett Packard Enterprise support for assistance with generating new certificates.

Viewing the Expiration Dates of Certificates

To view the expiration dates of both your CA and the certificate license, execute the following command:

```
kubeadm alpha certs check-expiration
```

For example:

```
kubeadm alpha certs check-expiration
[check-expiration] Reading configuration from the
cluster...                        [check-expiration] FYI: You can
look at this config file with 'kubectl -n kube-system getr
cm kubeadmin -oyaml'                        CERTIFICATE
ESPIRES                       RESIDUAL TIME  CERT AUTHORITY
EXT. MANAGED                        admin.conf                Aug
29, 2021 00:32  UTC 345d  345d
no                        apiserver                 Aug 29, 2021 00:32
UTC 345d  345d           ca              no
apiserver-etcd-client     Aug 29, 2021 00:32  UTC 345d  345d
etcd-ca         no                        apiserver-khbelet-client  Aug
29, 2021 00:32  UTC 345d  345d           ca
no                        controller-manager.conf   Aug 29, 2021
00:32  UTC 345d  345d                           no
etcd-healthcheck-client   Aug 29, 2021 00:32  UTC 345d  345d
etcd-ca         no                        etcd-peer                Aug
29, 2021 00:32  UTC 345d  345d           etcd-ca
no                        front-proxy-client        Aug 29, 2021 00:32
UTC 345d  345d           etcd-ca         no
scheduler.conf            Aug 29, 2021 00:32  UTC 345d
345d           front-proxy-ca  no                        CERTIFICATE
AUTHORITY  EXPIRES                RESIDUAL TIME  EXTERNALLY
MANAGED                        ca                     AUG 27,2030 00:22 UTC
9y             no                        etcd-ca                AUG 27,2030
00:22 UTC  9y             no                        front-proxy-ca
AUG 27,2030 00:22 UTC  9y             no# kubeadmin alpha certs
check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n
kube-system getr cm kubeadmin -oyaml'
CERTIFICATE               EXPIRES                       RESIDUAL TIME  CERT
AUTHORITY  EXT. MANAGED
admin.conf                Aug 29, 2021 00:32  UTC 345d
345d                           no
apiserver                 Aug 29, 2021 00:32  UTC 345d  345d
ca              no
apiserver-etcd-client     Aug 29, 2021 00:32  UTC 345d  345d
etcd-ca         no
apiserver-khbelet-client  Aug 29, 2021 00:32  UTC 345d  345d
ca              no
controller-manager.conf   Aug 29, 2021 00:32  UTC 345d
345d                           no
etcd-healthcheck-client   Aug 29, 2021 00:32  UTC 345d  345d
etcd-ca         no
etcd-peer                 Aug 29, 2021 00:32  UTC 345d  345d
```

```
etcd-ca         no
front-proxy-client        Aug 29, 2021 00:32  UTC 345d  345d
etcd-ca         no
scheduler.conf            Aug 29, 2021 00:32  UTC 345d  345d
front-proxy-ca  no
CERTIFICATE AUTHORITY  EXPIRES                RESIDUAL TIME  EXTERNALLY
MANAGED
ca                     AUG 27,2030 00:22 UTC  9y             no
etcd-ca                AUG 27,2030 00:22 UTC  9y             no
front-proxy-ca         AUG 27,2030 00:22 UTC  9y             no
```

Renewing a Certificate

CAUTION:

If the certificates are allowed to expire, the cluster will become unuseable until the certificates are manually re-generated.

About one month prior to the expiration of the certificate, contact Hewlett Packard Enterprise support for assistance with generating new certificates.

Certificate Authority (CA) Rotation

HPE Ezmeral Runtime Enterprise does not provide an automated method of rotating or replacing CA certificates. To manually rotate or replace CA certificates, see Manual Rotation of CA Certificates in the Kubernetes documentation (link opens an external website in a new browser tab or window).

When creating Kubernetes clusters, you can provide custom or external CA certificates and keys. HPE Ezmeral Runtime Enterprise uses kubeadm for initialization. The CA certificate and key that you provide during cluster initialization are written to the locations specified in Certificate Management with kubeadm in the Kubernetes documentation (link opens an external website in a new browser tab or window).

HPE Ezmeral Runtime Enterprise does not support the use of external CA certificates without keys.

Kubernetes Administrator Tasks

The topics in this section describe information and tasks that Kubernetes Administrators can perform in HPE Ezmeral Runtime Enterprise.

Dashboard - Kubernetes Administrator

Platform Administrator users who have access to the **Site Admin** tenant can access the Kubernetes Administrator **Dashboard** screen by selecting **Dashboard** in the main menu. The Kubernetes Administrator **Dashboard** screen presents a high-level overview of current Kubernetes activity. (See Dashboard - Platform Administrator on page 570 for information about the dashboard for EPIC Big Data tenants and AI/ML projects.)

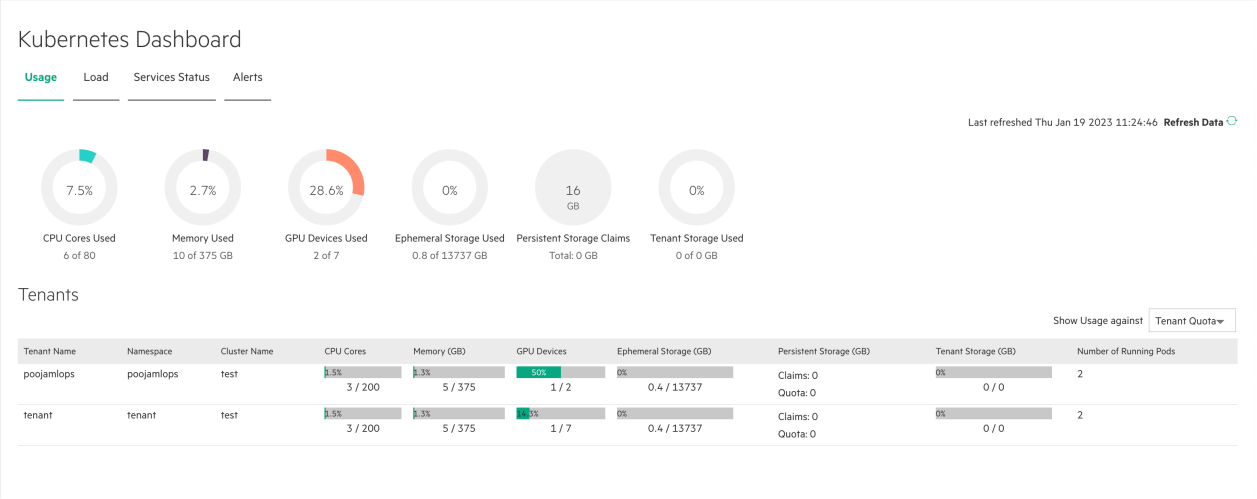
The top of this screen contains the **Refresh Data** function, which displays the date and time of the most recent **Dashboard** refresh. Clicking the **Refresh Data** button refreshes the data on this screen.

The following tabs are available:

- **Usage:** This tab displays usage information on a per-tenant basis. See Usage Tab.
- **Load:** This tab displays load statistics for on-premises CPU, memory, and network resources within the deployment. See Load Tab.
- **Services:** This section displays the health status for each component service within the deployment for each host. See Services Tab.
- **Alerts:** This tab displays any alert messages generated by the system. See Alerts Tab.

Usage Tab

The **Usage** tab displays usage statistics for the Kubernetes clusters and tenants.



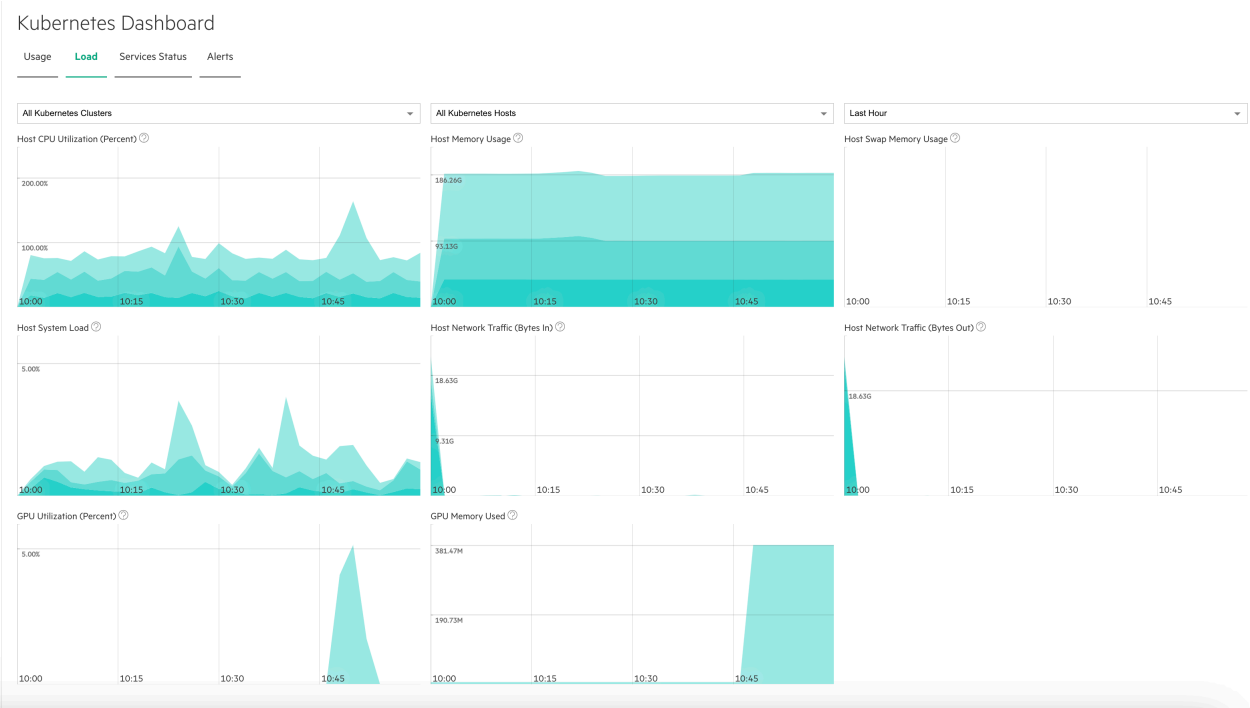
The top of the **Usage** tab displays dials showing the following aggregate information for all of the tenants in the deployment:

- **Cores Used:** Percentage of available virtual CPU cores being used by all of the tenants in the deployment.
- **Memory Used (GB):** Percentage of available RAM being used by all of the tenants in the deployment.
- **Ephemeral Storage Used (GB):** Percentage of available ephemeral storage used and the total available persistent storage, in GB.
- **Persistent Storage Used (GB):** Percentage of available persistent storage used and the total available persistent storage, in GB.
- **Tenant Storage Used (GB):** Percentage of available tenant storage used and the total available persistent storage, in GB.
- **GPU Devices Used:** Percentage of available GPU devices being used by all of the tenants in the deployment.
- The bottom of this tab contains a table that lists all of the Kubernetes tenants in the deployment. This table displays the **Tenant Name**, **Namespace**, **Cluster Name**, **Cores**, **Memory (GB)**, **Ephemeral Storage (GB)**, **Persistent Storage (GB)**, **Tenant Storage (GB)**, and the number of **Running Pods** being used by that tenant. This number is expressed as **x** of **y**, where **x** is the allotted number and **y** is either the Tenant Quota or total System Resources, depending on your **Show Usage against** menu selection.

NOTE: For information about how to download detailed usage and uptime information in comma-delimited (.csv) format, see Downloading Kubernetes Usage Details.

Load Tab

The **Load** tab displays a series of dials and charts. Hovering the mouse over a bar opens a popup with more detailed information for the selected time.



This tab shows the following information for the selected time period:

- **Host CPU Utilization Percent:** Percentage of host CPU utilization across all user space processes that are currently running for the selected host(s) over the selected time period. On multi-core systems, the percentages can be greater than 100%.
- **Host Memory Usage:** Current use of host memory across all cluster processes for the selected host(s) over the selected time period.
- **Host Swap Memory Usage:** Amount of swap file usage over the selected time period for the selected host(s) over the selected time period, in GB.
- **Host System Load:** One-minute average system load percentage for the selected host(s) over the selected time period.

  Host Network Traffic (Bytes In): Amount of incoming host network bandwidth being used by the selected host(s) over the selected time period.
- **Host Network Traffic (Bytes Out):** Amount of outgoing host network bandwidth being used by the selected host(s) over the selected time period.

The following additional information applies to tenants with GPUs enabled:

- **GPU Utilization (percent):** Selecting **All hosts** in the left pull-down menu displays aggregate GPU utilization in percent per host. Selecting an individual host displays per-GPU utilization for that host.
- **GPU Memory Usage:** Selecting **All hosts** in the left pull-down menu displays aggregate GPU memory usage in percent per host. Selecting an individual host displays per-GPU memory usage for that host.

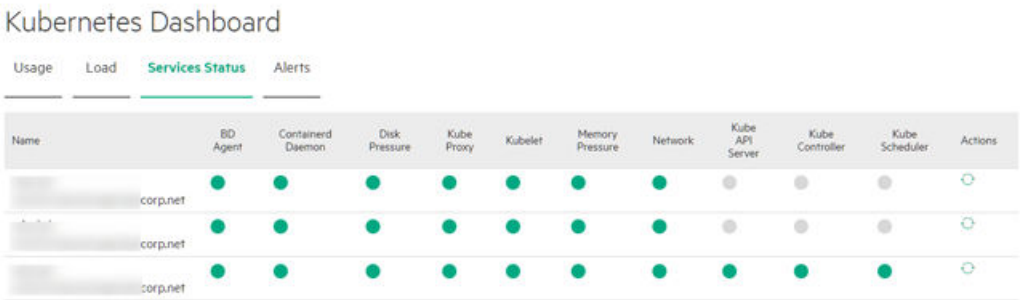
You may select the host(s) you want to view and also adjust the time period for which results appear using the pull-down menus at the right side of the **Load** tab. The available options are:

- Last Hour (default)
- 6 Hours
- Day

- Week

Services Tab

The **Services Status** tab displays the status of services for each host being used for Kubernetes tenants.



This tab displays information such as (but not necessarily limited to) the following for each host in the deployment:

- **Host Name:** Name of the host.
- **BD Agent:** Status of the management service, which handles back-end administration tasks.
- **Monitoring Collector:** Status of the monitoring engine that collects performance, usage, and other metrics.
- **Disk Pressure:** Whether the available disk space and inodes on either the node's root filesystem or image filesystem has satisfied an eviction threshold.
- **Containerd Daemon:** Status of the containerd daemon, which creates and manages containers.
- **Kube API Server:** Status of the Kubernetes API server.
- **Kube Controller:** Status of the Kubernetes controller host.
- **Kube Proxy:** Status of the Kubernetes proxy.
- **Kube Scheduler:** Status of the control plane Kubernetes scheduler.
- **Kubelet:** Maintains the pods that are running inside each host.
- **Memory Pressure:** Whether the available host memory has satisfied an eviction threshold.
- **Network:** Kubernetes network status.
- **FileServer:** File server status of the integrated persistent storage.
- **MountPoint:** Mount point status of the integrated persistent storage.
- **PosixClient:** Status of the POSIX Client of the integrated persistent storage.
- **Warden:** Warden status.

The status of a service can be either **OK** (green dot), **CRITICAL** (red dot), or **DISABLED** (intentionally not running; gray dot). Hovering the mouse over the status button opens a popup with additional information. In general:

- The Master host must not display any red dots. If the Master host has one or more error(s), then the Kubernetes cluster may not function properly.

- If all of the dots for a Worker host are red, then that host will not be able to provide resources to the cluster. This situation typically occurs because the host has been powered off, has lost network connectivity, or because HPE Ezmeral Runtime Enterprise is not properly installed.
- A Worker host with some red and some green dots may cause some Kubernetes cluster operations to fail, unless the errors are transient conditions caused by the host powering on or regaining network connectivity.

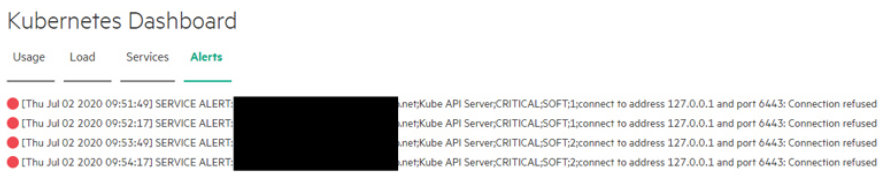
Please generate a support bundle and then contact HPE Technical Support if a host that is reporting service errors meets all of the following criteria:

- HPE Ezmeral Runtime Enterprise is completely installed.
- The host is powered on.
- The host has network connectivity.

See The Support/Troubleshooting Screen and Generating a Support Bundle.

Alerts Tab

The **Alerts** tab displays any alert messages from the Caching Node, Data Server, and Management services.



The following alerts appear in this tab:

- **Notifications:** Routine messages. A green dot appears next to each routine notification.
- **Error:** A minor error has occurred. A gray dot appears next to each error notification.
- **Warning:** A serious error has occurred. An orange dot appears next to each warning notification.
- **Critical:** A critical error has occurred. A red dot appears next to each critical notification.

NOTE: The presence of non-routine alerts does not mean that HPE Ezmeral Runtime Enterprise will not function normally.

Toolbar and Main Menu - Kubernetes Administrator

A **Kubernetes Administrator** is a **Platform Administrator** in the context of managing Kubernetes hosts, clusters, tenants, and users.

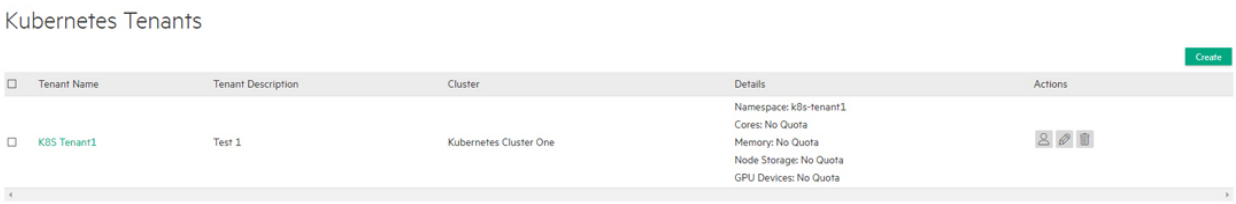
See the KUBERNETES section of Toolbar & Main Menu - Platform Administrator on page 575.

Kubernetes Tenant Administration

The topics in this section describe information and tasks related to Kubernetes tenant administration on HPE Ezmeral Runtime Enterprise.

The Kubernetes Tenants Screen

Selecting **Tenants** in the main menu opens the **Kubernetes Tenants** screen, which allows you to manage Kubernetes tenants.

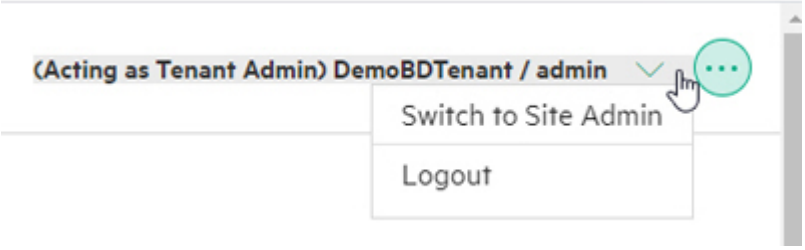


This screen contains the following buttons:

- **Create:** Clicking this button opens the **Create New Tenant** screen. See Creating a New Kubernetes Tenant.
- **Delete:** Selecting one or more tenant(s) and then clicking this button deletes the selected tenant(s), if they are eligible for deletion. See Deleting a Kubernetes Tenant. This button only appears if you select one or more Kubernetes tenant(s).

The table on this screen contains the following information and functions:

- **Tenant Name:** Name of the tenant. Clicking a tenant name opens the Kubernetes Tenant Administrator **Dashboard** screen for the selected tenant and displays the Kubernetes Tenant Administrator menu for the selected tenant. This allows you to work directly within the selected tenant as a Kubernetes Tenant Administrator. The message **Acting as Tenant Admin** appears in the **Toolbar** at the top of web interface screens while you are in this mode. Hovering the mouse over this message opens a menu that allows you to return to the **Site Admin** tenant. See Toolbar & Main Menu - Kubernetes Tenant Administrator for articles on using the Kubernetes Tenant Administrator interface.



- **Tenant Description:** Brief description of the tenant.
- Cluster: Name of the Kubernetes cluster where this tenant is located.
- **Details:** Detailed information about the tenant, including:
  - **Resources:** Resource quotas assigned for the tenant, compared to the total available resources in the system, such as **`Cores=8/16`**. Resource quotas can be assigned for virtual CPU cores, RAM, node storage, GPU devices (if the deployment is running RHEL/CentOS 7.x and has one or more GPU device(s) installed) and persistent tenant storage. If the tenant has no quota for a resource, then the display will show the resources being used (such as **`Cores=8`**) and the message **`No Quota`**.
- **Actions:** The following actions are available for each tenant:
  - **Users:** Clicking the **Users** icon (person) in the **Actions** opens the **Tenant Details** screen for that tenant, which allows you to either assign and revoke user roles or delete a user. See Viewing User Assignments.
  - **Edit:** Clicking the **Edit** icon (pencil) in the **Actions** column opens the **Edit Tenant** screen for the tenant. See Editing an Existing Kubernetes Tenant. You cannot edit the **Site Admin** tenant.

- **Delete:** This icon (trash can) appears if the tenant is eligible for deletion. See Deleting a Kubernetes Tenant.

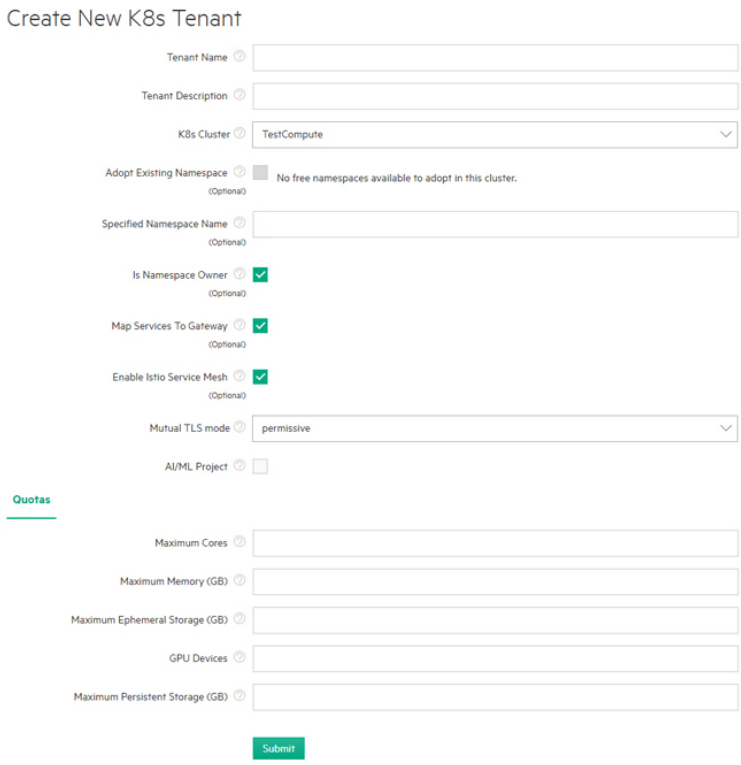
Creating a New Kubernetes Tenant or Project

You can create Kubernetes tenants that are associated with a Kubernetes cluster. Each Kubernetes tenant corresponds to a namespace on the cluster, an optional resource quota, and a set of privileges for various user roles within that namespace. Specific users can be assigned to have roles in Kubernetes tenants, or entire AD/LDAP groups can be mapped to Kubernetes tenant roles.

After creating a tenant, you can do the following:

- Use the web interface to send Kubernetes API requests for resource creation, modification, or deletion using using the privileges of the logged-in user's role.
- Use the web interface to access a Kubernetes a web terminal (see Kubernetes Web Terminal) to access a Linux environment set up with the kubectl CLI and a configuration appropriate for your user role.
- Download materials to configure kubectl on your own local workstation so that it can access the Kubernetes cluster using the privileges assigned to your role.

To create a new Kubernetes tenant, click the **Create** button in the **Kubernetes Tenants** screen to open the **Create New K8s Tenant** screen.



Create the Kubernetes tenant as follows:

1. Create at least one Kubernetes cluster, as described in Creating a New Kubernetes Cluster.
2. Enter a name for the new tenant in the **Tenant Name** field.
3. Enter a brief description for the new tenant in the **Tenant Description** field.
4. Use the **K8s Cluster** pull-down menu to select the Kubernetes cluster to associate with this tenant.

5. If you want to associate the tenant with an existing namespace, then check the **Adopt Existing Namespace** check box and use the **Existing Namespaces** pull-down menu to select the desired namespace.

   If not, then leave this check box blank and either enter a unique namespace name in the **Specified Namespace Name** field or leave this field blank to auto-generate a namespace name.

6. If you want the namespace and all of its contents to be deleted when the tenant is deleted, then check the **Is Namespace Owner** check box. If not, then leave this check box blank.

7. If you want to map the service endpoints that will exist in this tenant to Gateway host ports, then check the **Map Services to Gateway** check box. Leaving this check box blank will not map services to a Gateway host, and you will need to access service endpoints by SSHing directly into containers. See Gateway Hosts.

8. If the cluster supports Istio (see Creating a New Kubernetes Cluster and Istio), then you may check the **Enable Istio Service Mesh check box** and then use the **Mutual TLS Mode** pull-down menu to select one of the following:

   - **disable:** TLS encryption will not be used in the Istio service mesh.
   - **permissive (default):** The Istio service mesh will support both encrypted and unencrypted traffic.
   - **strict:** Only TLS-encrypted traffic will be accepted in the Istio Service mesh.

9. Either:

   - **Tenant:** If you are not creating an AI/ML project, then leave the **AI/ML Project** check box cleared.

     See Getting Started with General Kubernetes Functionality.

   - **Project:** If you are creating an AI/ML project, then check the **AI/ML Project** check box. (Not available in HPE Ezmeral Runtime Enterprise Essentials.)

     See HPE Ezmeral ML Ops on page 148.

10. Specify vCPU (cores), RAM, GPU, and/or storage quotas using the **Quotas** tab. When **AI/ML Project** is selected, an entry in **Maximum Cores** is required. Other **Quotas** fields are optional.

    See Kubernetes Tenant Quotas.

11. If applicable, specify the tenant-independent settings or LDAP/AD groups that will be able to access this tenant using the **External Authentication** tab.

    See Kubernetes Tenant External Authentication. This tab does not appear when the deployment is configured for platform-wide local authentication.

12. When you have finished creating the tenant, click **Submit** to save your changes.

A default DataTap is automatically created for the new tenant. See About DataTaps.

There are several things you should do to prepare the new tenant for use. See After Creating the Kubernetes Tenant, below.

After Creating the Kubernetes Tenant

Once the new Kubernetes tenant has been created, you should do all of the following:

- Assign at least one user to the tenant with either the Tenant Administrator or Platform Administrator role, as appropriate.

  See Assigning/Revoking User Roles (Local) or Assigning/Revoking User Roles (LDAP/AD), as appropriate.

- Add DataTaps or FS Mounts to the tenant. FS mounts are not available for imported Kubernetes Clusters.

  See Creating a New DataTap or Creating a New FS Mount. For information about imported clusters, see Importing an External Kubernetes Cluster.

- Download Kubectl to your system via the **Dashboard** screen for your assigned role.

- Access the Kubernetes Web Terminal.

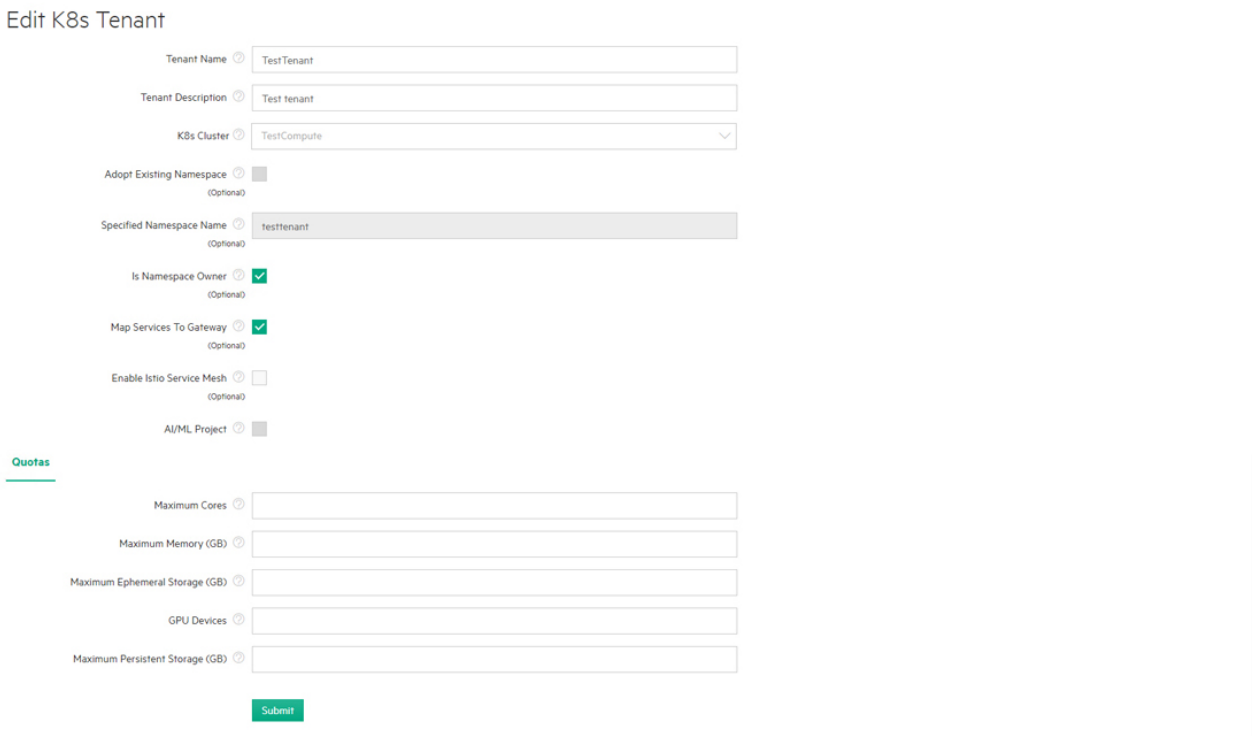
  See Kubernetes Web Terminal.

- Deploy or onboard one or more applications.

  See Applications Overview.

Editing an Existing Kubernetes Tenant or Project

In the **Kubernetes Tenants** screen, clicking the **Edit** icon (pencil) for a tenant opens the **Edit K8s Tenant** screen for that tenant.



You may edit some or all of the following parameters:

- **Tenant Name:** Name of the tenant.

- **Tenant Description:** Brief description of the tenant.

- If you want to associate the tenant with an existing namespace, then check the **Adopt Existing Namespace** check box and use the **Existing Namespaces** pull-down menu to select the desired namespace. If not, then leave this check box blank and either enter a unique namespace name in the **Specified Namespace Name** field or leave this field blank to auto-generate a namespace name.

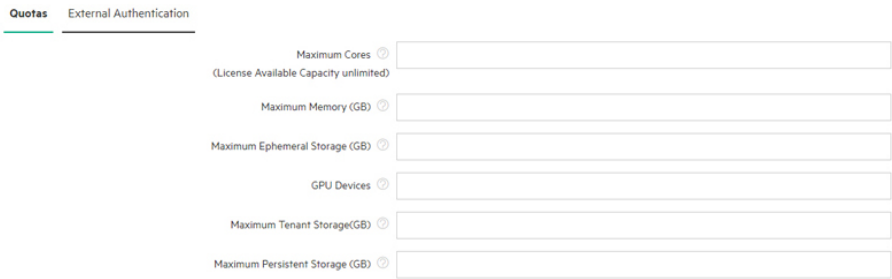
- If you want the namespace and all of its contents to be deleted when the tenant is deleted, then check the **Is Namespace Owner** check box. If not, then leave this check box blank.

- If you want to map the service endpoints that will exist in this tenant to Gateway host ports, then check the **Map Services to Gateway** check box. Leaving this check box blank will not map services to a Gateway host, and you will need to access service endpoints by SSHing directly into containers. See Gateway Hosts on page 106.
- If the cluster supports Istio (see Creating a New Kubernetes Cluster on page 463 and Istio Service Mesh on page 492), then you may check the **Enable Istio Service Mesh** check box and then use the **Mutual TLS Mode** pull-down menu to select one of the following:
  - **disable:** TLS encryption will not be used in the Istio service mesh.
  - **permissive (default):** The Istio service mesh will support both encrypted and unencrypted traffic.
  - **strict:** Only TLS-encrypted traffic will be accepted in the Istio Service mesh.
- Specify vCPU, RAM, GPU, and/or storage quotas using the **Quotas** tab. This step is optional for tenants. See Kubernetes Tenant/Project Quotas on page 455.
- If applicable, specify the tenant-independent settings or LDAP/AD groups that will be able to access this tenant using the **External Authentication** tab. See Kubernetes Tenant/Project External Authentication on page 456. This tab does not appear when the deployment is configured to use platform-wide local authentication.

When you have finished editing the tenant, click **Submit** to save your changes.

Kubernetes Tenant/Project Quotas

When you are creating or editing a Kubernetes tenant (see Creating a New Kubernetes Tenant or Editing an Existing Kubernetes Tenant), selecting the **Quotas** tab allows you to adjust CPU, storage, and QOS quotas for optimal Kubernetes tenant performance.



This tab allows you to specify the following settings:

- **Maximum Cores:** Enter the maximum number of virtual CPU cores that should be made available for use by this Kubernetes tenant.

  By default, this field will display a value equal to 25% of the available CPU cores with the appropriate license type, but you can specify any number you want. If you are creating or editing a Kubernetes tenant, then you may leave this field blank if you do not want to specify a quota.See Virtual Cores, RAM, Storage, and GPU Devices for important information about how virtual CPU cores are used. See also Licensing and License Tab for information on adding/updating a license.

- **Maximum Memory (GB):** Enter the maximum amount of RAM in GB that HPE Ezmeral Runtime Enterprise should make available for use by this Kubernetes tenant.

  By default, this field will display a value equal to 25% of the available memory that is not reserved for HPE Ezmeral Runtime Enterprise services, but you can specify any number you want, or leave this field blank if you do not want to specify a quota. Please see Virtual Cores, RAM, Storage, and GPU Devices for important information about how memory is used. See also Licensing and License Tab for information on adding/updating a license.

- **Maximum Ephemeral Storage (GB):** Enter the maximum space the tenant may use for ephemeral storage, in GB.

  By default, this field will display a value equal to 25% of the available ephemeral storage, but you can specify any number you want, or leave this field blank if you do not want to specify a quota. See Node Storage for more information about ephemeral storage.

- **GPU Devices:** Enter the maximum number of GPU devices the Kubernetes tenant may use.

  By default, this field will display a value of 0, but you can specify any number you want, or leave this field blank if you do not want to specify a quota. See Virtual Cores, RAM, Storage, and GPU Devices for more information about how GPU devices are used.

- **Maximum Tenant Storage (GB):** If the Kubernetes tenant uses local HDFS for tenant storage, then you can specify the maximum space to use for this storage, in GB.

  By default, this field will display a value equal to 25% of the available tenant storage, but you can specify any number you want, or leave this field blank if you do not want to specify a quota. See Tenant and Project Storage for more information about tenant storage.

NOTE: A Kubernetes tenant that performs lots of I/O to the base HDFS can exceed its assigned quota. This error appears as an INFO item in the NameNode log on the Controller host. Further, the CNODE log reports a socket disconnect (which can have multiple causes).

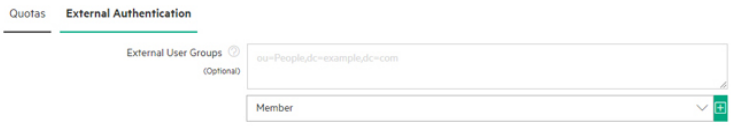
When diagnosing an I/O failure between a virtual node and the base HDFS, be sure to look for is the character string **`quota exceeded`** in the NameNode log.

When you have finished specifying quota settings, continue creating or editing the Kubernetes tenant, as described in Creating a New Kubernetes Tenant or Editing an Existing Kubernetes Tenant, as appropriate.

Kubernetes Tenant/Project External Authentication

The **External Authentication** tab of the **Create New K8s Tenant** or **Edit K8s Tenant** screen (see Creating a New Kubernetes Tenant or Editing an Existing Kubernetes Tenant) allows you to configure the user authentication options for the current tenant when platform-wide LDAP/AD user authentication is used.

- **Platform-wide LDAP/AD user authentication is used:** See Tenant Groups.



Please see User Authentication for information on user authentication, and Configuring User Authentication Settings for instructions on using the **External Authentication** tab.

When you have finished specifying authentication settings, continue creating or editing the Kubernetes tenant, as described in Creating a New Kubernetes Tenant or Editing an Existing Kubernetes Tenant, as appropriate.

Deleting a Kubernetes Tenant or Project
Describes the process and impact of deleting a Kubernetes tenant or project.

Deleting a Kubernetes Tenant or Project

Deleting a Kubernetes tenant does not affect any data placed on the storage services referenced by the tenant's DataTaps or FS Mounts. To delete a Kubernetes tenant:

1. Assign yourself as the Kubernetes Administrator using the **Assign Users** screen. See Assigning/Revoking User Roles (Local).

2. Log in to the web interface as a Tenant Administrator, and then select **Tenants** in the Kubernetes main menu to open the **Kubernetes Tenants** screen. See The Kubernetes Tenants Screen.

3. Click the **Delete** icon for each Kubernetes tenant you are deleting.

A confirmation dialog appears. Click **OK** to confirm the deletion.

CAUTION: You cannot undelete a Kubernetes tenant.

Impact of Deleting a Kubernetes Tenant

- Deleting a Kubernetes tenant unassigns all tenant -level user roles that were assigned to that tenant, but does not delete the user. If deleting this Kubernetes tenant causes one or more users to have zero roles left assigned to them, then the affected users will not be able to log in until they have been assigned at least one role, as described in Assigning/Revoking User Roles (Local).

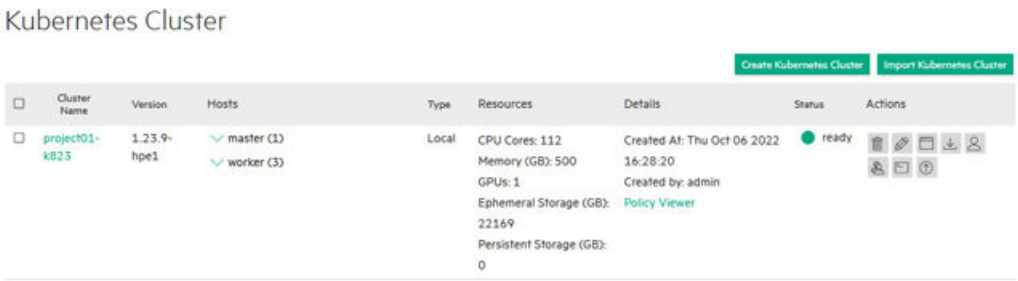
- Deleting a Kubernetes tenant does not delete the Tenant Storage. For information, see Tenant/Project Storage.

Clusters

The topics in this section describe information and tasks related cluster administration tasks performed by Kubernetes Administrators in HPE Ezmeral Runtime Enterprise.

The Kubernetes Clusters Screen

Selecting **Clusters** in the main menu opens the **Kubernetes Clusters** screen.



The top of this screen contains the following buttons:

- **Create Kubernetes Cluster**: Clicking this button opens the **Create Kubernetes Cluster** screen. See Creating a New Kubernetes Cluster.

- **Import Kubernetes Cluster**: Clicking this button opens the **Import Kubernetes Cluster** screen. See Importing an External Kubernetes Cluster.

The table on this screen contains the following information and functions:

- **Name**: Name of the cluster. Clicking a Kubernetes cluster name opens the **Kubernetes Cluster Details** screen for that Kubernetes cluster. See Viewing Kubernetes Cluster Details.

- **Version**: Version of Kubernetes running in the cluster.

  If the Kubernetes version number includes the phrase `-hpe<n>`, where <n> is a number, the host is running a Kubernetes version that is distributed by Hewlett Packard Enterprise, which uses the containerd runtime.

- **Resources**: This column presents the following information:

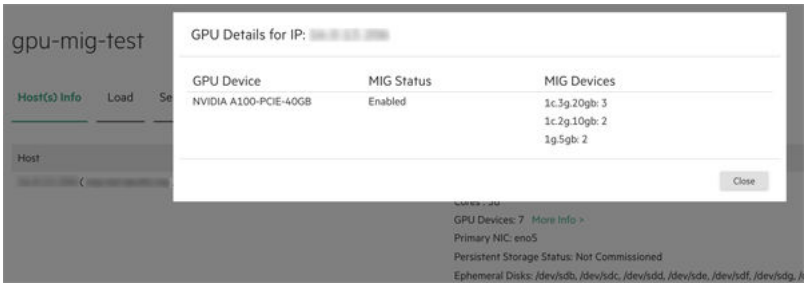
  - **Cores**: Number of CPU cores.

  - **Memory**: Amount of RAM, in GB.

- **GPUs**: Quantity of GPUs.

  If the GPU supports MIG, when you click the **More Info** link, **GPU Details** dialog shows information about the MIG configuration. For example:

  GPU Devices: The number of GPU devices.



  If the GPU device does not support MIG, the **GPU Details** dialog lists the GPU devices, but shows `N/A` in **MIG Status** and in **MIG Devices**.

- **Details**: Additional information about the cluster, such as the date and time it was created and the user who created it. For Data Fabric cluster, this column includes the following additional information:
  - `Datafabric`: Indicates `yes`, `YES`, `true`, or `TRUE` for Data Fabric Worker nodes.
  - Name of the Data Fabric associated with the cluster.
  - Links to the the Policy Viewer, the Grafana and Kibana services, and the Data Fabric Managed Control System. These links are not shown when the system is in Lockdown mode. See Lockdown Mode.

    NOTE: Grafana and Kibana Endpoints are not available for Footprint-Optimized configuration.

- **Actions**: The following actions are available for each Kubernetes cluster:
  - **Delete**: Clicking the **Delete** button (trash can) deletes the current Kubernetes cluster. See Deleting a Kubernetes Cluster.
  - **Edit**: Clicking the **Edit** icon (pencil) opens the **Edit Kubernetes Cluster** screen. See Editing an Existing Kubernetes Cluster.
  - **Access Kubernetes Dashboard**: Clicking the **Access Kubernetes Dashboard** button (screen) opens the Kubernetes dashboard for this Kubernetes cluster. See Accessing the Kubernetes Dashboard.
  - **Download Admin Kubeconfig**: Clicking the **Download Admin Kubeconfig** button (down arrow) downloads the Administrator Kubeconfig file for the Kubernetes cluster. See Downloading Admin Kubeconfig.
  - **Setup Log info**: Clicking the **Setup Log info** icon (envelope) opens the setup log for the Kubernetes cluster. See Viewing the Kubernetes Cluster Setup Log.
  - **Upgrade Kubernetes**: Clicking the **Upgrade Kubernetes** button (up arrow) allows you to upgrade the Kubernetes version that is installed on some or all of the virtual nodes/containers in the Kubernetes cluster. See Upgrading Kubernetes.
  - **Cluster Admin Users**: Clicking the **Cluster Admin Users** button (person) opens the **Kubernetes Cluster Administrator Users** screen, which allows you to assign/revoke the Kubernetes Cluster Administrator role. See Managing Kubernetes Cluster Admin Users.

- **Update Cluster Admin External Groups**: If HPE Ezmeral Runtime Enterprise is configured to use LDAP/AD authentication (see Configuring User Authentication Settings), then clicking the **Update Cluster Admin External Groups** button (people) allows you to assign the Kubernetes Cluster Administrator role to LDAP/AD groups. See Updating External Kubernetes Cluster Admin Groups.
- **Download CR**: For Data Fabric clusters, clicking the **Download CR** icon (down arrow) allows you to download the cluster CR in JSON format, which allows you to edit and fine-tune the cluster. See Manually Creating/Editing a Data Fabric Cluster.

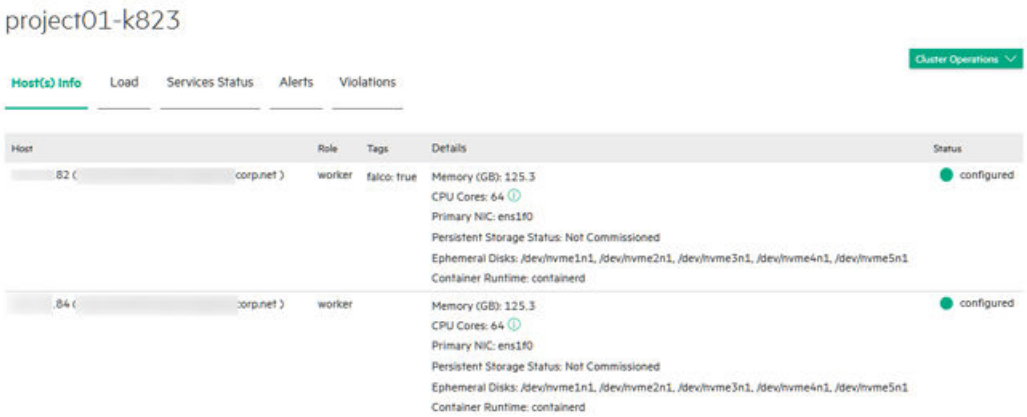
Viewing Kubernetes Cluster Details

Clicking a cluster name in the **Kubernetes Clusters** screen opens the **Cluster Details** screen for that cluster. This screen has the following tabs:

- **Host(s) Info:** Displays information about the hosts in the Kubernetes cluster. See Host(s) Info Tab.
- **Load Tab:** Displays resource usage information for the current Kubernetes cluster. See Load Tab.
- **Services Status:** This tab displays the status of various Kubernetes cluster services. See Services Status Tab.
- **Alerts:** Warnings or errors that affect the current Kubernetes cluster appear here. See Alerts Tab.

Host(s) Info Tab

The **Host(s) Info** tab of the **Cluster Details** screen appears as shown in the following image.



This tab contains the following button:

- **Cluster Operations:** Clicking this button opens a menu with the following options:
  - **Delete Cluster:** Deletes this Kubernetes cluster. See Deleting a Kubernetes Cluster.
  - **Edit Cluster:** Allows you to edit the current Kubernetes cluster. See Editing an Existing Kubernetes Cluster.
  - **Access Kubernetes Dashboard**: Launches the Kubernetes dashboard. See Accessing the Kubernetes Dashboard.
  - **Download Admin Kubeconfig:** Downloads the administrator Kubeconfig file for the cluster. See Downloading Admin Kubeconfig.
  - If this is a **HPE Ezmeral Data Fabric on Kubernetes** cluster, the following items are also displayed:
    - **Grafana Endpoint**
    - **Kibana Endpoint**

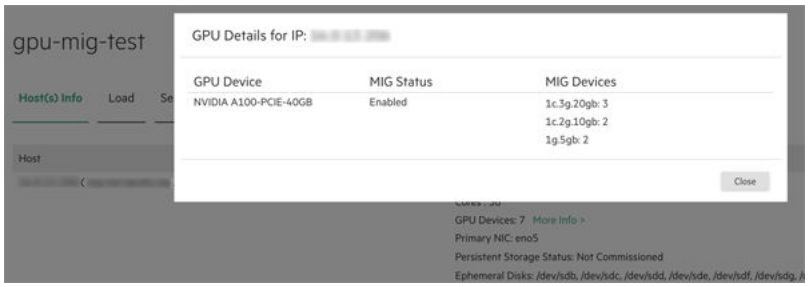
- **Data Fabric Managed Control System**

The **Host List** table on this tab displays the following information for each of the hosts in the selected cluster:

- **Name:** Name of the host.
- **Role:** Role of the host, such as Master (**master**) or Worker (**worker**). If custom roles are defined for this cluster, those will appear here as well.
- **Tags:** The tags that have been assigned to the host. For example, HPE Ezmeral Data Fabric hosts have the tag: `Datafabric: Yes`
- **Details:** Lists information about the host, such as the CPU cores, number of GPU devices, RAM, primary NIC, persistent storage status, the paths to the ephemeral and persistent storage, and the container runtime.

  If the host is running the Hewlett Packard Enterprise distribution of Kubernetes, the container runtime is `containerd`. If the host is part of a Kubernetes cluster that was created on a previous version of HPE Ezmeral Runtime Enterprise and has not been migrated to use the Hewlett Packard Enterprise distribution of Kubernetes, the container runtime is `Docker`.

  If the GPU supports MIG, when you click the **More Info** link, **GPU Details** dialog shows information about the MIG configuration. For example:



  If the GPU device does not support MIG, the **GPU Details** dialog lists the GPU devices, but shows `N/A` in **MIG Status** and in **MIG Devices**.

- **Status:** Status of the host.

Load Tab

The **Load** tab displays a series of dials and charts. Hovering the mouse over a bar opens a popup with more detailed information for the selected time.

This tab contains the following buttons:

- This tab shows the following information for the selected time period:
- **Host CPU Utilization Percent:** The chart indicates the current percentage of host CPU utilization across all cluster processes that are currently running for the selected host(s) over the selected time period.
- **Host Memory Usage:** The chart indicates the current use of host memory across all cluster processes for the selected host(s) over the selected time period.
- **Host Swap Memory Usage:** The chart indicates the amount of swap-file usage over the selected time period, in GB, for the selected host(s) over the selected time period.
- **Host System Load:** The graph shows the average percentage of host CPU cores used by the Kubernetes tenants (defined as the number of CPU cores in use vs. the total number of available CPU cores) for the selected host(s) over the selected time period.
- **Host Network Traffic (Bytes In):** The dial indicates the amount of incoming host network bandwidth being used by the selected host(s) over the selected time period.
- **Host Network Traffic (Bytes Out):** The dial indicates the amount of outgoing host network bandwidth being used by the selected host(s) over the selected time period.

The following additional information applies to Kubernetes clusters with GPUs enabled:

- **GPU Utilization (percent):** Selecting **All hosts** in the left pull-down menu displays aggregate GPU utilization in percent per host. Selecting an individual host displays per-GPU utilization for that host.
- **GPU Memory Usage:** Selecting **All hosts** in the left pull-down menu displays aggregate GPU memory usage in percent per host. Selecting an individual host displays per-GPU memory usage for that host.

You may select the host(s) you want to view and also adjust the time period for which results appear using the pull-down menus at the right side of the **Load** tab. The available options are:

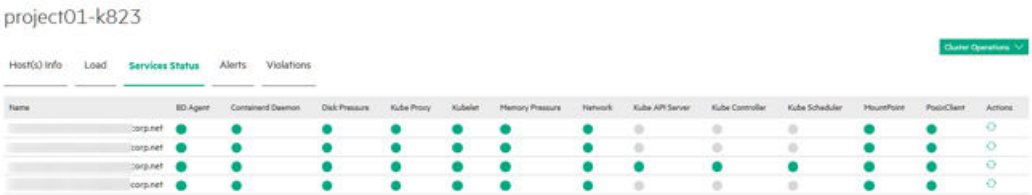
- Last Hour (default)
- 6 Hours
- Day

- Week

Services Status Tab

NOTE: This tab is not available for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

The **Services Status** tab of the Kubernetes **Cluster Details** screen appears as shown in the following image.



The information on this tab varies depending on the type of cluster. Hosts that are part of HPE Ezmeral Data Fabric on Kubernetes or Embedded Data Fabric deployment includes information about services related to the Data Fabric. This tab displays information such as (but not necessarily limited to) the following for each host in the deployment:

- **Host Name:**Name of the host.
- **BD Agent:** Status of the management service, which handles back-end administration tasks.
- **Monitoring Collector:** Status of the monitoring engine that collects performance, usage, and other metrics.
- **Disk Pressure:** Whether the available disk space and inodes on either the node's root filesystem or image filesystem has satisfied an eviction threshold.
- **Containerd Daemon:** Status of the containerd daemon, which creates and manages Kubernetes containers.
- **Kube API Server:** Status of the Kubernetes API server.
- **Kube Controller:** Status of the Kubernetes controller host.
- **Kube Proxy:** Status of the Kubernetes proxy.
- **Kube Scheduler:** Status of the control plane Kubernetes scheduler.
- **Kubelet:** Maintains the pods that are running inside each host.
- **Memory Pressure:** Whether the available host memory has satisfied an eviction threshold.
- **Network:** Kubernetes network status.
- **FileServer:** File server status of the integrated persistent storage.
- **MountPoint:** Mount point status of the integrated persistent storage.
- **PosixClient:** Status of the POSIX Client of the integrated persistent storage.
- **Warden:** Warden status.

The status of a service can be either **OK** (green dot), **CRITICAL** (red dot), or **DISABLED** (intentionally not running; gray dot). Hovering the mouse over the status button opens a popup with additional information. In general:

- The Master host must not display any red dots. If the Master host has one or more error(s), then the Kubernetes cluster may not function properly.
- If all of the dots for a Worker host are red, then that host will not be able to provide resources to the cluster. This situation typically occurs because the host has been powered off, has lost network connectivity, or because HPE Ezmeral Runtime Enterprise is not properly installed.
- A Worker host with some red and some green dots may cause some Kubernetes cluster operations to fail, unless the errors are transient conditions caused by the host powering on or regaining network connectivity.

Please generate a support bundle and then contact Hewlett Packard Enterprise Technical Support if a host that is reporting service errors meets all of the following criteria:

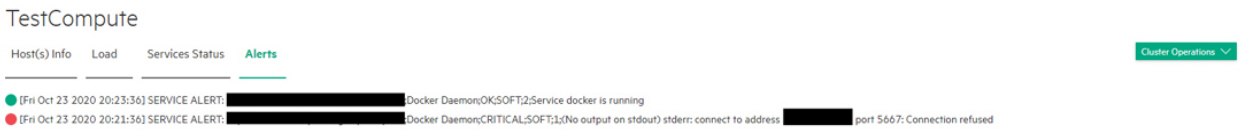
- HPE Ezmeral Runtime Enterprise is completely installed.
- The host is powered on.
- The host has network connectivity.

See The Support/Troubleshooting Screen and Generating a Support Bundle.

Alerts Tab

NOTE: This tab is not available for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

The **Alerts** tab displays any alert messages from the Caching Node, Data Server, and Management services.



The following alerts appear in this tab:

- **Notifications:** Routine messages. A green dot appears next to each routine notification.
- **Error:** A minor error has occurred. A gray dot appears next to each error notification.
- **Warning:** A serious error has occurred. An orange dot appears next to each warning notification.
- **Critical:** A critical error has occurred. A red dot appears next to each critical notification.

NOTE: The presence of non-routine alerts does not mean that HPE Ezmeral Runtime Enterprise will not function normally.

See Troubleshooting Overview for assistance diagnosing and resolving errors.

Creating a New Kubernetes Cluster

Use this procedure to create a Kubernetes cluster that is not implementing HPE Ezmeral Data Fabric on Kubernetes.

Prerequisites

- If applicable, you have enabled Platform HA protection. See Enabling Platform High Availability on page 740.

- If you are using an air-gapped configuration, you must configure air gap settings before creating any Kubernetes clusters. See Air Gap Tab on page 799.
- You have installed the Kubernetes hosts. See Installing Kubernetes Hosts on page 528.
- If you want Kubernetes clusters to use storage provided by HPE Ezmeral Data Fabric on Kubernetes, then you must create the Data Fabric cluster before creating other Kubernetes clusters. See Creating a New Data Fabric Cluster on page 611. (Not available in HPE Ezmeral Runtime Enterprise Essentials.)
- The system is not in Lockdown mode. See Lockdown Mode on page 916

  Creating or editing a Kubernetes cluster while the site is in Lockdown mode can result in errors related to the cluster connections to services, or in service endpoints not being displayed for that Kubernetes cluster.
- **Required access rights:** Kubernetes Administrator

About This Task

This process consists of the following steps:

- Step 1: Host Configurations
- Step 2: Cluster Configuration
- Step 3: Authentication
- Step 4: Application Configurations (Not available in HPE Ezmeral Runtime Enterprise Essentials.)
- Step 5: Summary

The images in this article are taken from an existing Kubernetes cluster in order to provide real-world examples. The screens you see when creating a new Kubernetes cluster will be identical, except that all fields and other options will be blank.

CAUTION:

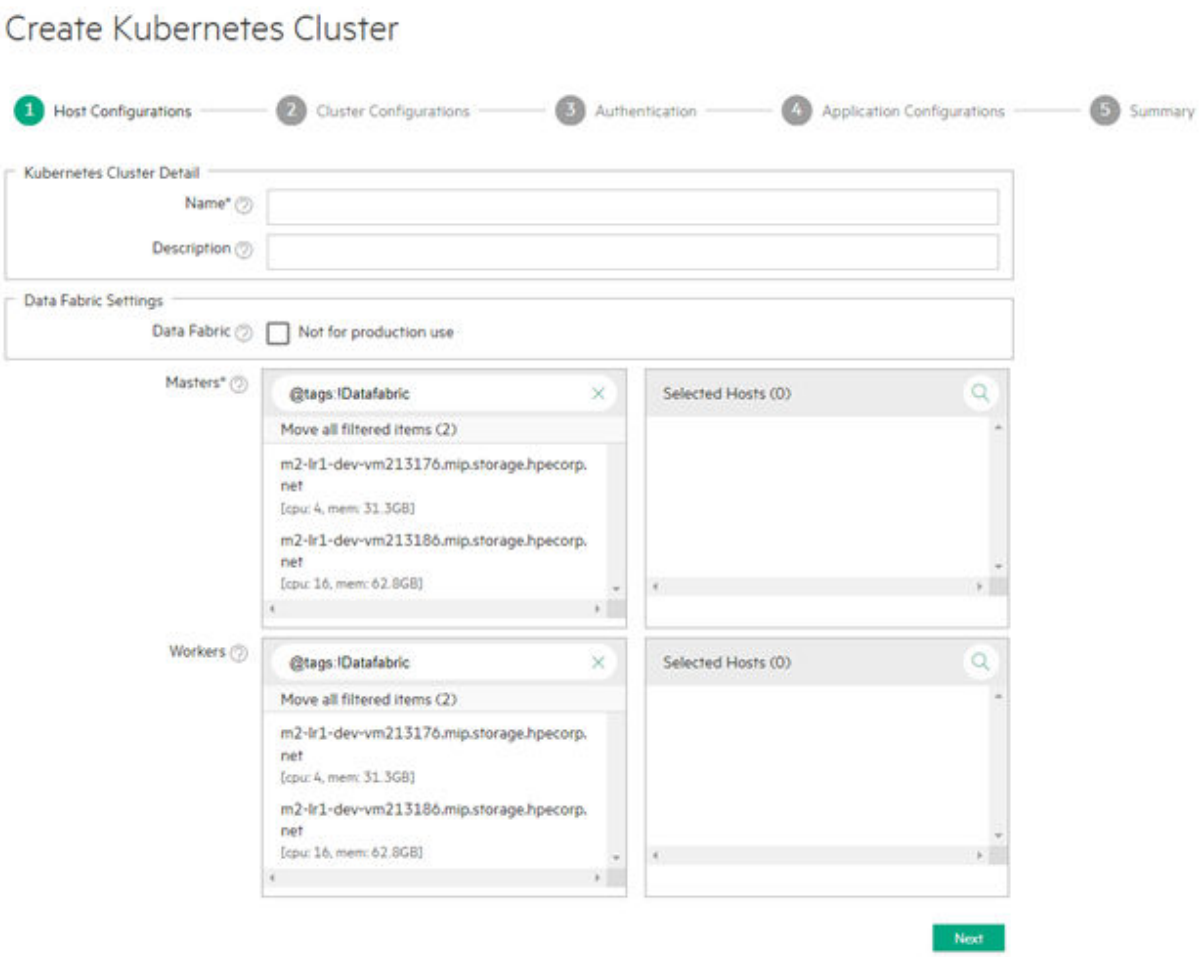
Kubernetes cluster certificates are created with a one-year duration. If the certificates are allowed to expire, the cluster will become unuseable until the certificates are manually re-generated. To prevent this situation from occurring, see Kubernetes Certificate Management on page 444.

Step 1: Host Configurations

To begin creating a new Kubernetes cluster:

1. Open the **Kubernetes Clusters** screen and click **Create Kubernetes Cluster**.

   The **Step 1: Host Configurations** screen appears.



2. Enter a name for the new Kubernetes cluster in the **Name** field.
3. Enter a brief description of the new Kubernetes cluster in the **Description** field.
4. Ensure that the **DataFabric** check box is clear (not checked).

   CAUTION:

   Checking the **DataFabric** check box will attempt to create an HPE Ezmeral Data Fabric on Kubernetes cluster, as described in Creating a New Data Fabric Cluster on page 611. Only one Data Fabric cluster may exist in an HPE Ezmeral Runtime Enterprise deployment.

5. In the **Masters** row of the **Hosts** table, hover the mouse over a host in the **Available** column. You may also search for a host by name, tag, etc. by entering your desired search term in the field and then clicking the **Search** icon (magnifying glass).
6. A right arrow appears.

7. Move the mouse to this arrow, and then click the arrow.

   The selected host moves from the **Available Hosts** column to the **Selected Hosts** column. If you make a mistake, you may hover the mouse over a selected host and then click the left arrow to move it back to the **Available Hosts** column.

   To provide High Availability protection for the Kubernetes cluster, you must select three or more Master hosts. Hewlett Packard Enterprise recommends that you select an odd number of control plane ("master") hosts in order to have a quorum with the best failure tolerance and least chance of a "split brain" failure condition.

   For more information about quorums, failure tolerance, and etcd clusters, see Failure Tolerance in the etcd documentation (link opens an external website in a new browser tab or window).

   By default, a taint is placed on the Master hosts that prevents them from being able to run pods. If you want these hosts to be able to run pods, you must untaint the hosts as described in the Kubernetes documentation here (link opens an external web site in a new browser tab/window).

8. Repeat Steps 4 and 5 for the Worker Hosts. You can add as many Worker hosts as needed to this cluster.

   NOTE: If you are installing an add-on such as Istio (see Add-Ons Overview), then you might need to select hosts with the appropriate tag assignments. Please see the appropriate add-on documentation for additional information.

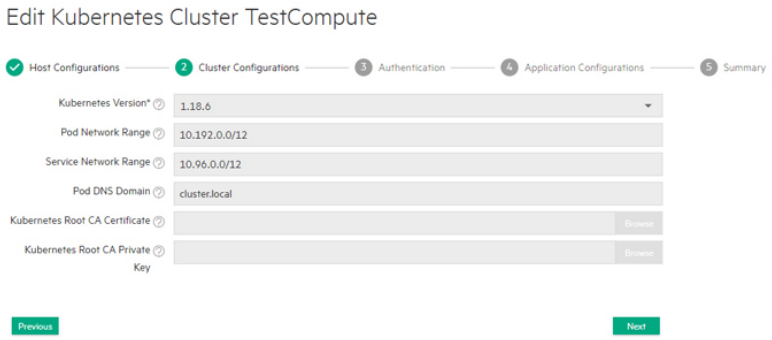
   This feature is not available in HPE Ezmeral Runtime Enterprise Essentials.

   NOTE: You can search for hosts by clicking the **Search** icon (magnifying glass) above any of the four cells in the **Hosts** table and then typing any portion of the hostname. The list of hosts automatically refreshes as you type.

9. Click **Next**.

Step 2: Cluster Configuration

The **Step 2: Cluster Configuration** screen appears.



1. Use the **Kubernetes Version** menu to select the version of Kubernetes to install on the new cluster.

   If you select a version of Kubernetes that is not supported for new cluster creation, an error message is displayed.

2. Enter the network range and mask to use for the pods in this cluster in the **Pod Network Range** field.

   The Calico and Flannel Kubernetes CNI plug ins are pre-installed and configured, and defaults are provided for the Pod CIDR that is within a private range. Ensure that the range of the Pod-IP-address does not conflict or overlap with other ranges—your internal network range, or the service network range—that are already in use.

   ATTENTION: If there is a conflict or overlap in the range, pods will not be able to contact any of the internal hosts whose IP addresses fall within the pod network range.

   Check the **Choosing IP Address** section here for additional information (the link opens an external website in a new browser tab/window).).

3. Enter the network range and mask to use for the endpoint services in this cluster in the **Service Network Range** field.

   The Calico and Flannel Kubernetes CNI plugins are pre-installed and configured, defaults are provided for the Pod CIDR that is within a private range. Ensure that the range of the Pod-IP-address does not conflict or overlap with other ranges—your internal network range, or the service network range—that are already in use.

   ATTENTION: If there is a conflict or overlap in the range, pods will not be able to contact any of the internal hosts whose IP addresses fall within the pod network range.

   Check the **Choosing IP Address** section here for additional information (the link opens an external website in a new browser tab/window).

4. Enter the DNS domain to use for the service endpoints in this cluster in the **Pod DNS Domain** field.

5. Enter the path to the Kubernetes root CA certificate in the **Kubernetes Root CA Certificate** field.

   This is the certificate authority that Kubernetes will use to generate the certificates needed for various Kubernetes components, such as `etcd` and `auth proxy`/`front-proxy`. Clicking the **Browse** button opens a standard **Open** dialog that allows you to navigate to and select the desired file.

6. Enter the path to the Kubernetes root CA private key in the **Kubernetes Root CA Private Key** field.

   This is the private key portion of the root CA certificate. Clicking the **Browse** button opens a standard **Open** dialog that allows you to navigate to and select the desired file.

7. If you are satisfied with your changes, then click **Next** to proceed.

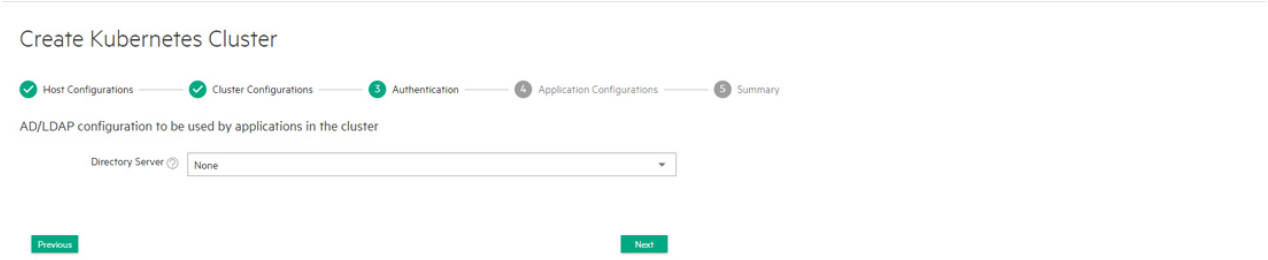
   Alternatively, you can click **Previous** to return to the **Step 1: Host Configurations** screen.

Step 3: Authentication

The **Step 3: Authentication** screen appears. You may either:

- Use the global HPE Ezmeral Runtime Enterprise user authentication.
- Specify user authentication options on a per-Kubernetes-cluster basis.

This is where you enter the AD/LDAP user authentication configuration that will be used by the applications running in this cluster (required for running HPE Ezmeral ML Ops on Kubernetes). Any information entered in this screen is posted as a secret in the cluster.



1. You may either:
   - Click **Next** to use the platform-wide authentication settings.
   - Click the **Copy from Platform Authentication** button to copy the platform-level AD/LDAP authentication to this Kubernetes cluster for further editing, as described in Configuring User Authentication Options.
   - Manually enter authentication settings that will only apply to this Kubernetes cluster, as described in Configuring User Authentication Options.
2. Click **Next** to proceed.

Step 4: Application Configurations

The **Step 4: Application Configurations** screen appears.(Not available in HPE Ezmeral Runtime Enterprise Essentials.)

1. Verify that all of the hosts in the cluster meet the host requirements and the cumulative requirements for all the applications that will be selected, and then select the check boxes for the applications.

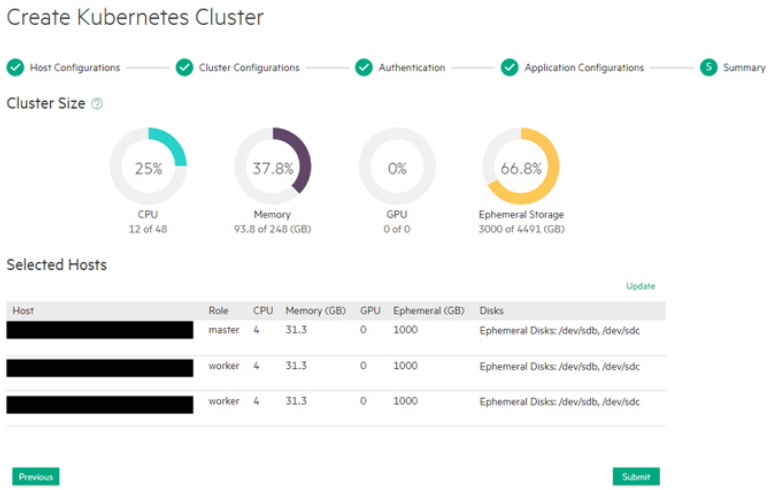
   Not all applications are appropriate for all clusters. For example, Do not select the Istio application when creating or editing a Data Fabric cluster. Istio Service Mesh is not supported on HPE Ezmeral Data Fabric on Kubernetes clusters.

   For information about host requirements, see Kubernetes Host Requirements.

   For information about add-on applications, see Add-ons Overview. Requirements are cumulative; for example, if you add two applications, then all the hosts in the cluster must meet the combined requirements of both applications.
2. Review your application selections, and then click **Next** to proceed. Alternatively, you can click **Previous** to return to the **Step 3: Authentication** screen.

Step 5: Summary

The **Step 5: Summary** screen appears.



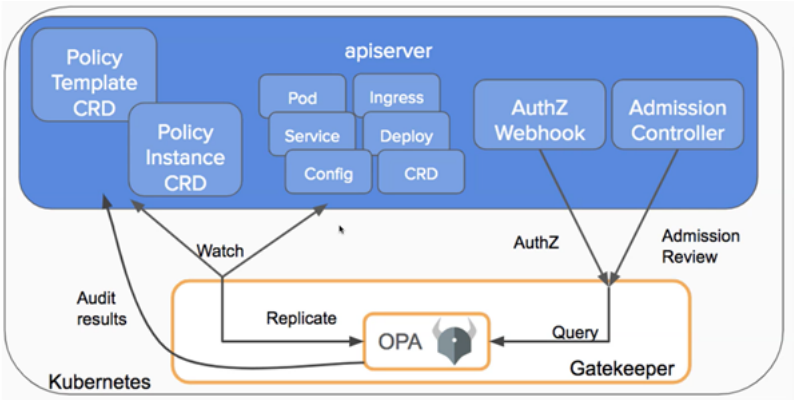
1. Review the summary of resources to be assigned to this cluster, and then either click **Submit** to finish creating the new Kubernetes cluster, or click **Previous** to return to the **Step 4: Application Configurations** screen.

If you need to configure the Open Policy Agent, then see OPA Gatekeeper Policy Configuration on page 469.

OPA Gatekeeper Policy Configuration
Describes configuration of policies using Open Policy Agent (OPA) Gatekeeper, a Rego-based policy engine implemented in HPE Ezmeral Runtime Enterprise as an admission controller for Kubernetes clusters.

HPE Ezmeral Runtime Enterprise leverages OPA Gatekeeper as an admission controller to validate and enforce policies on the cluster. OPA Gatekeeper is installed as a mandatory system add-on, which is automatically created for HPE Ezmeral Runtime Enterprise users. For more information about OPA Gatekeeper, see the official OPA Gatekeeper documentation (link opens an external website in a new browser tab or window).

This feature is not available in HPE Ezmeral Runtime Enterprise Essentials.



You can use Centralized Policy Management to define and manage OPA Gatekeeper policies stored in a Git repository, and apply them to clusters managed by HPE Ezmeral Runtime Enterprise. For information, see Centralized Policy Management on page 336.

Default OPA Policies for Kubernetes Clusters

HPE Ezmeral Runtime Enterprise automatically configures default policies on Kubernetes clusters. Use the `kubectl get constraints` command to list all default policies.

NOTE: To ensure that KubeDirector applications function as expected, HPE Ezmeral Runtime Enterprise automatically configures some default policies as *dry run*. HPE Ezmeral Runtime Enterprise does not enforce these policies, but lists workloads which violate them in the **Violations** tab. For more information on viewing policy violations, see Viewing Policy Violations on page 338.

The following default policies are configured as dry run:

- `psp-non-root-user-and-group`
- `psp-host-network-ports`
- `psp-host-filesystem`

Default policies are as follows:

- The `psp-privileged-container` policy ensures that privileged workloads run only in reserved (system) namespaces.

```
k8spspprivilegedcontainer.constraints.gatekeeper.sh/
psp-privileged-container
```

- The `psp-non-root-user-and-group` policy ensures pods with nonroot user and group run only in reserved (system) namespaces.

```
k8spspnonrootuserandgroup.constraints.gatekeeper.sh/
psp-non-root-user-and-group
```

- The `psp-host-network-ports` policy ensures pods that use host network and host port run only in reserved (system) namespaces.

```
k8spsphostnetworkingports.constraints.gatekeeper.sh/psp-host-network-ports
```

- The `psp-host-filesystem` policy ensures pods that use host file system run only in reserved (system) namespaces.

```
k8spsphostfilesystem.constraints.gatekeeper.sh/psp-host-filesystem
```

Creating OPA Policies

Create OPA Gatekeeper policies with Rego policy language, as described in Rego Policy Language on page 471.

For information and tutorials on using OPA Gatekeeper with Kubernetes, see the official OPA Gatekeeper documentation (link opens an external website in a new browser tab or window).

Applying, Modifying, and Deleting OPA Policies

Apply, modify, and delete OPA Gatekeeper policies, including default policies, as follows:

- After you have created a policy with Rego, apply the constraint and template objects on a Kubernetes cluster as follows. On the Kubernetes master node enter the commands:

```
kubectl apply -f constraint_template.yaml
```

```
kubectl apply -f constraint.yaml
```

- To modify a policy, update the constraint object associated with the policy with the following command:

```
kubectl edit constraint.yaml
```

- To delete a policy, use the following command:

```
kubectl delete constraint.yaml
```

Related concepts

Centralized Policy Management on page 336
Defines centralized policy management and describes the features and benefits of applying policies to Kubernetes clusters managed by HPE Ezmeral Runtime Enterprise. Not available in HPE Ezmeral Runtime Enterprise Essentials.

Related tasks

Viewing Policy Violations on page 338
Describes how to view a detailed log of policy violations and denials triggered on a Kubernetes cluster managed by HPE Ezmeral Runtime Enterprise.

More information

Rego Policy Language on page 471
Describes Rego, the policy language used to write OPA Gatekeeper template objects in HPE Ezmeral Runtime Enterprise.

Rego Policy Language
Describes Rego, the policy language used to write OPA Gatekeeper template objects in HPE Ezmeral Runtime Enterprise.

Rego Policy Language

To write OPA Gatekeeper template objects, you must use Rego. Rego is the policy language for OPA Gatekeeper. For more information about Rego and working with policies and constraints, see these resources:

- Rego
- Policies and Constraints
- How to use Gatekeeper

Organizing Template and Constraint Objects

You can organize pairs of template and constraint objects in two ways:

- **Combine multiple template and constraint objects into one YAML file.** This “one big YAML file” becomes a collection of policies – or one big policy – that includes pairs of templates and constraints. See this example (`onebigpolicy.yaml`).
- **Create a directory of policies with each policy represented as a single YAML file** that contains a pair of constraint and template objects. See this example directory.

View sample YAML policies:

https://github.com/open-policy-agent/gatekeeper-library/tree/master/library/general

Example Policy

The following example policy (`allowedrepo-policy.yaml`) validates all pods in the cluster and ensures that they come from the `openpolicyagent` repo.

In this example, the constraint object appears first, followed by the template object. The template object contains logic for how the policy should be validated. In the template object, lines of code in **bold face** indicate Rego commands. The constraint object contains the values that the template will validate against.

If a pod that is not from the `openpolicyagent` repo is detected, an error is generated.

```
---
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sAllowedRepos
metadata:
  name: repo-is-openpolicyagent
spec:
  match:
    kinds:
    - apiGroups: [""]
      kinds: ["Pod"]
    namespaces:
    - "default"
  parameters:
    repos:
    - "openpolicyagent"
---
apiVersion: templates.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: k8sallowedrepos
  annotations:
    description: Requires container images to begin with a repo string from
a specified
      list.
spec:
  crd:
    spec:
      names:
        kind: K8sAllowedRepos
      validation:
        # Schema for the `parameters` field
        openAPIV3Schema:
          properties:
            repos:
              type: array
              items:
                type: string
  targets:
  - target: admission.k8s.gatekeeper.sh
    rego: |
      package k8sallowedrepos
      violation[{"msg": msg}] {
        container := input.review.object.spec.containers[_]
        satisfied := [good | repo = input.parameters.repos[_] ; good =
startswith(container.image, repo)]
        not any(satisfied)
        msg := sprintf("container <%v> has an invalid image repo
<%v>, allowed repos are %v", [container.name, container.image,
input.parameters.repos])
      }
      violation[{"msg": msg}] {
        container := input.review.object.spec.initContainers[_]
        satisfied := [good | repo = input.parameters.repos[_] ; good =
startswith(container.image, repo)]
        not any(satisfied)
        msg := sprintf("container <%v> has an invalid image repo
<%v>, allowed repos are %v", [container.name, container.image,
```

```
input.parameters.repos])
      }
```

Policy Enforcement Example

After your policies are created and applied to a cluster, you can observe the enforcement of them when operations violate a policy. The following example shows the effect of applying an object that violates multiple policies configured for a cluster:

```
# kubectl apply -f disallowedcontainerlimit.yaml
Error from server ([denied by container-image-must-have-digest] container
<opa> uses an image with a digest <openpolicyagent/opa:0.9.2>
[denied by container-must-have-limits] container <opa> memory limit <2Gi>
is higher than the maximum allowed of <1Gi>
[denied by must-have-probes] Container <opa> in your <Pod> <opa-disallowed>
has no <readinessProbe>
[denied by must-have-probes] Container ,opa> in your <Pod>
<opa-disallowed> has no <livenessProbe>): error when creating
"disallowedcontainerlimit.yaml": admission
 webhook "validation.gatekeeper.sh" denied the request: <denied by
container-image-must-have-digest] container <opa> uses an image without a
digest <openpolicyagent/opa:0.9.2>
[denied by container-must-have-limits] container <opa> memory limit <2Gi>
is higher than the maximum allowed of <1Gi>
[denied by must-have-probes] Container <opa> in your <Pod> <opa-disallowed>
has no <readinessProbe>
[denied by must-have-probes] Container <opa> in your <Pod> <opa-disallowed>
has no <livenessProbe>
```

Overly Restrictive Policies

As with any security system, it is possible to create policies that interfere with normal system operations and that result in unwanted behavior. For example, when creating policies for HPE Ezmeral Runtime Enterprise, setting the root file system directory to “read only” results in numerous errors, because fsmount daemonset pods must have write access to the `/opt/bluedata/share` directory on all of the Kubernetes hosts and the `/opt/bluedata/share` directory inside the pod. One such error is the failure to configure a Kubernetes Web Terminal.

To view a list of policy violations and denials that are occurring in a cluster, see: Viewing Policy Violations on page 338.

Alternatively, to display a JSON-formated list of policy violations that are occurring in a cluster, enter the following command:

```
kubectl get constraints -o json
```

Related concepts

Centralized Policy Management on page 336
Defines centralized policy management and describes the features and benefits of applying policies to Kubernetes clusters managed by HPE Ezmeral Runtime Enterprise. Not available in HPE Ezmeral Runtime Enterprise Essentials.

Related tasks

Viewing Policy Violations on page 338
Describes how to view a detailed log of policy violations and denials triggered on a Kubernetes cluster managed by HPE Ezmeral Runtime Enterprise.

More information

OPA Gatekeeper Policy Configuration on page 469

Describes configuration of policies using Open Policy Agent (OPA) Gatekeeper, a Rego-based policy engine implemented in HPE Ezmeral Runtime Enterprise as an admission controller for Kubernetes clusters.

Troubleshooting OPA Gatekeeper
Describes how to disable Open Policy Agent (OPA) Gatekeeper on a Kubernetes cluster for troubleshooting purposes, and re-enable OPA Gatekeeper after any issues have been corrected.

As part of OPA Gatekeeper deployment, HPE Ezmeral Runtime Enterprise creates an admission webhook. This webhook intercepts requests to the API server, and returns a response to the API server. Depending on the response received, and the policies currently in place, the API server decides whether the request can be fulfilled.

If OPA Gatekeeper is preventing a cluster from operating correctly, this admission webhook can be disabled to remove all OPA Gatekeeper admission checks while the issue is being fixed.

See: https://open-policy-agent.github.io/gatekeeper/website/docs/emergency/

Proceed as follows:

1. Save the definition of the original webhook present in the system with the following command:

   ```
   kubectl get validatingwebhookconfigurations.admissionregistration.k8s.io
   gatekeeper-validating-webhook-configuration -o yaml > webhook.yaml
   ```

2. To disable the admission webhook, enter the following command:

   ```
   kubectl delete
   validatingwebhookconfigurations.admissionregistration.k8s.io
   gatekeeper-validating-webhook-configuration
   ```

   NOTE: While the admission webhook is disabled, OPA Gatekeeper will no longer impose policies in the cluster. To reimpose policies, re-enable OPA Gatkeeper.

3. While the admission webhook is disabled, you can correct any issues you may be experiencing with OPA Gatekeeper.

   For information on debugging OPA Gatekeeper, see: https://open-policy-agent.github.io/gatekeeper/website/docs/debug/.

   For information on OPA Gatekeeper issues, see: https://github.com/open-policy-agent/gatekeeper/issues.

4. After you have fixed any issues, re-apply the admission webhook with the following command:

   ```
   kubectl apply -f webhook.yaml
   ```

NOTE: If you did not save the original webhook before deleting it, re-enable OPA Gatekeeper as follows:

Apply the following YAML manifest on the Kubernetes cluster:

```
apiVersion: admissionregistration.k8s.io/v1beta1
kind: ValidatingWebhookConfiguration
metadata:
  creationTimestamp: null
  labels:
    gatekeeper.sh/system: "yes"
  name: gatekeeper-validating-webhook-configuration
webhooks:
- clientConfig:
    caBundle: Cg==
    service:
      name: gatekeeper-webhook-service
      namespace: gatekeeper-system
      path: /v1/admit
  failurePolicy: Ignore
  name: validation.gatekeeper.sh
  namespaceSelector:
    matchExpressions:
    - key: admission.gatekeeper.sh/ignore
      operator: DoesNotExist
  rules:
  - apiGroups:
    - '*'
    apiVersions:
    - '*'
    operations:
    - CREATE
    - UPDATE
    resources:
    - pods
  sideEffects: None
  timeoutSeconds: 3
- clientConfig:
    caBundle: Cg==
    service:
      name: gatekeeper-webhook-service
      namespace: gatekeeper-system
      path: /v1/admitlabel
  failurePolicy: Fail
  name: check-ignore-label.gatekeeper.sh
  rules:
  - apiGroups:
    - ""
    apiVersions:
    - '*'
    operations:
    - CREATE
    - UPDATE
    resources:
    - namespaces
  sideEffects: None
  timeoutSeconds: 3
```

.

Importing an External Kubernetes Cluster

Importing an external Kubernetes cluster is not supported at this time. Kubernetes clusters must be created using HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise provides a unified control plane to manage a fleet of Kubernetes clusters located both on-premises as well as those running in the managed cloud providers such as Elastic Kubernetes Service (EKS) in AWS. The dashboard allows you to monitor cluster health and usage statistics, and to provide Identity and Access Management across clusters.

Once an external cluster is imported and registered within the platform, you may then create tenants, set up Role Based Access Controls, deploy applications, and use many of the same features as a Kubernetes cluster created within HPE Ezmeral Runtime Enterprise. Importing external Kubernetes clusters is not supported in HPE Ezmeral Runtime Enterprise Essentials.

You can import clusters from:

- **Microsoft Azure:** See AKS.
- **Amazon Elastic Kubernetes Service:** See EKS.
- **Google Kubernetes Engine:** See GKE.
- **Enterprise Pivotal Container Service:** See PKS.

Be sure to also see Limitations at the bottom of this article for information on the limitations that apply to imported Kubernetes clusters compared to those created within HPE Ezmeral Runtime Enterprise.

Importing the Cluster

This section describes importing an existing Kubernetes cluster.

Requirements

The following requirements must be met in order to import an external Kubernetes cluster from EKS:

- The external Kubernetes cluster must already exist before you begin the import process. Please refer to your provider's documentation for instructions.
- The cloud-based Kubernetes clusters requires a private endpoint, and all of the pods in this cluster must be able to access the HPE Ezmeral Runtime Enterprise control plane.
- A bidirectional VPN must be configured to allow communications between HPE Ezmeral Runtime Enterprise and the external Kubernetes cluster, if applicable. Please refer to your provider's documentation for instructions.
- Domain name resolution must be configured so that the pods launched on the external cluster can resolve the hostnames of the HPE Ezmeral Runtime Enterprise control plane. This can be achieved in multiple ways. Please refer to the Kubernetes documentation and your provider's documentation for instructions (links open external websites in new browser tabs/windows).
- The external Kubernetes cluster must be accessible using `kubectl` to get the information needed to import that cluster.
- The user accessing the external Kubernetes cluster must have privileges to create and elevate a service account.
- The external Kubernetes cluster must include a service account with a cluster-admin role binding. See Step One: Create the Service Account, below.
- A Default StorageClass must already be set up for the external Kubernetes Cluster. Please refer to your cloud provider documentation for instructions.

CAUTION: HPE EZMERAL CONTAINER PLATFORM ONLY SUPPORTS EXTERNAL KUBERNETES CLUSTERS THAT HAVE A DEFAULT STORAGECLASS CONFIGURED IN PRODUCTION.

Step 1: Gather Necessary Information

Execute the following commands on any host that can access the external Kubernetes cluster using a `kubeconfig` received by following your cloud provider's instructions.

1. Create a service account by executing the command below. You can choose any namespace for the account as long as the rest of the commands also specify the same namespace. This example uses the default namespace.

   ```
   kubectl create serviceaccount abc123
   ```

2. Assign the `cluster-admin` role binding for the newly created service account. The default in the following command is the namespace where the service account was created.

   ```
   kubectl create clusterrolebinding
   add-on-cluster-admin --clusterrole=cluster-admin --serviceaccount=default
   :abc123
   ```

3. Obtain the CA certificate and bearer token data by executing the following commands. You will provide the contents of the token.base64 and ca.crt.base64 files to the HPE Ezmeral Runtime Enterprise web interface when importing the cluster.

   ```
   SA_TOKEN=`kubectl get serviceaccount/abc123 -o
   jsonpath={.secrets[0].name}`
   kubectl get secret $SA_TOKEN  -o jsonpath={.data.token} > token.base64
   kubectl get secret $SA_TOKEN  -o jsonpath={'.data.ca\.crt'} >
   ca.crt.base64
   ```

4. The pod DNS domain name is generally set to `cluster.local` by default on all external clusters. If you are unsure, then execute the following command to check the DNS Corefile:

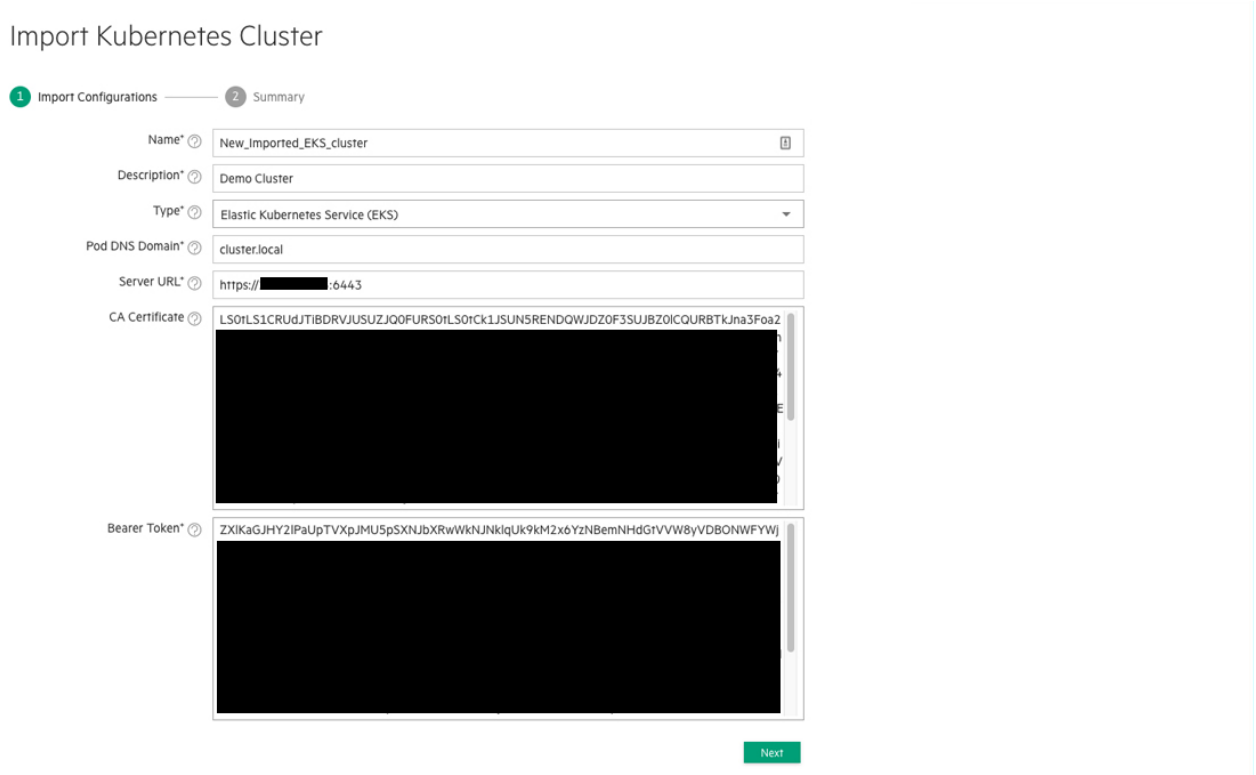
```
kubectl describe configmaps/coredns  -n kube-system
```

Step Two: Import the Cluster

To import an external Kubernetes cluster:

1. Click the **Import Kubernetes Cluster** button in the **Kubernetes Clusters** screen.

   The **Step 1: Import Configurations** screen appears.



2. Enter the following information in the appropriate fields:
   - **Name:** Provide a name for the external Kubernetes cluster that you are importing.
   - **Description:** Enter a brief description for this cluster.
   - **Type:** Use this pull-down menu to select **Generic**, **AKS**, **EKS**, **GKE**, or **P**KS, as appropriate.
   - **Pod DNS Domain:** Enter the FQDN of the pods in the imported cluster.
   - **Server URL:** Enter the complete URL to the server that hosts the cluster, including the port number.
   - **CA Certificate:** Paste the CA certificate that you received by executing the script in Step 1: Gather Necessary Information into this field.
   - **Bearer Token:** Paste the bearer token that you received by executing the script in Step 1: Gather Necessary Information into this field.
3. Click **Next**.

   The **Step 2:Summary** screen appears.

4. Review the summary of cluster import parameters, and then either click **Submit** to finish importing the external Kubernetes cluster or click **Previous** to return to the **Step 2: Application Configurations** screen.

AKS

Please visit the following links for additional background information when importing an existing Kubernetes cluster from Microsoft Azure Kubernetes Service (AKS) (links open external websites in a new browser tab/window):

- AKS Networking Concepts
- AKS Overview
- AKS Storage Options

EKS

Please visit the following links for additional background information when importing an external Kubernetes cluster from EKS (links open external websites in a new browser tab/window):

- General Amazon EKS documentation
- Amazon EKS VPN connections
- Amazon EKS storage classes

GKE

Please visit the following links for additional background information when importing an existing Kubernetes cluster from Google Kubernetes Engine (GKE). Please visit the following links for additional background information (links open external websites in a new browser tab/window):

- GKE Overview
- Creating a VPC-Native Cluster
- Setting the Default Storage Class

PKS

Please visit the following links for additional background information when importing an existing Kubernetes cluster from Enterprise Pivotal Container Service (PKS). Please visit the following link for additional background information (link opens an external website in a new browser tab/window):

- PKS General Information

PKS clusters are created inside your local environment. Thus:

- VPC and VPN are not required.
- DNS and default Storage Class are required.

Limitations

The following limitations apply to imported Kubernetes clusters:

- The **Kubernetes Cluster Details** screen does not display the **Services** and **Alerts** tabs.
- FS mounts are not available. A member of a tenant in an imported cluster therefore cannot use the **Kubectl** tab of the **Kubernetes Applications** screen to apply YAML or JSON files from a shared filesystem. See The Kubernetes Applications Screen.
- HPE Ezmeral Runtime Enterprise does not manage the lifecycle (expanding, shrinking, upgrading, or deleting) of the external Kubernetes cluster. You must make these changes using the provider's console.
- Deleting an imported Kubernetes cluster only unregisters it from HPE Ezmeral Runtime Enterprise; it does not delete the cluster from the provider.

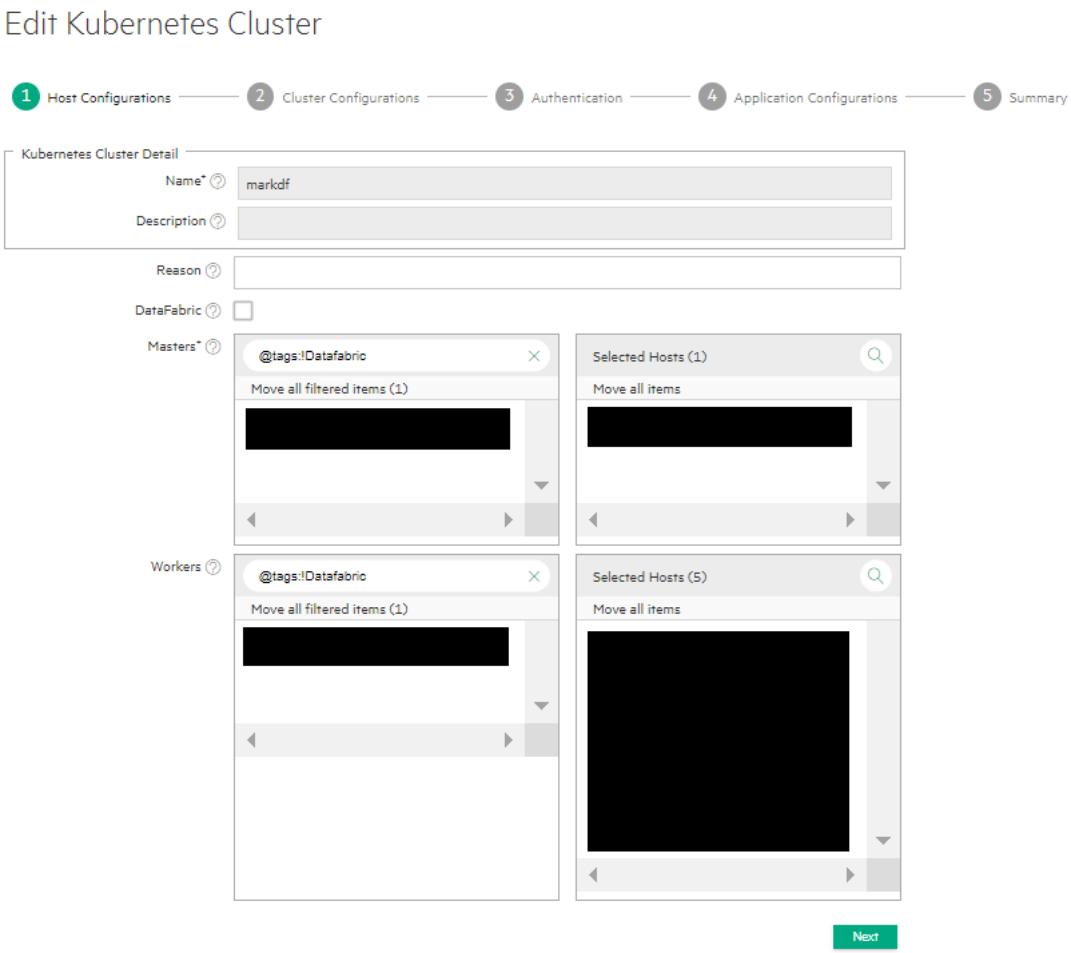
Editing an Existing Kubernetes Cluster

NOTE:

Do not enter Lockdown mode when creating or editing a Kubernetes cluster. Creating or editing a Kubernetes cluster while the site is in Lockdown mode can result in errors related to the cluster connections to services, or in service endpoints not being displayed for that Kubernetes cluster.

Clicking the **Edit** button (pencil) for a Kubernetes cluster in the **Kubernetes Clusters** screen opens the **Step 1: Host Configurations** screen for that cluster.

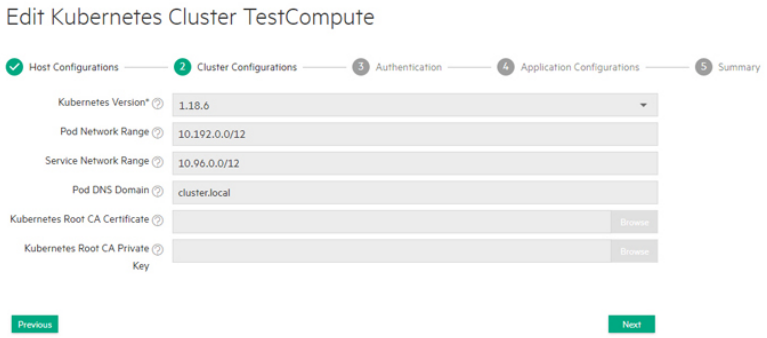
last-updated: May 06, 2024



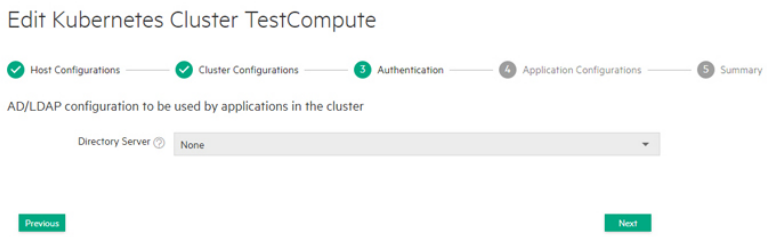
1. You may edit some or all of the following options on this screen:
   - Enter a description for this Kubernetes cluster in the **Description** field, if desired.
   - Number of Master and Worker hosts in the cluster. You must select an odd number of Master hosts in order to have a quorum. See Expanding or Shrinking a Kubernetes Cluster.

   NOTE: If you are installing an add-on such as Istio (see Add-Ons Overview), then you may need to select hosts with the appropriate tag assignments. Please see the appropriate add-on article for additional information. (Not available in HPE Ezmeral Runtime Enterprise Essentials.)

2. Click **Next** to open the **Step 2: Cluster Configurations** screen. All of the fields on this screen are read-only.



3. Click Next to open the **Step 3: Authentication** screen. All of the fields on this screen are read-only.



4. Click **Next** to open the **Step 4: Application Configurations** screen. (Not available in HPE Ezmeral Runtime Enterprise Essentials.)

   The available add-on applications may vary from those shown below.

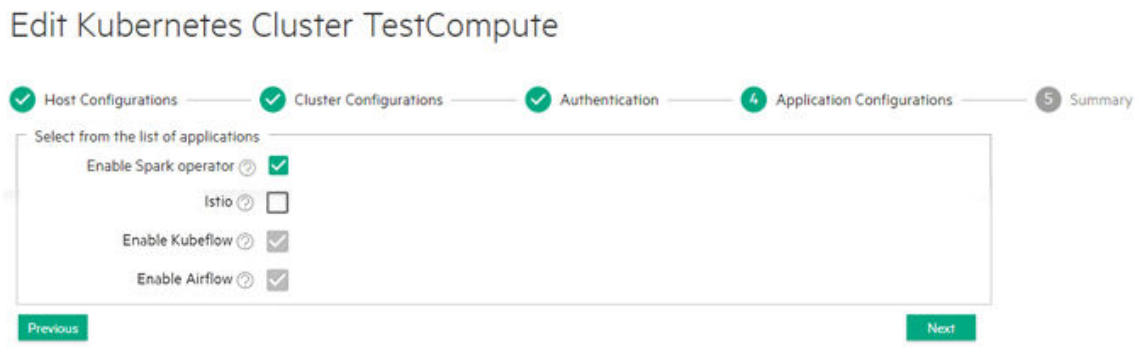
   To remove an application, clear the check box for that application.

   To add an application, verify that all of the hosts in the cluster meet the host requirements and the cumulative requirements for all the applications that will be selected, and then select the check box for that application.

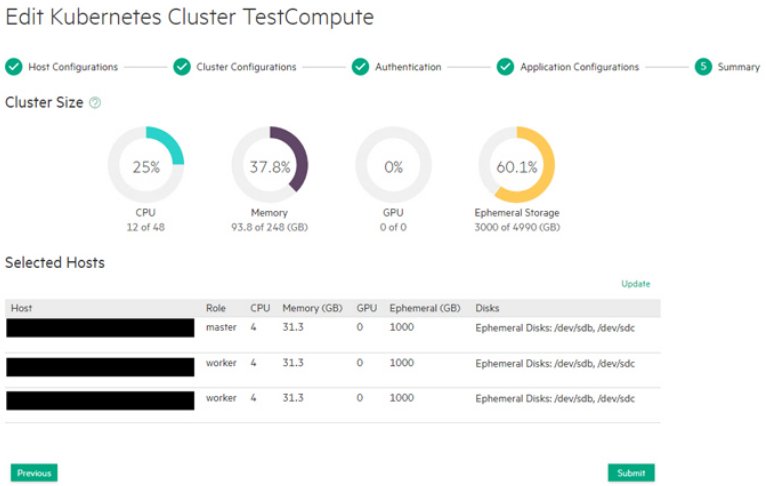
   Not all applications are appropriate for all clusters. For example, Do not select the Istio application when creating or editing a Data Fabric cluster. Istio Service Mesh is not supported on HPE Ezmeral Data Fabric on Kubernetes clusters.

   For information about host requirements, see Kubernetes Host Requirements.

   For information about add-on applications, see Add-ons Overview. Requirements are cumulative; for example, if you add two applications, then all the hosts in the cluster must meet the combined requirements of both applications in addition to the applications that are already installed.



5. Click **Next** to open the **Step 5: Summary** screen.



6. Review the summary of resources to be assigned to this cluster, and then either click **Submit** to finish editing the existing Kubernetes cluster or click **Previous** to return to the **Step 3: Application Configurations** screen.

Expanding or Shrinking a Kubernetes Cluster

Prerequisites

The hosts that will added to the Kubernetes cluster must be added to the deployment before you can add them to the Kubernetes cluster.

If you are expanding a Kubernetes cluster that was created on a previous version of HPE Ezmeral Runtime Enterprise and has not been migrated to use the Hewlett Packard Enterprise distribution of Kubernetes,you must override the default Kubernetes runtime when you add the new hosts to the deployment.

See Kubernetes Worker Installation Overview on page 528.

About this task

You can expand or shrink the size of an internal (not imported) Kubernetes cluster while editing that cluster.

If you are expanding a Kubernetes cluster that was created on a previous version of HPE Ezmeral Runtime Enterprise and has not been migrated to use the Hewlett Packard Enterprise distribution of Kubernetes, you must select hosts that use the Docker container runtime (see Prerequisites). Otherwise, select hosts that use the containerd runtime.

Procedure

1. In the **Kubernetes Clusters** screen (see The Kubernetes Clusters Screen), click the **Edit** icon (pencil).

   The **Step 1: Hosts Configuration** screen appears.

2. In the **Masters** row of the **Hosts** table, hover the mouse over a host in the **Available** column.

   A right arrow appears.

3. Click the arrow.

   The selected host moves from the **Available Hosts** column to the **Selected Hosts** column. If you make a mistake, you may hover the mouse over a selected host and then click the left arrow to move it back to the **Available Hosts** column. You must select an odd number of Master hosts in order to have a quorum (e.g. 3, 5, 7, etc.). Selecting three or more Master hosts provides High Availability protection for the Kubernetes cluster.

> **TIP:** You can search for hosts by clicking the **Search** icon (magnifying glass) above any of the four cells in the **Hosts** table and then typing any portion of the hostname. The list of hosts automatically refreshes as you type.

By default, a taint is placed on the Master hosts that prevents them from being able to run pods. You must untaint these hosts if you want them, to be available to run pods, as described here (link opens an external web site in a new browser tab/window).

- Expanding from 1 Master host to 3 Master hosts adds High Availability protection to the Kubernetes cluster.
- Shrinking from 3 Master hosts to 1 Master host removes High Availability protection from the Kubernetes cluster. Further, if the Master hosts have been untainted, attempting to shrink the number of Master hosts may fail if this would not leave sufficient resources for the pods within that cluster.

4. Repeat Steps 4 and 5 for the Worker Hosts.

You may add or remove as many Worker hosts as needed to this cluster. Removing Worker hosts may fail if doing so would not leave sufficient resources for the pods within that cluster.

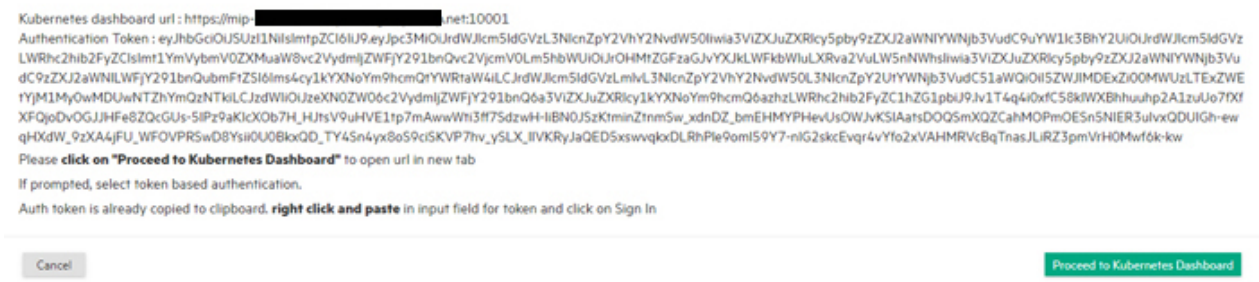
Accessing the Kubernetes Dashboard

To access the Kubernetes dashboard:

1. Accessing this function varies by your assigned role:

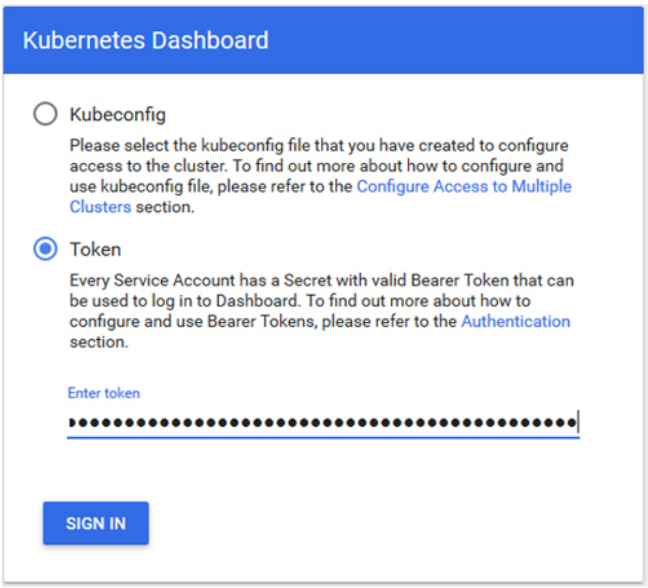
- If you are a Platform Administrator user, you may click the **Access Kubernetes Dashboard** icon (screen) for the desired cluster in the **Clusters** screen. See The Kubernetes Clusters Screen.
- Platform and Kubernetes Cluster Administrator users can select **Access Kubernetes Dashboard** from the **Cluster Operations** menu in the **Cluster Details** screen. See Viewing Kubernetes Cluster Details and The Kubernetes Cluster Details Screen.

  A popup appears with the authentication token.



2. Click **Proceed to Kubernetes Dashboard**. This copies the token to your clipboard.

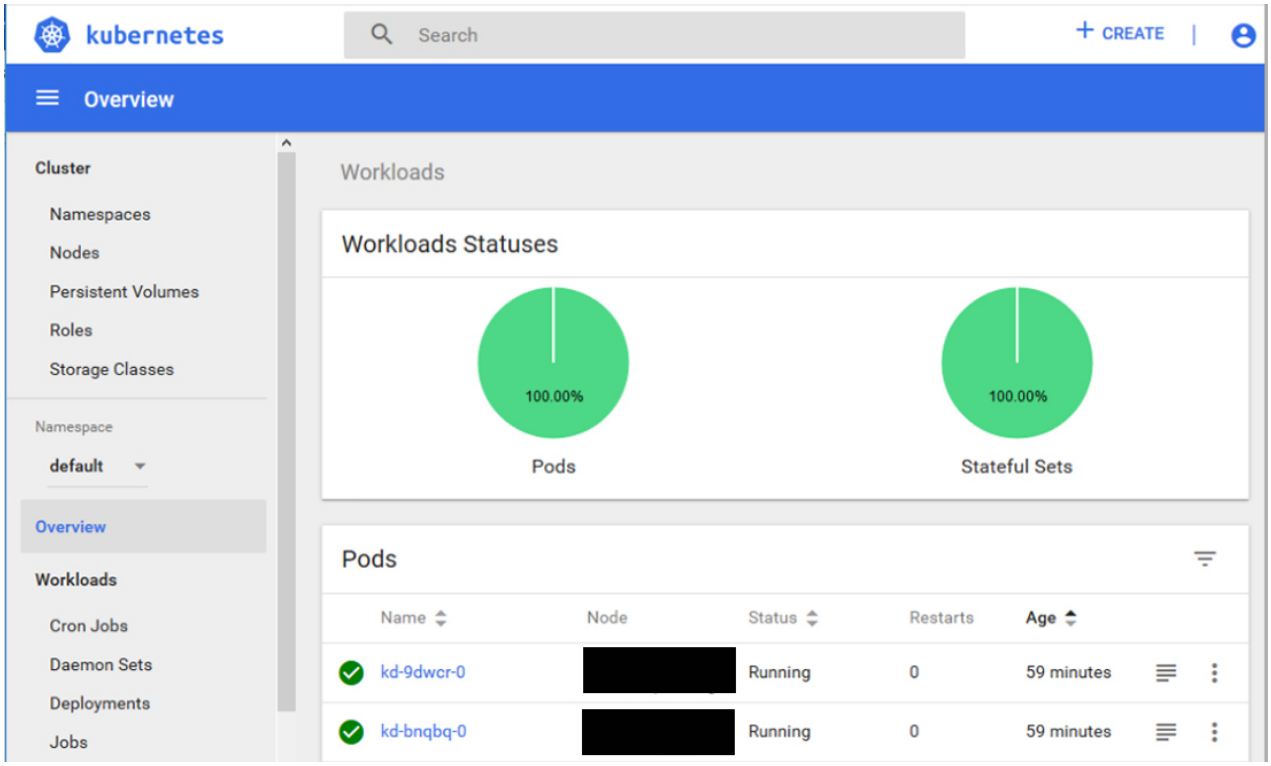
The **Kubernetes Dashboard** sign-on appears.



3. Check the **Token** radio button, and then paste the token into the **Enter Token** field.

4. Click **Sign In**.

The Kubernetes dashboard appears.



If you are having issue accessing the Kubernetes Dashboard on a subsequent attempt, then:

1. Delete your browser cache and cookies.

2. Restart the browser.

3. Restart the Kubernetes dashboard.

Downloading Admin Kubeconfig

Kubernetes Cluster Adminstrator and Platform Adminstrator users can download the Admin Kubeconfig file for a cluster, as follows:

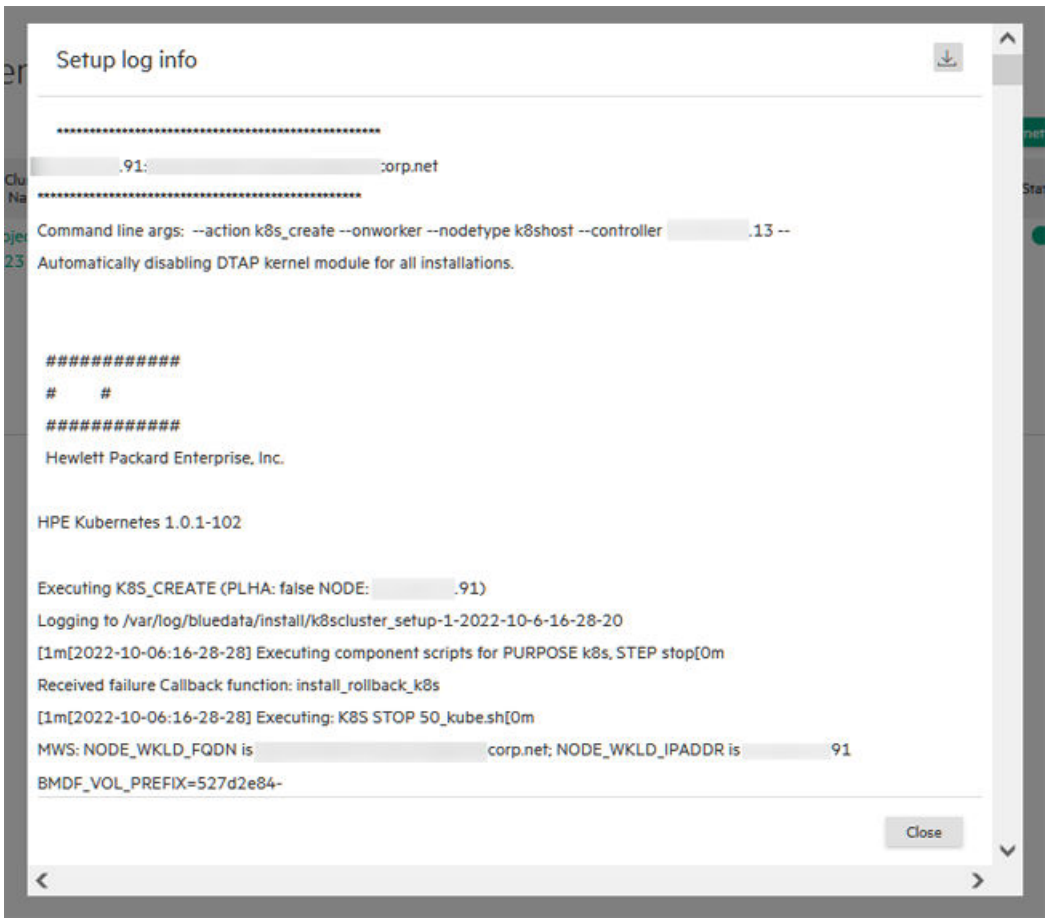
- **Cluster Administrator:** Select **Download Admin Kubeconfig** from the **Cluster Operations** pull-down menu in the **Kubernetes Cluster Details** screen. See Viewing Kubernetes Cluster Details.
- **Platform Administrator:** Click the **Download Admin Kubeconfig** icon (down arrow) for the desired cluster in the **Kubernetes Clusters** screen. See The Kubernetes Clusters Screen.

The downloaded file will look something like this:

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: <certificate goes here>
    server: https://mip.storage.enterprise.net:10000
  name: k8s-1
contexts:
- context:
    cluster: k8s-1
    user: kubernetes-admin
  name: kubernetes-admin@k8s-1
current-context: kubernetes-admin@k8s-1
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: <certificate goes here>
    client-key-data: <key goes here>
```

Viewing the Kubernetes Cluster Setup Log

In the **Kubernetes Clusters** screen (see The Kubernetes Clusters Screen), clicking the **Setup Log Info** icon (envelope) in the **Actions** column opens the setup log for the selected Kubnernetes cluster. The `kubeadm` tools and framework are used to create and update Kubernetes clusters while also including specific components to monitor and manage the lifecycle of these clusters.



Upgrading Kubernetes

Before Upgrading Kubernetes

There are some situations in which you will need help from Hewlett Packard Enterprise support or in which you must perform workaround tasks.

HPE Ezmeral Data Fabric on Kubernetes is Deployed

If you are not upgrading Data Fabric, do the following:

1. Bring the Data Fabric cluster to offline status by performing the steps in Shutting Down a Data Fabric Cluster on page 618 to make the cluster offline.
2. Upgrade the Kubernetes version by performing steps of Upgrade Procedure in this page.
3. After you perform the Upgrade, Restart the Data Fabric cluster by performing the steps in Restarting the Data Fabric Cluster on page 620.

Kubeflow Add-On is Deployed

If you are upgrading a Kubernetes cluster in HPE Ezmeral Runtime Enterprise, the Kubeflow add-on must be version 1.6 or greater.

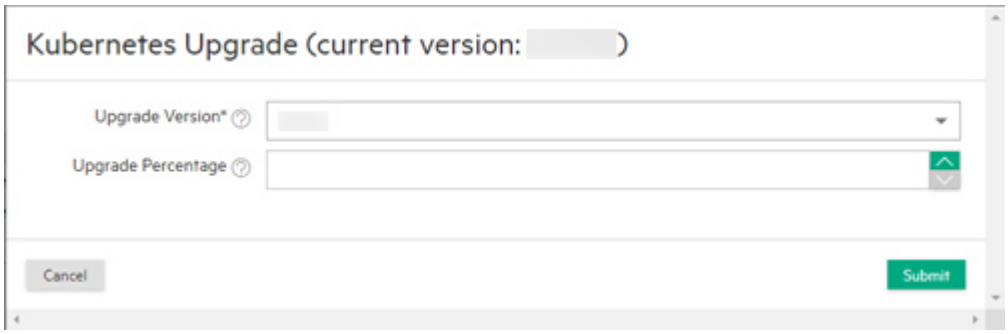
To check your version of Kubeflow, look in the Kubeflow dashboard. See Accessing the Kubeflow Dashboard on page 359.

If your Kubernetes version is lower than 1.6, upgrade the Kubeflow and Istio add-ons before proceeding

	with the Kubernetes cluster upgrade. See Upgrading Kubernetes Add-Ons on page 900.
Istio Add-On is Deployed	When you deploy Istio add-on on Kubernetes cluster, one or more Worker nodes will fail to upgrade the Kubernetes version. The Kubernetes version upgrade fails with the following errors: • `Warning: one or more workers failed to upgrade` on the **Kubernetes Cluster** screen. • `Upgrade error: Failed to drain node` error at the individual **Kubernetes Host Status** screen To resolve the errors, see EZCP-1608 in Issues and Workarounds on page 15
Pods use PVCs provisioned by the CSI driver	If pods on this cluster use a persistent volume claim (PVC) provisioned through HPE CSI driver 1.0.x or 1.1.x, before you upgrade from Kubernetes 1.18.x, upgrade the HPE CSI driver to version `1.2.5-1.0.5`. For instructions, see Upgrading the CSI Plug-In on page 635. If you do not upgrade the CSI driver, pods fail to come up, and the **Kubernetes Cluster** screen displays the message: `one or more workers failed to upgrade`.

Upgrade Procedure

In the **Kubernetes Clusters** screen (see The Kubernetes Clusters Screen), clicking the **Upgrade Kubernetes** icon (up arrow) in the **Actions** column opens the **Kubernetes Upgrade** popup.



To upgrade the Kubernetes version:

- Use the **Upgrade Version** pull-down menu to select the new version of Kubernetes to install. This menu is disabled if no new version of Kubernetes is available.
- Use the **Upgrade Percentage** pull-down menu to specify the number of Kubernetes Worker nodes that will be upgraded at any one time. The default selection is 20%. For example, if the cluster has five Worker nodes, then they will be upgraded one at a time; a cluster with 15 Worker nodes will be upgraded 3 at a time, and so forth. This menu is disabled if no new version of Kubernetes is available.

  NOTE: For Kubernetes Data Fabric clusters in HPE Ezmeral Runtime Enterprise 5.3.5 or later, the percentage setting is ignored. Clusters are upgraded one node at a time.

The status of Worker hosts changes to **Upgrading** during the upgrade process. During the upgrade:

- The Master hosts are upgraded first.

- If any of the Master hosts fails to upgrade, then the upgrade will be rolled back, the Kubernetes version will remains unchanged, and the status of the Kubernetes cluster will change to **WARNING**.
- The Worker hosts are upgraded in batches according to the percentage setting after the Master hosts have been successfully upgraded.
- If any Worker host fails to upgrade, then its status will change to **UPGRADE ERROR** and its Kubernetes version remains unchanged.
- You can retry the upgrade on any failed hosts by clicking the **Retry Upgrade** button.

When upgrading Kubernetes or HPE Ezmeral Runtime Enterprise:

- **Kubernetes clusters:** This is a "rolling" process where containers are upgraded sequentially.
- **Control plane (Controller, Shadow, and Arbiter hosts):** This is not a rolling process, and upgrades do not affect running Kubernetes clusters or workloads. However, the authentication proxy will be down for a short period during the upgrade process, which will interrupt access to Kubernetes clusters. Access will be restored shortly.

See General Kubernetes Application/Deployment Issues for assistance if you experience any errors while upgrading Kubernetes.

Managing Kubernetes Admin Users

Clicking the **Cluster Admin Users** icon (person) in the **Kubernetes Clusters** screen (see The Kubernetes Clusters Screen) opens the **Cluster Users** screen for that Kubernetes cluster.

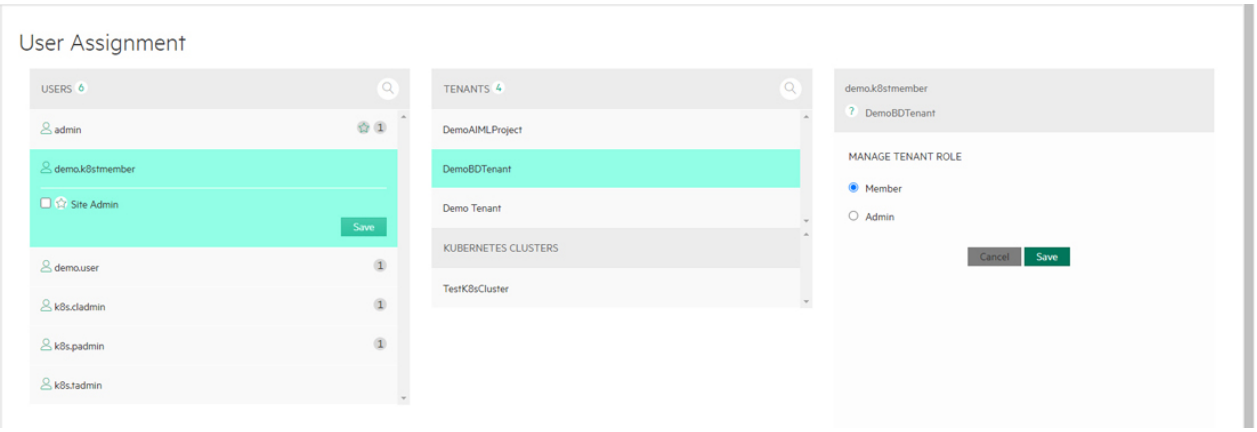
The table on this screen contains the following information for each user who currently has a role within the current Kubernetes cluster:

- **Login Name:** Name that the user uses to log in.
- **Full Name:** Full name of the user.
- **Role:** Role assigned to the user in the current Kubernetes cluster. For Kubernetes Cluster Administrator users, this will say **K8S Admin**.
- **Actions:** Clicking the **Revoke** icon (person) for a user revokes their role within the current Kubernetes cluster. You can also select multiple users and then click the red **Revoke** button to revoke multiple users at once.

To assign the Kubernetes Cluster Administrator for this cluster to one or more users:

1. Click the **Assign** button.

   The **User Assignment** screen appears.



2. Select the user to assign in the **USERS** column on the left side of this screen.

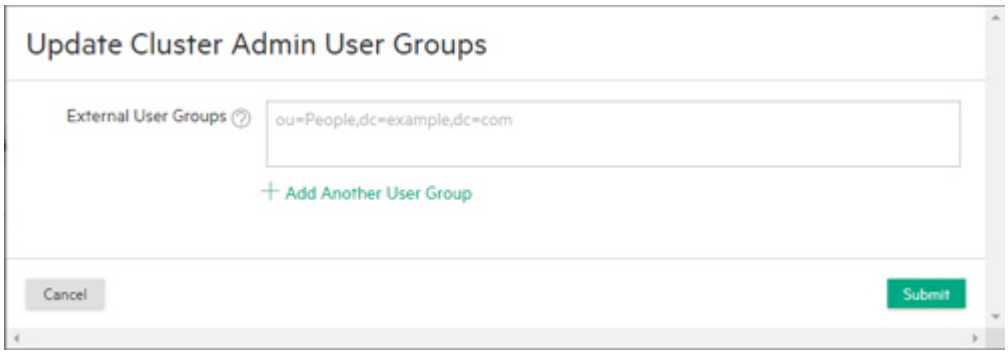
3. Select the Kubernetes cluster to which to assign the user using the **KUBERNETES CLUSTERS** section in the middle of this screen.

4. Check the **K8S Admin** radio button in the **MANAGE CLUSTER ROLE** column on the right side of this screen.

5. Click **Save** to finish assigning the user.

Updating External Kubernetes Cluster Admin Groups

If HPE Ezmeral Runtime Enterprise is configured to use LDAP/AD authentication (see Configuring User Authentication Settings), then the **Update Cluster Admin External Groups** button (persons) appears in the **Actions** column of the **Kubernetes Clusters** screen (see The Kubernetes Clusters Screen). Clicking this button opens the **Update Site Admin User Groups** popup, which allows you to specify LDAP/AD user group(s) that will be assigned the Kubernetes Cluster Administrator role.



To configure the LDAP/AD group(s) that will be assigned the Kubernetes Cluster Administrator role:

1. In the **Kubernetes Clusters** screen, click the **Update Cluster Admin External Groups** button (persons) in the **Actions** column for the Kubernetes cluster for which you want to make the assignment.

2. Enter the first group to associate with the tenant in the field that appears, as shown in the example above.

3. To add another group, click the **Add Another User Group** icon (plus sign) to the right of the field.

4. To remove a group, click the **Remove Group** icon (minus sign) to the right of the group you want to remove.

When you have finished making your desired changes, click the **Submit** button to close the popup and return to the **Kubernetes Clusters** screen. See The Kubernetes Clusters Screen. HPE Ezmeral Runtime Enterprise will confirm the exact DN of the group in the LDAP or AD server and use that DN to do group membership checks on users.

Deleting a Kubernetes Cluster

Deleting a Kubernetes cluster will remove the cluster from the deployment, as follows:

Local cluster	Any tenants or pods that are currently running will be terminated, and no output will be captured.
External cluster	The cluster information is removed from the deployment, but the cluster itself and all tenants/pods therein remain unchanged. You can re-import this cluster as described in Importing an External Kubernetes Cluster.
Data Fabric cluster	• You cannot delete a Data Fabric cluster after it has been registered, as described in HPE Ezmeral Data Fabric as Tenant/Persistent Storage. • Deleting the Data Fabric cluster also deletes the HPE Ezmeral Data Fabric for that cluster. You will receive a confirmation message before this type of cluster can be deleted. • After you delete a Data Fabric cluster, you must reboot the Kubernetes hosts before the hosts can be reused in HPE Ezmeral Runtime Enterprise.

To delete a Kubernetes cluster:

1. Open the **Kubernetes Clusters** screen.
2. Select the cluster to delete, and then click the **Delete** icon (trash can) for that cluster in the **Actions** column.
3. A popup warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without deleting the cluster.
4. If you deleted a Data Fabric cluster, reboot the Kubernetes hosts that were part of the Data Fabric cluster.

CAUTION:

You cannot undelete a Kubernetes cluster. Deleting a Kubernetes cluster immediately ends any tenant(s) and pod(s) running on that cluster.

Add-ons

Kubernetes cluster add-ons provide additional functionality to your HPE Ezmeral Runtime Enterprise deployment. While some add-ons are installed directly on hosts, others are enabled from the **Applications Tab** of the **Create Cluster** or **Edit Cluster** screen.

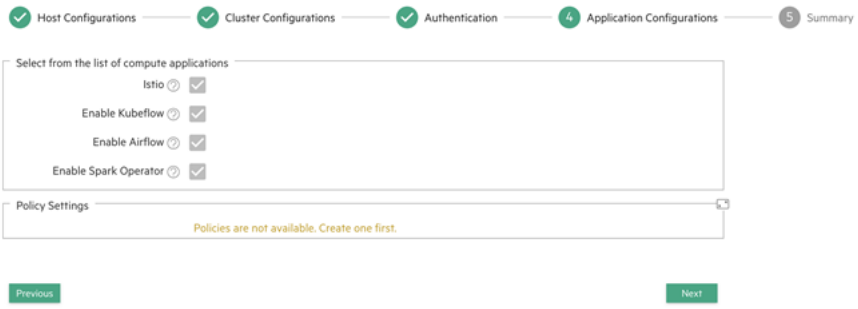
For information about the add-ons that are supported for this release of HPE Ezmeral Runtime Enterprise, see Support Matrixes on page 54.

Not all applications are appropriate for all clusters.

Kubernetes cluster add-ons are installed or enabled in different ways:

- Required add-ons are installed or enabled by default on each Kubernetes cluster created through HPE Ezmeral Runtime Enterprise.

- Some add-ons are installed directly on the hosts, or are enabled through the use of host tags.
- Some add-ons are enabled by selecting the add-on in the **Applications** tab of **Create Cluster** or **Edit Cluster** screen. See Creating a New Kubernetes Cluster and Editing an Existing Kubernetes Cluster.



The following list introduces and links to more information about specific add-ons in HPE Ezmeral Runtime Enterprise:

- Kubernetes cluster applications:
  - Istio Service Mesh on page 492 provides both a transparent open-source service mesh that overlays onto existing distributed applications and a platform that includes APIs for integration with any logging, telemetry, or policy system.
  - Kubeflow on page 503 is a machine learning (ML) toolkit for Kubernetes that makes deployments of ML workflows and pipelines on Kubernetes simple, portable and scalable.
  - Airflow on page 515 is an open-source workflow automation and scheduling system that can be used to author and manage data pipelines.
  - The Spark Operator on page 264 allows you to run Spark Applications in your Kubernetes cluster.
- Falco Container Runtime Security on page 499 provides security and threat detection for hosts and containers.
- NVIDIA GPU Monitoring on page 501 collects GPU metrics such as GPU utilization, GPU memory usage, GPU temperature, and other metrics per GPU device and worker node.

Related tasks

Upgrading Kubernetes Add-Ons on page 900
Use this procedure to upgrade the Kubernetes add-ons and to install new required add-ons on existing Kubernetes clusters in HPE Ezmeral Runtime Enterprise.

Istio Service Mesh

This topic describes Istio Service Mesh and its implementation and versions in HPE Ezmeral Runtime Enterprise.

Shifting to a microservice-based architecture delivers numerous benefits for building distributed fault-tolerant applications. However, this approach also introduces many challenges, such as security, network tracing, and traffic routing that are often left to the application developer to code. This approach can lead to inconsistent and fragmented implementation. A service mesh is designed to solve these problems.

A *service mesh* is a network of microservices that consists of applications and interactions between those applications. Istio provides both a transparent open-source service mesh that overlays onto existing distributed applications and a platform that includes APIs for integration with any logging, telemetry, or policy system.

For a detailed description of Istio features, see What is Istio? (link opens an external website in a new browser tab or window).

To deploy Istio in a Kubernetes cluster in HPE Ezmeral Runtime Enterprise, see Deploying Istio Service Mesh on page 496. You can enable or disable Istio Service Mesh and enable mTLS for each tenant within the cluster.

To access Kiali visualization for Istio Service Mesh, see Accessing Kiali Visualization for Istio Service Mesh on page 498.

Istio Service Mesh is not supported on **HPE Ezmeral Data Fabric on Kubernetes** clusters.

Istio Versions

For information about the versions of Istio that are supported for this release of HPE Ezmeral Runtime Enterprise, see Support Matrixes on page 54.

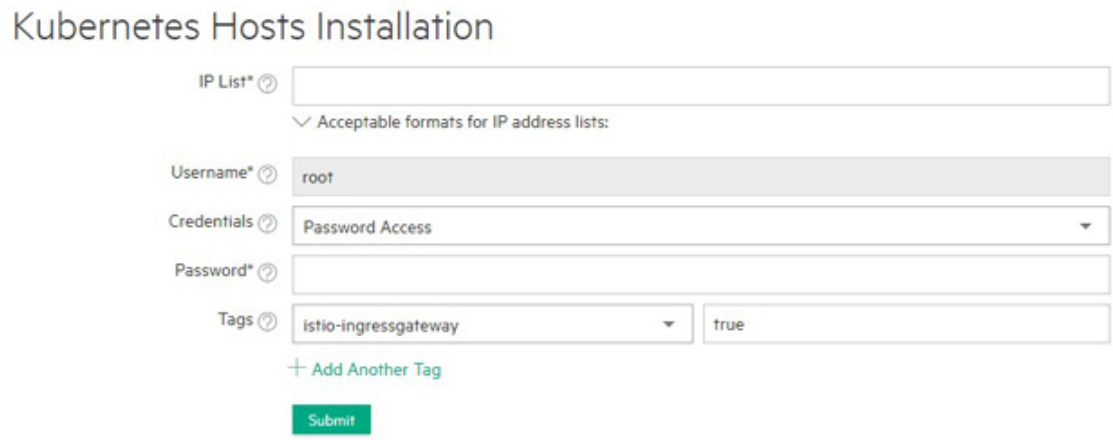
Step One: Add or Assign Istio Ingress Gateway Nodes

All Istio-enabled Kubernetes clusters require one or more Istio Ingress gateways to be configured to allow incoming traffic into the mesh. To add one or more Istio Ingress Gateway nodes, you may either:

- **Add new nodes:** Select the `istio-ingressgateway` tag during Kubernetes Host Step 2: Select the Hosts, and then assign the value `true` to that tag.
- **Assign existing nodes:** Select one or more existing Kubernetes nodes in the **Kubernetes Host Installation** screen (see The Kubernetes Installation Screen), and then assign the `istio-ingressgateway=true` tag, as described in Assigning Tags to a Host.

NOTE: If you are not using the web interface, then `mtls` mode must have a valid value even if Istio is not enabled.

Adding an Istio Ingress Gateway node automatically creates a key value pair for that node, if you added a public SSH key when adding the node. See Kubernetes Host Step 1: Add the Public SSH Key.



Step Two: Create or Edit a Kubernetes Cluster

While creating or editing a Kubernetes cluster, check the **Istio** check box in the **Application Configurations** screen. See Creating a New Kubernetes Cluster and Editing an Existing Kubernetes Cluster.

CAUTION:

Do not select the Istio application when creating or editing an **HPE Ezmeral Data Fabric on Kubernetes** cluster.

Step Three: Enable/Disable Istio Injection

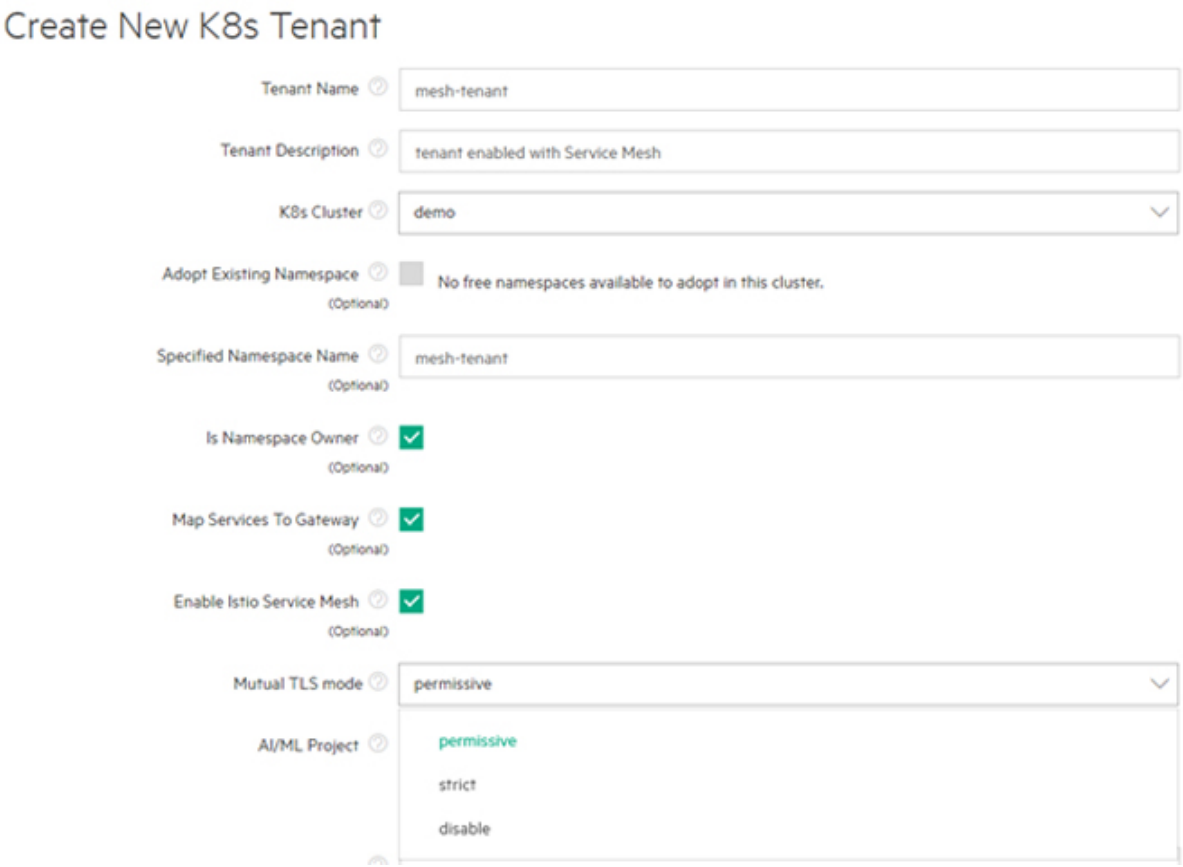
While creating or editing a Kubernetes tenant:

1. Check the **Enable Istio Service Mesh** check box in the **Create New Kuberentes Tenant** or **Edit K8s Tenant** screen. See Creating a New Kubernetes Tenant and Editing an Existing Kubernetes Tenant.

   The **Manual TLS Mode** pull-down menu appears, which allows you to specify the security level to apply to envoy communications.

2. Select one of the following options:
   - **Disable:** Service mesh communication will not be encrypted.
   - **Permissive:** Envoys will accept either plain or TLS-enabled communications. This is the default setting. You can use this setting while creating or migrating workloads and then switch to the **Strict** level later.
   - **Strict:** Envoys only accept TLS-enabled communications.

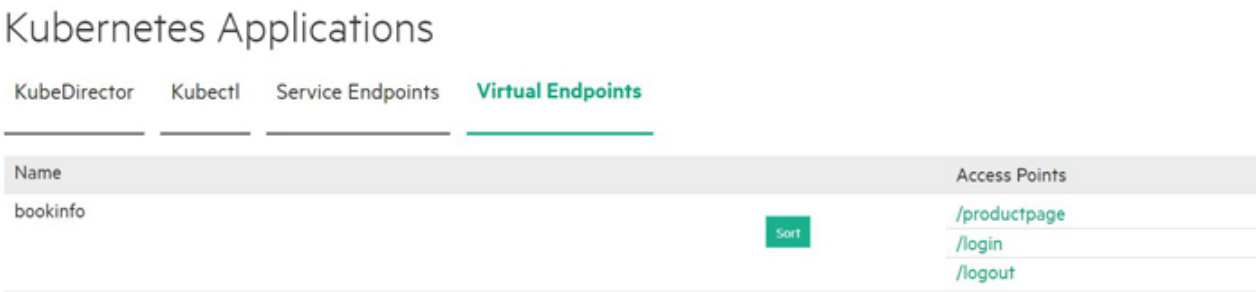
NOTE: Assigning multiple nodes as Istio Ingress Gateways adds load balancing for improved performance in large deployments.



Step 4: Add Applications

After creating the Kubernetes cluster and tenant:

- You may add applications as described in Deploying Applications and Onboarding Applications.
- You can then access Istio virtual services using the **Virtual Services** tab of the **Kubernetes Applications** screen. See Virtual Services Tab.



Visualization Using Kiali

To access Kiali visualization for the Istio service mesh:

1. Open the **Service Endpoints** tab of the **Kubernetes Applications** screen. See Service Endpoints Tab.
2. Click the endpoint you want to add.

   The **Kiali dashboard...** popup appears.

3. Copy the token to your clipboard.
4. Click the Proceed to **Kiali Dashboard** button.

   The **Log in Kiali** screen appears.
5. Paste the token you copied into the Token field, and then click the Log In button.

   The Kiali **Overview** screen appears.

For information about using Kiali, see the Kiali documentation (link opens an external website in a new browser tab/window).

Deploying Istio Service Mesh
This topic describes how to deploy Istio Service Mesh on a Kubernetes cluster in HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Kubernetes Administrator

You have chosen a Kubernetes cluster for which cluster-level installation of Istio is appropriate. For information about the kinds of Kubernetes clusters on which you can use this procedure to deploy Istio Service Mesh, see Istio Service Mesh on page 492.

About this task

You can deploy Istio Service Mesh while creating or editing Kubernetes clusters in HPE Ezmeral Runtime Enterprise. You can also enable or disable Istio Service Mesh and enable mTLS for each tenant within the cluster.

NOTE:

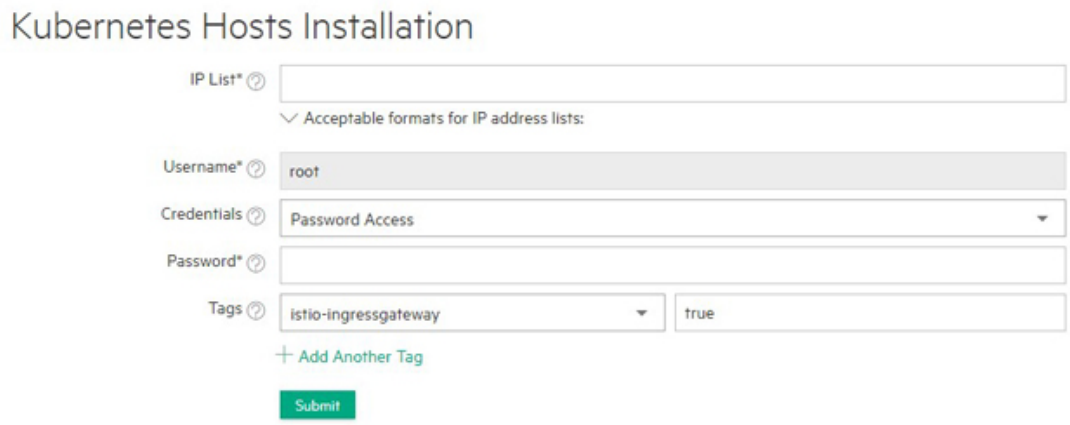
If you are not using the HPE Ezmeral Runtime Enterprise web interface to create or edit the Kubernetes cluster, then `mtls` mode must have a valid value, even if Istio is not enabled.

Procedure

1. Add or assign Istio Ingress gateway nodes.

   To allow incoming traffic into the mesh, all Istio-enabled Kubernetes clusters require one or more Istio Ingress gateways. Assigning multiple nodes as Istio Ingress Gateways adds load balancing for improved performance in large deployments.

   - **Add new nodes:** Select the `istio-ingressgateway` tag during Kubernetes Host Step 2: Select the Hosts, and then assign the value `true` to that tag.
   - **Assign existing nodes:** Select one or more existing Kubernetes nodes in the **Kubernetes Host Installation** screen (see The Kubernetes Installation Screen), and then assign the `istio-ingressgateway=true` tag, as described in Assigning Tags to a Host.



If you added a public SSH key when adding the node, adding an Istio Ingress Gateway node automatically creates a key value pair for that node. See Kubernetes Host Step 1: Add the Public SSH Key.

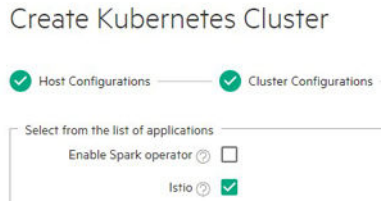
2. Create or edit a Kubernetes cluster, and during the cluster creation or editing process, on the **Application Configurations** screen, select **Istio**.

   IMPORTANT:

   This step deploys "standalone" Istio on the Kubernetes cluster. Not all Kubernetes clusters support the use of standalone Istio. See Istio Service Mesh on page 492.

   For detailed information about creating or editing Kubernetes clusters, see Creating a New Kubernetes Cluster or Editing an Existing Kubernetes Cluster.

   For example:



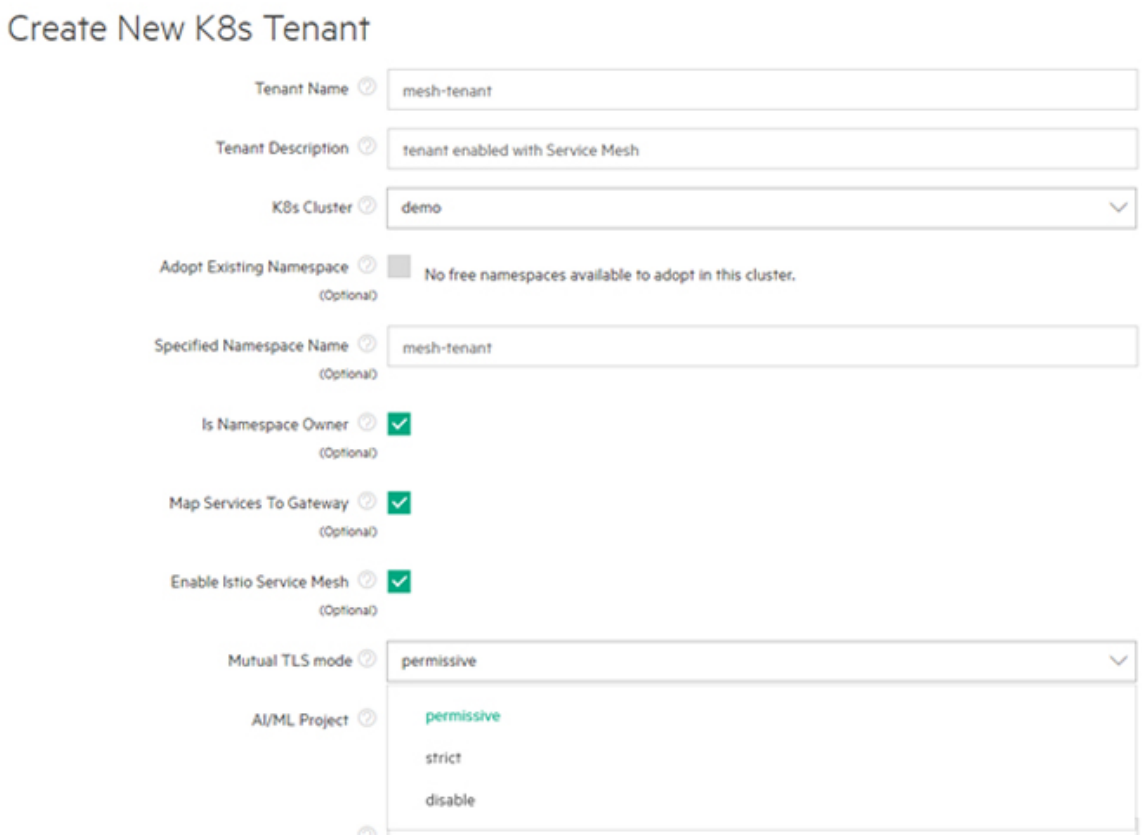
3. When creating or editing a Kubernetes tenant, enable Istio Service Mesh and set the Mutual TLS Mode.

   Mutual TLS Mode specifies the security level to apply to envoy communications.

   For detailed instructions, see one of the following:

   - Creating a New Kubernetes Tenant
   - Editing an Existing Kubernetes Tenant

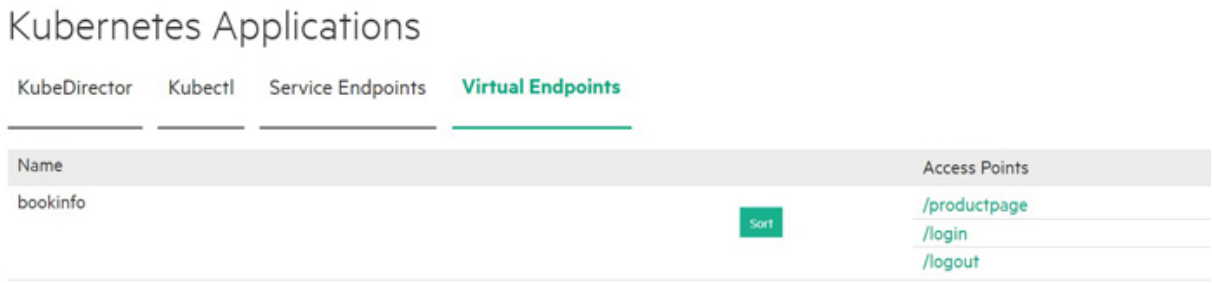
   For example:



4. Add Kubernetes applications as described in Deploying Applications and Onboarding Applications.
5. Access Istio virtual services using the **Virtual Endpoints** tab of the **Kubernetes Applications** screen.

   See The Kubernetes Applications Screen on page 560.

   For example:



Accessing Kiali Visualization for Istio Service Mesh

This topic describes how to access the Kiali visualization services for Istio Service Mesh on HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Kubernetes Administrator

Procedure

1. Open the **Service Endpoints** tab of the **Kubernetes Applications** screen.

2. Click the endpoint you want to add.
3. From the **Kiali dashboard...** dialog, copy the token to your clipboard.
4. Click the **Proceed to Kiali Dashboard** button.

   The **Log in Kiali** screen appears.
5. Paste the token you copied into the **Token** field, and then click the **Log In** button.

   The Kiali **Overview** screen appears.
6. For information about using Kiali, see the Kiali documentation (link opens an external website in a new browser tab or window).

Falco Container Runtime Security
The Falco Container Runtime Security feature of HPE Ezmeral Runtime Enterprise improves container security and threat detection.

Container Runtime Security

Falco container runtime security detects anomalies in the host and in containers by using the extended Berkeley Packet Filter (eBPF) to isolate kernel system calls. The feature is enabled by default and based on the Falco Open Source Software.

For more information about Falco, see the official Falco documentation.

Challenges, Features, and Benefits

Container runtime security is becoming increasingly important in Kubernetes deployments because of some common challenges:

- Vulnerability scanning (Shift Left) is good but not sufficient for many deployments.
- Never-ending CVE exploits and malicious intrusions are a fact of life.
- Microservices present a wider attack surface.
- Threats you do not see are impossible to deter.

Container runtime security features provide:

- Runtime security by Falco that is enabled by default.
- Git-integrated automatic synchronization of new rules.

These features combine to provide the following benefits:

- Improved container security and threat detection.
- Reduced risk with immediate alerting.
- An early-warning system that leverages the most current detection rules for CVEs and malicious exploits.

Deploying Falco on an HPE Ezmeral Runtime Enterprise Kubernetes Cluster

Installing the Falco Kernel Module is optional.

If you choose to install Falco Kernel Modules see the following requirements and recommendations:

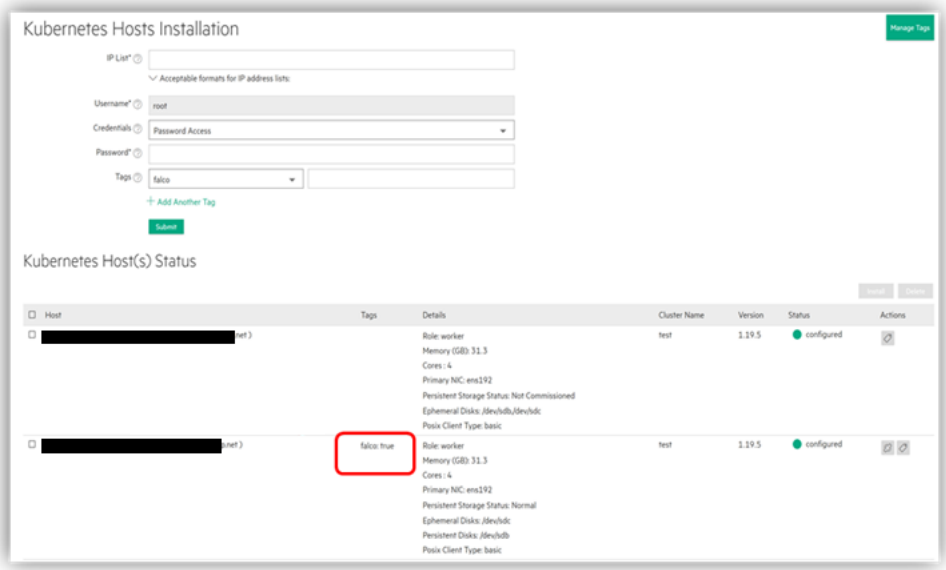
- The Falco Kernel Module you install on the host must be the correct version for the host OS and OS version, must support Falco Kernel Driver API Schema version 2.1.0, and must be installed using the Linux `modprobe` tool. See the official Falco documentation.

  When you use `modprobe` tool to install the module, the `modinfo` tool can collect information about the module. For information about `modprobe`, see the `modprobe(8)` manpage.

- Hewlett Packard Enterprise recommends that you install the modules on all the hosts in a Kubernetes cluster.

- Hewlett Packard Enterprise recommends that you install the Falco Kernel module after you install the OS on the host, but before you install the HPE Ezmeral Runtime Enterprise software on that host.

- If you install the Falco Kernel Modules on hosts before you add the hosts to HPE Ezmeral Runtime Enterprise, then HPE Ezmeral Runtime Enterprise automatically tags the hosts as



  `falco:true`:

- If you are installing the Falco Kernel Module on the hosts that are already in Kubernetes cluster, after you install the Falco Kernel Module on all the hosts in the cluster, you must manually add the `falco:true` tag to each Kubernetes node in the cluster.

Enabling the Falcosidekick UI

The Falcosidekick UI enables you to view the latest events from Falco in real time through your web browser.

For information on the Falcosidekick UI, see this page from the Falco GitHub (link opens an external site in a new browser tab or window).

If you are using Kubernetes version 1.22 or higher, proceed as follows:

1. To deploy the Falcosidekick UI, you must set up a storage class (PV/PVC) for your Kubernetes cluster.
   - If your Kubernetes cluster is configured with Data Fabric or has a storage class marked as `default`, the Falcosidekick UI uses the default storage class.
   - If your Kubernetes cluster does not have a default storage class defined, then HPE Ezmeral Runtime Enterprise is unable to deploy the Falcosidekick UI.

2. After setting up a storage class on your Kubernetes cluster, contact Hewlett Packard Enterprise support for assistance to manually deploy the Falcosidekick UI.

   NOTE: Beginning with Kubernetes version 1.22, you must use Falcosidekick UI version 2.2.5 or greater.

If you are using a version of Kubernetes lower than 1.22, proceed as follows:

1. To enable the Falcosidekick UI for a Kubernetes cluster, enter the following command:

```
kubectl patch svc falco-falcosidekick-ui --type='json' -p
'[{"op":"replace","path":"/spec/type","value":"NodePort"}]' -n
hpecp-falco
```

2. The annotation values that return tell you the gateway port you can use to access the Falcosidekick UI.

   For example:

```
!#to get gateway port
kubectl describe svc falco-falcosidekick-ui -n hpecp-falco
Name: falco-falcosidekick-ui
Namespace: hpecp-falco
...
Annotations: <example-gateway>/2802: m2-ess-vm77.<example.net>:10035
meta.helm.sh/release-name: falco
...
Type: NodePort
...
!#to access Falco UI:
http://<example.net>:10035/ui
```

UEFI Secure Boot Limitation

If the Kubernetes node has UEFI Secure Boot enabled, any Falco-related functionality associated with the node will not be operational. For more information about UEFI Secure Boot, see What Is UEFI Secure Boot.

NVIDIA GPU Monitoring

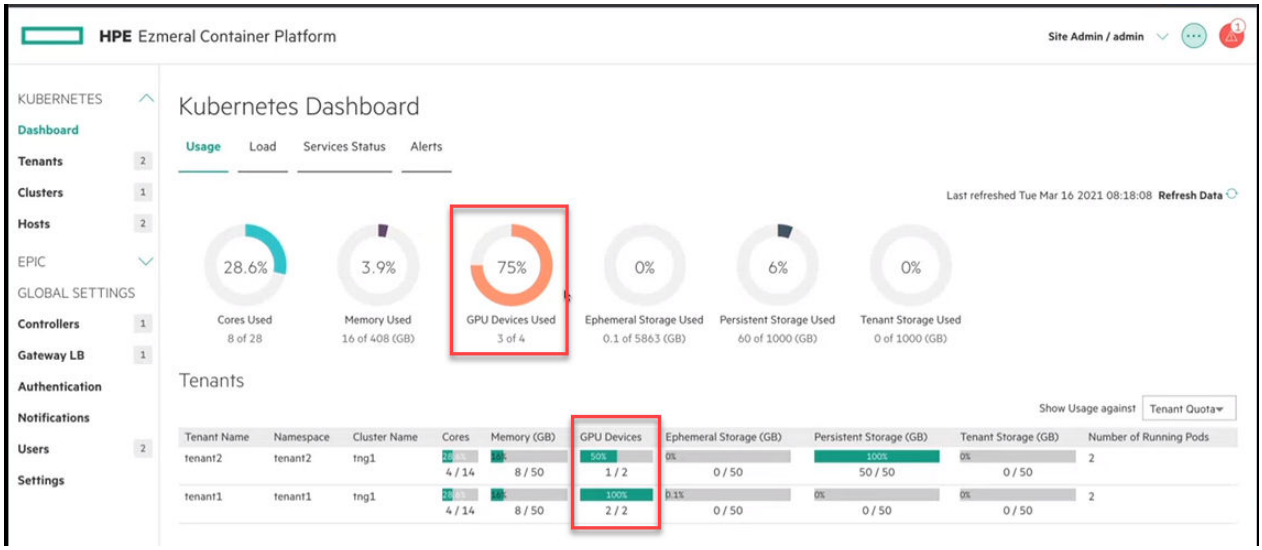
HPE Ezmeral Runtime Enterprise includes an hpecp-nvidiagpubeat add-on that is deployed by default on non-imported Kubernetes clusters. The hpecp-nvidiagpubeat add-on deploys the nvidiagpubeat DaemonSet, which deploys an nvidiagpubeat collector pod on each worker node with one or more NVIDIA GPUs. The collector pod collects GPU metrics such as GPU utilization, GPU memory usage, GPU temperature, and other metrics per GPU device and worker node.

For more information about nvidiagpubeat, see nvidiagpubeat.

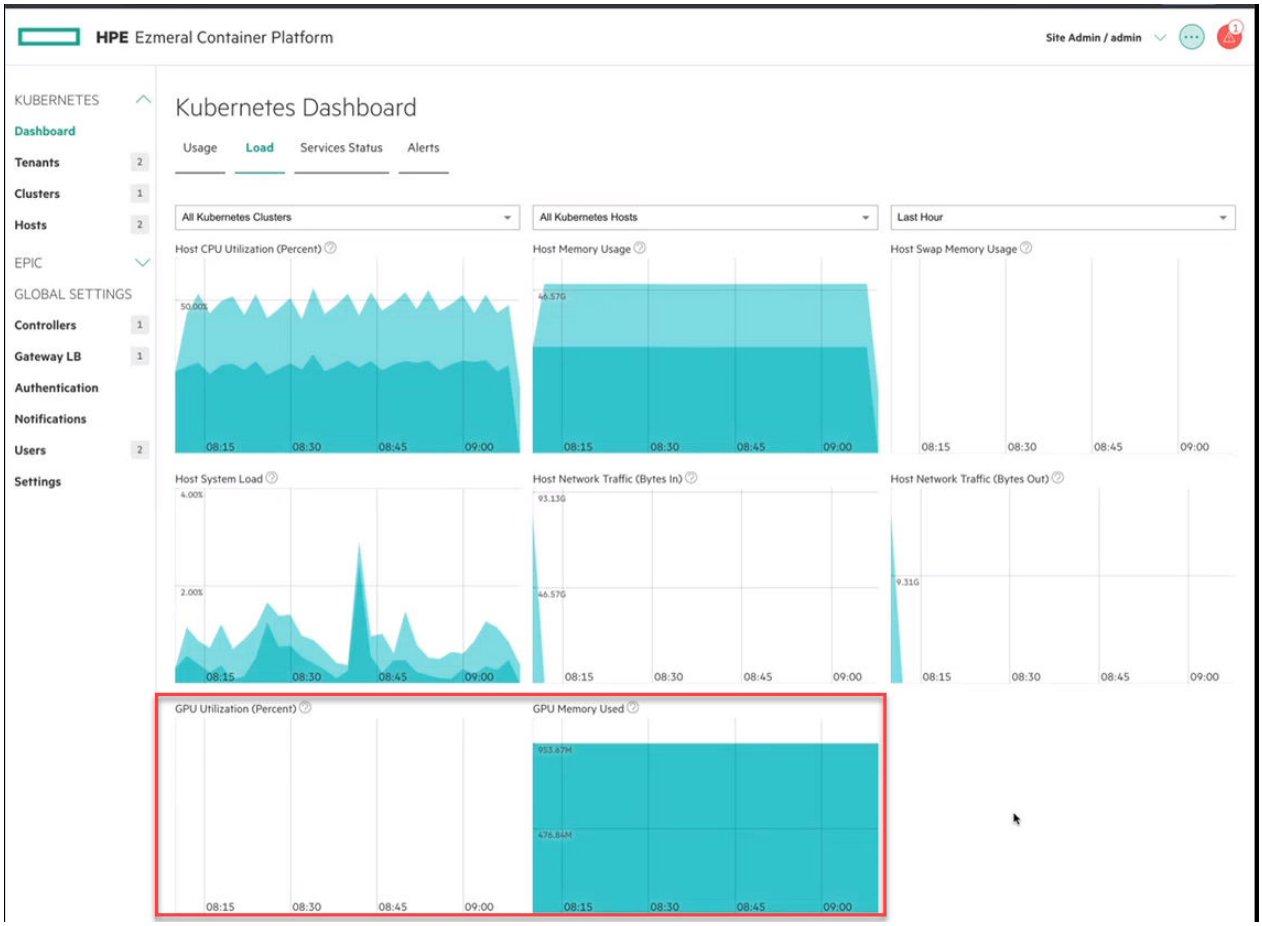
GPU Charts and Statistics

HPE Ezmeral Runtime Enterprise displays GPU metrics on the **Usage** tab of the **Kubernetes Dashboard**. The **Usage** tab shows allocated GPUs vs. total available or GPU quota per tenant.

For cluster administrators and Platform Administrators, the **Dashboard** > **Usage** tab shows the GPU devices used system wide. The tenant table shows the GPU devices in use per tenant:



The **Dashboard** > **Load** tab shows new graphs for GPU utilization and GPU memory used:



nvidiagpubeat Add-On Installation

The hpecp-nvidiagpubeat add-on is a required system add-on and is deployed by default on Kubernetes clusters.

On each host that contains GPUs, you must install an OS-compatible GPU driver that supports your GPU model. You must install the driver **before** adding the GPU host to HPE Ezmeral Runtime Enterprise. For installation instructions, see GPU Driver Installation on page 838.

The number of GPU hosts that you add determines the number of collector pods that are created and deployed on the cluster. For example, if your Kubernetes cluster contains one master node (non-GPU machine) and one worker node (GPU machine), one nvidiagpubeat pod is deployed.

nvidiagpubeat and Imported Clusters

The hpecp-nvidagpubeat add-on is not supported for imported clusters.

Logs for the nvidiagpubeat Pods

To check the metrics logs for nvidiagpubeat pods, execute this command:

```
kubectl -n kube-system logs <nvidiagpubeat-pod-name>
```

Alternatively, you can download the logs to a file:

```
kubectl -n kube-system logs <nvidiagpubeat-pod-name> >
<nvidiagpubeat-pod-name>.log
```

Kubeflow

Kubeflow is a machine learning (ML) toolkit for Kubernetes that makes deployments of ML workflows and pipelines on Kubernetes simple, portable and scalable.

Kubeflow is a machine learning (ML) toolkit for Kubernetes. Kubeflow makes deployments of ML workflows on Kubernetes simple, portable and scalable. Kubeflow is for operational teams who want to deploy ML pipelines to different environments for development, testing, and production use.

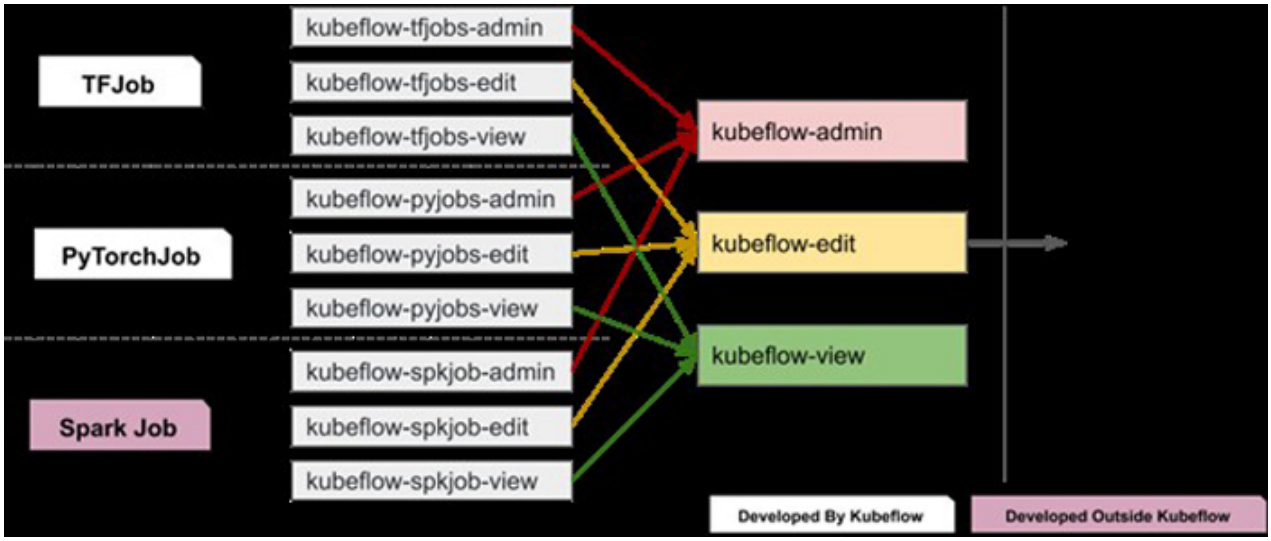
To learn more about Kubeflow, see the official Kubeflow documentation.

NOTE: Beginning with HPE Ezmeral Runtime Enterprise 5.5.1, Kubeflow notebooks are available. However, Hewlett Packard Enterprise recommends that you use full-featured KubeDirector notebooks instead. See Creating Notebook Servers on page 169.

Before Installing Kubeflow

Kubeflow automatically assigns the following Kubeflow-specific aggregating cluster roles:

- `kubeflow-admin`
- `kubeflow-edit`
- `kubeflow-view`



See the following resources for more information on how Kubeflow implements RBAC. The following links open external websites in a new browser tab or window:

- Kubeflow Multi-user Isolation
- Design document
- Community guidelines
- Blog post

System Requirements

For supported Kubernetes versions, see Kubernetes Version Requirements on page 832. For issues and workarounds, see Issues and Workarounds on page 15.

The following resources must be available to install Kubeflow:

- Minimum number of nodes for compute cluster: 2 (1 master, 1 worker)
- Minimum core and memory resources required:
  - CPU Cores: 36
  - Memory (GB): 160

Dynamic Volume Provisioning for Persistent Volumes

Persistent Volumes for Kubeflow are dynamically provisioned using a default Storage Class.

- If you are using HPE Ezmeral Data Fabric, a default Storage Class is created and marked as `default`. For example:

```
kubectl get sc

NAME                                PROVISIONER
RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
cluster-1664736275961 (default)   com.mapr.csi-kdf
Delete          Immediate             true                   19h
cluster-1664736275961-nfs         com.mapr.csi-nfskdf
Delete          Immediate             true                   19h
hpe-hdd-storage                   kubernetes.io/no-provisioner
Delete          WaitForFirstConsumer   false                 19h
hpe-nvme-storage                  kubernetes.io/no-provisioner
Delete          WaitForFirstConsumer   false                 19h
hpe-ssd-storage                   kubernetes.io/no-provisioner
Delete          WaitForFirstConsumer   false                 19h
```

- If you are using a storage configuration **other than** HPE Ezmeral Data Fabric on Kubernetes, mark a Storage Class as `default` prior to Kubeflow installation.

  Use the following command to mark a Storage Class as `default`:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

  For information on available storage configurations, see Storage on page 804.

- MinIO can be configured to provide Kubeflow persistent volumes. For information, see Configuring External MinIO on page 505.

Kubeflow Components

See Support Matrixes on page 54.

Kubeflow Installation

Installing Kubeflow on a New Kubernetes Cluster

Deploy Kubeflow on a Kubernetes cluster by specifying the **Enable Kubeflow** check box and the **Istio** check box on the **Application Configurations** tab during cluster creation.

See Creating a New Kubernetes Cluster on page 463.

Installing Kubeflow on an Existing Kubernetes Cluster

Deploy Kubeflow after a Kubernetes cluster is created by editing the cluster and selecting **Enable Kubeflow** and the **Istio** check box on the **Application Configurations** tab. Save your changes and wait until the cluster is ready and Kubeflow services are up and running.

NOTE: Istio is required for Kubeflow installation. If Istio is not installed, Kubeflow will not work.

NOTE: Kubeflow can use any external Istio with the same version as the platform Istio add-on (see Support Matrixes on page 54). However, Hewlett Packard Enterprise recommends installing the platform Istio add-on instead.

Authentication Setup

Kubeflow supports both Active Directory and LDAP platform authentication.

If you have not configured Active Directory or LDAP authentication, Kubeflow can still be deployed. In this case, the admin username is set to `admin@kubeflow.org` and the password is set to `12341234` by default.

Configuring External MinIO

This article describes how to configure external MinIO for Kubeflow. You can perform this action before or after deploying Kubeflow on your Kubernetes cluster.

Before Deploying Kubeflow

To configure external MinIO before deploying Kubeflow on your Kubernetes cluster:

1. Connect through `ssh` to the Kubernetes master node.
2. Make sure the `kubeflow-minio` namespace exists. If it does not exist, create it now by executing the following in the shell:

```
if ! kubectl get ns kubeflow-minio > /dev/null 2> /dev/null; then
 kubectl create ns kubeflow-minio
fi
```

3. Create a secret with the external MinIO config:

- If minio is deployed in the `kubeflow` namespace, execute the following command:

  ```
  kubectl -n kubeflow-minio create secret
  generic  kubeflow-external-minio --from-literal='host=<minio
  service address>' --from-literal='port=<minio
  service port>' --from-literal='insecure=<true/
  false>' --from-literal='accesskey=<minio access
  key>' --from-literal='secretkey=<minio secret key>' -n kubeflow-minio
  ```

  For example:

  ```
  kubectl -n kubeflow-minio create secret generic
  kubeflow-external-minio --from-literal='host=minio-service.minio.svc.cl
  uster.local' --from-literal='port=9000' --from-literal='secure=false'
   --from-literal='accesskey=minio' --from-literal='secretkey=minio123'
  ```

- If minio is not deployed in the `kubeflow` namespace, you must provide an additional option `namespace` to `kubeflow-external-minio-secret`. The default value is set to `kubeflow`.

  Execute the following command:

  ```
  kubectl create secret generic
  kubeflow-external-minio --from-literal='namespace=<minio
  namespace>' --from-literal='host=<minio
  service address>' --from-literal='port=<minio
  service port>' --from-literal='secure=<true/
  false>' --from-literal='accesskey=<minio access
  key>' --from-literal='secretkey=<minio secret key>' -n kubeflow-minio
  ```

4. Transfer the artifacts from the Kubeflow MinIO to your external S3 storage. Find the necessary instructions and scripts here (link opens an external website in a new browser tab or window).

5. Deploy Kubeflow, as described in Kubeflow Installation on page 505.

After Deploying Kubeflow

To configure external MinIO after deploying Kubeflow on your Kubernetes cluster:

1. Edit the secret `mlpipeline-minio-artifact` in the namespace `kubeflow`:

   ```
   kubectl edit secret mlpipeline-minio-artifact -n kubeflow
   ```

   Edit the secret `mlpipeline-minio-artifact` in all other profile (user) namespaces:

   ```
   kubectl edit secret mlpipeline-minio-artifact -n <profile_namespace>
   ```

   Change the data in the secret so that it matches your MinIO configuration. All values must be encoded in `base64` format.

   For `base64` encoding, you can use this tool.

2. Transfer the artifacts from the Kubeflow MinIO to your external S3 storage. Find the necessary instructions and scripts here (link opens an external website in a new browser tab or window).

3. Restart the `ml-pipeline` and `workflow-controller` deployments:

```
kubectl rollout restart deploy -n kubeflow ml-pipeline
```

```
kubectl rollout restart deploy -n kubeflow workflow-controller
```

4. To apply changes, restart the `ml-pipeline-ui-artifact` pod in each profile namespace:

```
kubectl delete ml-pipeline-ui-artifact-xxx -n <profile_namespace>
```

Kubeflow in an Air-Gapped Environment

If Air Gap is configured on the platform, push all of the required images to the registry before installing Kubeflow.

See List of Kubeflow Images to be installed.

Validating the Kubeflow Installation

Prerequisites

Required access rights: Platform Administrator or Cluster Administrator

About this task

When your Kubernetes cluster is ready, perform the following basic tests to validate the Kubeflow installation.

The Kubeflow dashboard link in the AI/ML tenant UI will be active after the Kubeflow service is up, which may take 10-15 minutes.

Procedure

1. Run the following command on the master node to ensure that the kubeflow-installer pod has been completed.

```
kubectl get pods -n kubeflow-jobs

NAME                            READY   STATUS      RESTARTS   AGE
dex-secret-generator-wq4l4      0/1     Completed   0          17m
hpecpconfig-patch-qz2gd         0/1     Completed   0          14m
minio-config-generator-4gksg    0/1     Completed   0          17m
proxy-cm-generator-jmb8q        0/1     Completed   0          17m
```

2. Confirm that the Dex config secret was created according to the authentication settings provided during cluster creation.

```
kubectl get secret dex-config-secret -n auth -o yaml
```

3. Confirm that the pods are up and running. Deployment of Kubeflow manifests may take 10-15 minutes.

```
kubectl get pods -n auth && kubectl get pods -n cert-manager
&& kubectl get pods -n knative-eventing && kubectl get pods -n
knative-serving && kubectl get pods -n kubeflow && kubectl get pods -n
kubeflow-user-example-com && kubectl get pods -n prism-ns
```

The output should appear as follows:

```
NAME                    READY   STATUS    RESTARTS   AGE
dex-798fd4d8f9-tzd88    1/1     Running   0          32m
NAME                                        READY   STATUS    RESTARTS
AGE
cert-manager-67b5569945-clt67               1/1     Running   0
33m
cert-manager-cainjector-7dbb46f46d-zvtv6    1/1     Running   0
33m
cert-manager-webhook-7c48978bcc-ctqv4       1/1     Running   0
33m
NAME                                     READY   STATUS    RESTARTS   AGE
eventing-controller-5f4c7bbf4b-qjmjp     1/1     Running   0          32m
eventing-webhook-585df69b96-ztpz4        1/1     Running   0          32m
NAME                                       READY   STATUS    RESTARTS
AGE
activator-859bc95758-6cj6x                 2/2     Running   0
32m
autoscaler-67f94897c8-n75gt                2/2     Running   0
32m
controller-c5959bc48-qpp6g                 1/1     Running   0
32m
domain-mapping-bfdd97c95-bvq6l             2/2     Running   0
32m
domainmapping-webhook-65cf4d6986-m2sqw     2/2     Running   1 (32m ago)
32m
net-istio-controller-6cfbd68cbf-fsfkc      2/2     Running   1 (32m ago)
32m
net-istio-webhook-677cfd865b-s78mx         2/2     Running   1 (32m ago)
32m
webhook-87b94cb45-xhfwj                    2/2     Running   1 (32m ago)
32m
NAME                                                        READY
STATUS    RESTARTS      AGE
admission-webhook-deployment-7978f87497-t4x4v               1/1
Running   0             31m
cache-deployer-deployment-7f7d7f757f-rfjqp                  2/2
Running   1 (31m ago)   32m
cache-server-977bdbdd-w4jdr                                 2/2
Running   0             32m
centraldashboard-84695f67cf-fdngl                           2/2
Running   0             31m
jupyter-web-app-deployment-597d47b664-ggxft                 1/1
Running   0             31m
katib-controller-6478fbd64c-wbxxr                           1/1
Running   0             31m
katib-db-manager-78fc8b7895-sfbc8                           1/1
Running   0             31m
katib-mysql-6975d6c6c4-xvz4q                                1/1
Running   0             31m
katib-ui-5cb6cc4d97-k4dct                                   1/1
Running   0             31m
kserve-controller-manager-0                                 2/2
Running   0             31m
```

```
kserve-models-web-app-75f5c6cc9f-dx5xl                    2/2
Running   0             31m
kubeflow-pipelines-profile-controller-858fdbf777-znb6l   1/1
Running   0             32m
metacontroller-0                                          1/1
Running   0             32m
metadata-envoy-deployment-f4c868c97-zng4p                 1/1
Running   0             32m
metadata-grpc-deployment-679b49cc95-wtj9d                 2/2
Running   2 (31m ago)   32m
metadata-writer-7459bcd96b-prp86                          2/2
Running   0             32m
minio-console-6bc546d664-mhlf9                            1/1
Running   0             31m
minio-dcb5fcb5c-rwzqq                                     2/2
Running   0             31m
ml-pipeline-777989d7f8-c6kb8                              2/2
Running   1 (30m ago)   32m
ml-pipeline-persistenceagent-848b7bbc88-pgznx             2/2
Running   0             32m
ml-pipeline-scheduledworkflow-546fc65b4c-gz6gk            2/2
Running   0             32m
ml-pipeline-ui-8847c787b-6gctl                            2/2
Running   0             32m
ml-pipeline-viewer-crd-5c79ccf5b6-r7mzk                   2/2
Running   1 (31m ago)   32m
ml-pipeline-visualizationserver-8666b88867-lb49z          2/2
Running   0             32m
mysql-76b487989d-ckkxg                                    2/2
Running   0             32m
notebook-controller-deployment-568544dfcf-f2s9k           1/1
Running   0             31m
profiles-deployment-77489847c9-bntg9                      3/3
Running   2 (29m ago)   31m
seldon-controller-manager-77c74849b6-jgpxt                1/1
Running   0             30m
tensorboards-web-app-deployment-65b8646ff6-m8cld          1/1
Running   0             31m
training-operator-866bcdf8fb-7n4tg                        1/1
Running   0             31m
volumes-web-app-deployment-c49cd595f-wmgcx                1/1
Running   0             31m
workflow-controller-566b84599c-krv9p                      2/2
Running   1 (31m ago)   32m
NAME                                                 READY   STATUS
RESTARTS   AGE
ml-pipeline-ui-artifact-68bc7b65b5-r48kw             2/2     Running
0          28m
ml-pipeline-visualizationserver-5c867dbddd-lfftq     2/2     Running
0          28m
NAME                            READY   STATUS    RESTARTS   AGE
kftoken-app-65f794d6ff-fhrvj    1/1     Running   0          30m
prism-647f69f9bb-drdgw          1/1     Running   0          30m
```

Kubeflow Post-Installation Steps

(Recommended) Configure KFServing

KFServing is based on top of Knative. By default, Knative enables a scale-to-zero feature. For testing or development purposes, it can be useful to know how to disable this feature.

1. To disable scale-to-zero, enter the following command:

   ```
   kubectl -n knative-serving edit cm config-autoscaler
   ```

   Add the following to the bottom of the file:

   ```
   data:
     enable-scale-to-zero: "false"
   ```

2. If your environment is behind a proxy, add the appropriate environmental variables to controller deployment in the `knative-serving` namespace. From the Kubernetes master node, execute the following commands:

   ```
   kubectl set env deployment/controller -n knative-serving
   http_proxy="$http_proxy"
   kubectl set env deployment/controller -n knative-serving
   https_proxy="$https_proxy"
   kubectl set env deployment/controller -n knative-serving
   no_proxy="$no_proxy"
   ```

Upgrading Kubeflow Clusters

This topic describes the steps to upgrade Kubernetes clusters with the Kubeflow add-on to a newer Kubernetes version. You can also migrate legacy Kubernetes clusters with the Kubeflow add-on from Docker container runtime to the the new Hewlett Packard Enterprise distribution of Kubernetes, which implements containerd runtime.

Upgrading Kubeflow Clusters in HPE Ezmeral Runtime Enterprise

- If you are upgrading from HPE Ezmeral Runtime Enterprise version 5.4.2 or above, HPE Ezmeral Runtime Enterprise automatically upgrades the Kubeflow add-on during the platform upgrade.

  To upgrade the Kubeflow cluster to a newer version of Kubernetes:

  1. Back up any sensitive user data, such as user notebooks and models trained by users.
  2. Upgrade the cluster to a newer version of Kubernetes as described in Upgrading Kubernetes on page 487.
  3. Restore the user data from backup.

- If you are upgrading from a HPE Ezmeral Runtime Enterprise version below 5.4.2, see Upgrading Kubeflow Clusters in HPE Ezmeral Runtime Enterprise on page 510.

Migrating Legacy Kubeflow Clusters in HPE Ezmeral Runtime Enterprise

Legacy Kubernetes clusters with an existing Kubeflow add-on remain on Docker container runtime until you manually migrate them to the the new Hewlett Packard Enterprise distribution of Kubernetes. For information on legacy clusters, see Kubernetes Cluster Types and Compatibility on page 322.

To migrate existing legacy Kubeflow clusters to the new Hewlett Packard Enterprise distribution of Kubernetes, perform the following:

1. Back up any sensitive user data, such as user notebooks and models trained by users.
2. If you are migrating from a HPE Ezmeral Runtime Enterprise version prior to 5.4.2, remove the Kubeflow add-on. See Uninstalling Kubeflow on page 515.
3. Migrate the Kubeflow cluster to the new Hewlett Packard Enterprise distribution of Kubernetes, as described in Migrating Kubernetes Clusters from Docker to containerd on page 323.

4. If you are performing the migration from a HPE Ezmeral Runtime Enterprise version prior to 5.4.2, re-enable the Kubeflow add-on in the UI after the migration is complete. See Kubeflow Installation on page 505.

5. Restore the user data from backup.

Multiuser Support in AI/ML Tenants

Kubeflow supports multiuser isolation, which applies access control over namespaces and user-created resources in a deployment.

When you create an AI/ML tenant, a new Kubeflow profile is created automatically.

When a user signs into the KD Notebook, Kubeflow automatically adds the user to the profile as a contributor.

NOTE: The current version of Kubeflow does not support groups.

Adding Contributors with the Users Secret

Add contributors to an AI/ML tenant-based Kubeflow profile.

Prerequisites

- **Required access rights:** Kuberentes Administrator or Tenant Administrator

About this task

NOTE: Kubeflow automatically generates a secret in the tenant namespace for users who are signed in to any KD Jupyter notebook in the ML Ops Tenant. For users who are not signed in, follow this procedure to generate a secret manually.

Contributors added to the AI/ML tenant-based Kubeflow profile gain access to the tenant namespace. In the **Kubeflow Dashboard**, the tenant namespace becomes available in the dropdown list of available namespaces.

Procedure

Create a users secret in the tenant namespace:

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: kubeflow-users-secret
  namespace: <tenant_namespace>
data:
  users: <base64_encoded_usernames>
```

For encoding purposes, use a comma to separate usernames. For example:

```
users: ZGV2MSxxYTEK # - dev1,qa1
```

After you create the users secret, Kubeflow creates a role binding for each user.

You can check the rolebindings to validate that a contributor was added successfully. For example:

```
kubectl get rolebindings -n <tenant_namespace> | grep <user_name>

 user-<user_name>-clusterrole-edit-cf4b9chf5t ClusterRole/kubeflow-edit 17s
```

List of Kubeflow Images

This article lists the images required for Kubeflow and the tutorials.

The images for HPE Ezmeral Runtime Enterprise are:

```
alpine:3.10
alpine:latest
bitnami/kubectl:1.18
bluedata/
kubeflow-retagged-for-airgap:knative.dev-eventing-cmd-apiserver_receive_adap
ter-v1.2.4
bluedata/pytorch:mnist-ddp-cpu
busybox:1.28
busybox:latest
curlimages/curl:latest
docker.io/horovod/horovod:0.20.0-tf2.3.0-torch1.6.0-mxnet1.5.0-py3.7-cpu
docker.io/istio/pilot:1.12.0
docker.io/istio/pilot:1.13.5
docker.io/istio/pilot:1.9.0
docker.io/istio/pilot:1.9.6
docker.io/istio/proxyv2:1.12.0
docker.io/istio/proxyv2:1.13.2
docker.io/istio/proxyv2:1.13.5
docker.io/istio/proxyv2:1.9.0
docker.io/istio/proxyv2:1.9.6
docker.io/kfserving/kfserving-controller:v0.6.1
docker.io/kfserving/pytorchserver:v0.6.1
docker.io/kfserving/pytorchserver:v0.6.1-gpu
docker.io/kfserving/sklearnserver:v0.6.1
docker.io/kubeflowkatib/cert-generator:v0.14.0
docker.io/kubeflowkatib/earlystopping-medianstop:v0.14.0
docker.io/kubeflowkatib/enas-cnn-cifar10-cpu:v0.14.0
docker.io/kubeflowkatib/file-metrics-collector:v0.14.0
docker.io/kubeflowkatib/katib-controller:v0.14.0
docker.io/kubeflowkatib/katib-db-manager:v0.14.0
docker.io/kubeflowkatib/katib-ui:v0.14.0
docker.io/kubeflowkatib/mxnet-mnist:v0.14.0
docker.io/kubeflowkatib/pytorch-mnist-cpu:v0.14.0
docker.io/kubeflowkatib/pytorch-mnist:v0.13.0
docker.io/kubeflowkatib/suggestion-chocolate:v0.14.0
docker.io/kubeflowkatib/suggestion-darts:v0.14.0
docker.io/kubeflowkatib/suggestion-enas:v0.14.0
docker.io/kubeflowkatib/suggestion-goptuna:v0.14.0
docker.io/kubeflowkatib/suggestion-hyperband:v0.14.0
docker.io/kubeflowkatib/suggestion-hyperopt:v0.14.0
docker.io/kubeflowkatib/suggestion-optuna:v0.14.0
docker.io/kubeflowkatib/suggestion-pbt:v0.14.0
docker.io/kubeflowkatib/suggestion-skopt:v0.14.0
docker.io/kubeflowkatib/tf-mnist-with-summaries:latest
docker.io/kubeflowkatib/tfevent-metrics-collector:v0.14.0
docker.io/kubeflownotebookswg/jupyter-web-app:v1.6.0
docker.io/kubeflownotebookswg/kfam:v1.6.0
docker.io/kubeflownotebookswg/poddefaults-webhook:v1.6.0
docker.io/kubeflownotebookswg/tensorboards-web-app:v1.6.0
docker.io/kubeflownotebookswg/volumes-web-app:v1.6.0
docker.io/metacontrollerio/metacontroller:v2.0.4
docker.io/seldonio/engine:1.12.0
docker.io/seldonio/mlserver:0.5.3
docker.io/seldonio/seldon-core-executor:1.12.0
docker.io/seldonio/seldon-core-operator:1.12.0
gcr.io/arrikto/kubeflow/oidc-authservice:2cb5bf6
gcr.io/arrikto/kubeflow/oidc-authservice:6ac9400
gcr.io/google-containers/busybox:latest
```

```
gcr.io/knative-releases/knative.dev/eventing/cmd/broker/filter:v1.2.4
gcr.io/knative-releases/knative.dev/eventing/cmd/broker/ingress:v1.2.4
gcr.io/knative-releases/knative.dev/eventing/cmd/controller:v1.2.4
gcr.io/knative-releases/knative.dev/eventing/cmd/mtping:v1.2.4
gcr.io/knative-releases/knative.dev/eventing/cmd/webhook:v1.2.4
gcr.io/knative-releases/knative.dev/net-istio/cmd/controller:v1.2.0
gcr.io/knative-releases/knative.dev/net-istio/cmd/webhook:v1.2.0
gcr.io/knative-releases/knative.dev/serving/cmd/activator:v1.2.5
gcr.io/knative-releases/knative.dev/serving/cmd/autoscaler:v1.2.5
gcr.io/knative-releases/knative.dev/serving/cmd/controller:v1.2.5
gcr.io/knative-releases/knative.dev/serving/cmd/
domain-mapping-webhook:v1.2.5
gcr.io/knative-releases/knative.dev/serving/cmd/domain-mapping:v1.2.5
gcr.io/knative-releases/knative.dev/serving/cmd/queue:v1.2.5
gcr.io/knative-releases/knative.dev/serving/cmd/webhook:v1.2.5
gcr.io/kubebuilder/kube-rbac-proxy:v0.8.0
gcr.io/kubeflow-ci/tf-mnist-with-summaries:1.0
gcr.io/kubeflow-images-public/kubebench/workflow-agent:bc682c1
gcr.io/mapr-252711/kubeflow/argo/init-container:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/central-dashboard:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/installer:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/minio/console:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/minio/init:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/minio/server:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/minio/wait-script:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/ml-pipeline-api-server:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/ml-pipeline-cache-server:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/notebook-controller:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/prism/kftokenpod:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/prism/prism-operator:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/profile-controller:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/python-k8s:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/seldon/mlflowserver:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/tensorboard-controller:ecp-5.5.0-release
gcr.io/mapr-252711/kubeflow/thirdparty/knative/eventing/
channel_broker:v1.2.4
gcr.io/mapr-252711/kubeflow/thirdparty/knative/eventing/
channel_controller:v1.2.4
gcr.io/mapr-252711/kubeflow/thirdparty/knative/eventing/
channel_dispatcher:v1.2.4
gcr.io/ml-pipeline/argoexec:v3.1.6-patch-license-compliance
gcr.io/ml-pipeline/argoexec:v3.3.8-license-compliance
gcr.io/ml-pipeline/cache-deployer:2.0.0-alpha.3
gcr.io/ml-pipeline/frontend:2.0.0-alpha.3
gcr.io/ml-pipeline/kfp-launcher:1.7.1
gcr.io/ml-pipeline/metadata-envoy:2.0.0-alpha.3
gcr.io/ml-pipeline/metadata-writer:2.0.0-alpha.3
gcr.io/ml-pipeline/minio:RELEASE.2019-08-14T20-37-41Z-license-compliance
gcr.io/ml-pipeline/mysql:5.7
gcr.io/ml-pipeline/persistenceagent:2.0.0-alpha.3
gcr.io/ml-pipeline/scheduledworkflow:2.0.0-alpha.3
gcr.io/ml-pipeline/viewer-crd-controller:2.0.0-alpha.3
gcr.io/ml-pipeline/visualization-server:2.0.0-alpha.3
gcr.io/ml-pipeline/workflow-controller:v3.2.3-license-compliance
gcr.io/tfx-oss-public/ml_metadata_store_server:1.5.0
gcr.io/tfx-oss-public/tfx:1.2.0
gcr.io/tfx-oss-public/tfx:1.4.0
gcr.io/tfx-oss-public/tfx:1.5.0
ghcr.io/dexidp/dex:v2.31.2
istio/pilot:1.12.0
istio/pilot:1.13.2
k8s.gcr.io/pause:3.2
kadalu/kadalu-csi:0.8.15
kadalu/kadalu-operator:0.8.15
```

```
kfexamples/mlflow-cli:1.12.0
kfexamples/sandbox:finseries-tensorflow
kfexamples/sandbox:seldon-issue-summarization
kfserving/agent:v0.6.1
kserve/agent:v0.8.0
kserve/aix-explainer:latest
kserve/alibi-explainer:latest
kserve/art-explainer:latest
kserve/kserve-controller:v0.8.0
kserve/lgbserver:v0.8.0
kserve/models-web-app:v0.8.0
kserve/paddleserver:v0.8.0
kserve/pmmlserver:v0.8.0
kserve/storage-initializer:v0.8.0
kserve/sklearnserver:v0.8.0
kserve/torchserve-kfs:0.5.3
kserve/torchserve-kfs:0.5.3-gpu
kserve/xgbserver:v0.8.0
kubeflow/training-operator:v1-e1434f6
kubeflownotebookswg/codeserver-python:v1.6.0
kubeflownotebookswg/jupyter-pytorch-cuda-full:v1.6.0
kubeflownotebookswg/jupyter-pytorch-full:v1.6.0
kubeflownotebookswg/jupyter-scipy:v1.6.0
kubeflownotebookswg/jupyter-tensorflow-cuda-full:v1.6.0
kubeflownotebookswg/jupyter-tensorflow-full:v1.6.0
kubeflownotebookswg/rstudio-tidyverse:v1.6.0
kubernetesui/dashboard:v2.6.1
kubernetesui/metrics-scraper:v1.0.8
mcr.microsoft.com/onnxruntime/server:v1.0.0
metacontroller/metacontroller:v0.3.0
minio/mc:RELEASE.2022-03-17T20-25-06Z
mpioperator/kubectl-delivery:latest
mysql:8.0.29
nginx:stable-alpine
nvcr.io/nvidia/tritonserver:21.08-py3
nvcr.io/nvidia/tritonserver:21.09-py3
python:3.7
pytorch/torchserve-kfs:0.4.0-gpu
pytorch/torchserve-kfs:0.5.3
quay.io/argoproj/argoexec:latest
quay.io/argoproj/workflow-controller:latest
quay.io/jetstack/cert-manager-cainjector:v1.5.0
quay.io/jetstack/cert-manager-controller:v1.5.0
quay.io/jetstack/cert-manager-webhook:v1.5.0
seldonio/alibiexplainer:1.12.0
seldonio/mlserver:1.0.0.rc1-alibi-explain
seldonio/mlserver:1.0.0.rc1-mlflow
seldonio/mlserver:1.0.0.rc1-sklearn
seldonio/mlserver:1.0.0.rc1-slim
seldonio/mlserver:1.0.0.rc1-xgboost
seldonio/rclone-storage-initializer:1.12.0
seldonio/sklearnserver:1.12.0
seldonio/tfserving-proxy:1.12.0
seldonio/xgboostserver:1.12.0
tensorflow/serving:1.14.0-gpu
tensorflow/serving:2.1.0
tensorflow/serving:2.6.2
tensorflow/serving:2.6.2-gpu
tensorflow/tensorflow:1.11.0-py3
tensorflow/tensorflow:2.1.0
```

Uninstalling Kubeflow

1. Rollback the Kubeflow addon:

```
# get numeric ID of the k8s cluter
kubectl get cm -n hpecp-bootstrap hpecp-bootstrap-bdconfig -o
jsonpath={.data.bds_k8s_clusterid} |  cut -d '/' -f5
```

```
K8S_CLUSTERID=1;  #<numeric ID of the k8s cluter>
KF_ADDON_NAME="kubeflow";
echo "{ok, Kfaddon} = bd_mgmt_k8s_manifest:get_addon(\"${KF_ADDON_NAME}
\")." >> /opt/bluedata/common-install/bd_mgmt/tmp.w;
echo "bd_mgmt_k8s_bootstrap:cluster_bootstrap_rollback_addons(\"$
{K8S_CLUSTERID}\", [Kfaddon])." >>/opt/bluedata/common-install/bd_mgmt/
tmp.w;
```

   Wait about 10 to 20 minutes for the Kubeflow addon to uninstall.

2. To verify successful uninstallation, check that the `kubeflow` namespace is deleted and that there are no running pods in the `hpecp-bootstrap` namespace.

3. Teardown the Kubeflow addon:

```
K8S_CLUSTERID=1;  #<numeric ID of the k8s cluter>
KF_BOOTSTRAP_DEPLOYMENT="hpecp-bootstrap-kubeflow";
echo "bd_mgmt_k8s_bootstrap:addon_teardown(\"${K8S_CLUSTERID}\", \"$
{KF_BOOTSTRAP_DEPLOYMENT}\")." >> /opt/bluedata/common-install/bd_mgmt/
tmp.w;
```

4. To verify successful teardown, check that none of the following exists:
   - `deployment` `hpecp-bootstrap-kubeflow` in namespace `hpecp-bootstrap`
   - `configmap` `hpecp-bootstrap-kubeflow` in namespace `hpecp-bootstrap`
   - `persistentvolumeclaim` `hpecp-bootstrap-kubeflow` in namespace `hpecp-bootstrap`

5. Update the database to refelct the removal of the Kubeflow add-on:

```
K8S CLUSTERID=1; #<numeric ID of the k8s cluter>
KF_ADDON_NAME="kubeflow";
echo "{ok, C} = bd_util_db_generic:get_obj(bdm_k8s_cluster, \"$
{K8S_CLUSTERID}\")." >> /opt/bluedata/common-install/bd_mgmt/tmp.w;
echo "Alist = erlang:element(11, C)." >> /opt/bluedata/common-install/
bd_mgmt/tmp.w;
echo "Nlist = erlang:subtract(Alist, [\"${KF_ADDON_NAME}\"])." >> /opt/
bluedata/common-install/bd_mgmt/tmp.w;
echo "D = erlang:setelement(11, C, Nlist)." >> /opt/bluedata/
common-install/bd_mgmt/tmp.w;
echo "bd_util_db_generic:update_obj(D)." >> /opt/bluedata/common-install/
bd_mgmt/tmp.w
```

Airflow

Describes Airflow, an open-source workflow automation and scheduling system that can be used to author and manage data pipelines.

Airflow uses workflows made of Directed Acyclic Graphs (DAGs) of tasks. You can use DAGs to collect tasks together and organize dependencies and relationships between them to determine how they should run.

The version of Airflow in HPE Ezmeral Runtime Enterprise 5.5.0 is 2.3.4.

For more information, see https://airflow.apache.org/ at Apache Documentation.

Airflow Base Entities

- `af-base-postgres`
- `af-base-nfs`

Airflow Limitations

Airflow has the following limitations:

- Airflow clusters can be installed only into the tenant namespace.
- Airflow does not support multiple authentication locations.

Airflow Requirements

Describes the system, computation, and storage requirements for Airflow clusters.

System Requirements

- Pod DNS Domain is `cluster.local`.
- Cluster authentication is configured as AD or LDAP.

Computation Requirements

You can use DAGs to create many workers that can run simultaneously. In general, for an Airflow cluster with workers, approximately 5 CPUs and 5000Mi memory are needed.

Specifically, the following resources must be available:

- To install the Airflow Operator and Base during cluster creation or editing:
  - CPU: 1800m
  - Memory: 4100Mi
- To install Airflow cluster per tenant:
  - CPU: 3000m
  - Memory: 4000Mi
- To launch one worker:
  - CPU: 1500m
  - Memory: 712Mi

Storage Requirements

Airflow requires dynamic-volume provisioning. Persistent storage must be configured and registered in the HPE Ezmeral Runtime Enterprise deployment. For information, see Storage on page 804.

Installing Airflow

Describes how to install Airflow on a Kubernetes cluster in HPE Ezmeral Runtime Enterprise.

Prerequisites

- For system, computation, and storage requirements, see Airflow Requirements on page 516.

- Authentication of the Kubernetes cluster must be set to AD or LDAP.
- You must have SSH access to the Kubernetes master node.
- **Required access rights:** Kubernetes Cluster Administrator

Procedure

1. Enable Airflow installation on the Kubernetes cluster by doing one of the following:
   - If the Kubernetes cluster has not been created, during Kubernetes cluster creation, deploy Airflow by selecting **Enable Airflow** in the **Application Configurations** tab.

     (Optional) To run Spark workflows, select **Enable Spark Operator**.

     For information about creating a Kubernetes cluster, see Creating a New Kubernetes Cluster on page 463.
   - If the Kubernetes cluster exists, deploy Airflow as follows:
     a. Edit the Kubernetes cluster, as described in Editing an Existing Kubernetes Cluster on page 480.
     b. On the **Application Configurations** tab, select **Enable Airflow**.

        (Optional) To run Spark workflows, select **Enable Spark Operator**.
     c. Save your changes.
     d. Wait until the Kubernetes cluster is ready and Airflow services are up and running.
2. Create the Airflow cluster. Select one of the following methods:
   - **Creating an Airflow Cluster Automatically**: Use this method to create an Airflow cluster through the HPE Ezmeral Runtime Enterprise UI. This is the recommended Airflow cluster creation method.

     To create the Airflow cluster automatically, see Creating an Airflow Cluster Automatically on page 517.
   - **Creating an Airflow Cluster Manually**: Use this method to perform extra tuning of your Airflow cluster. For example, if you are using a proxy server that requires authentication.

     To create the Airflow cluster manually, see Creating an Airflow Cluster Manually on page 520.

Creating an Airflow Cluster Automatically

Describes how to create an Airflow Kubernetes cluster from a Git repository through the HPE Ezmeral Runtime Enterprise UI. This is the recommended method of Airflow cluster creation.

Prerequisites

- For system, computation, and storage requirements, see Airflow Requirements on page 516.
- **Required access rights:** Platform Administrator or Tenant Administrator/Member
- Airflow is enabled on the Kubernetes cluster, as described in Installing Airflow on page 516.

About this task

NOTE: HPE Ezmeral Runtime Enterprise does not allow the creation of source control with proxy servers that require authentication. In this case, install Airflow on the Kubernetes cluster with bootstrap scripts. For more information, see Installing Airflow on page 516.

Procedure

1. Perform one of the following:
   - If you are creating an Airflow cluster in an HPE Ezmeral ML Ops project:

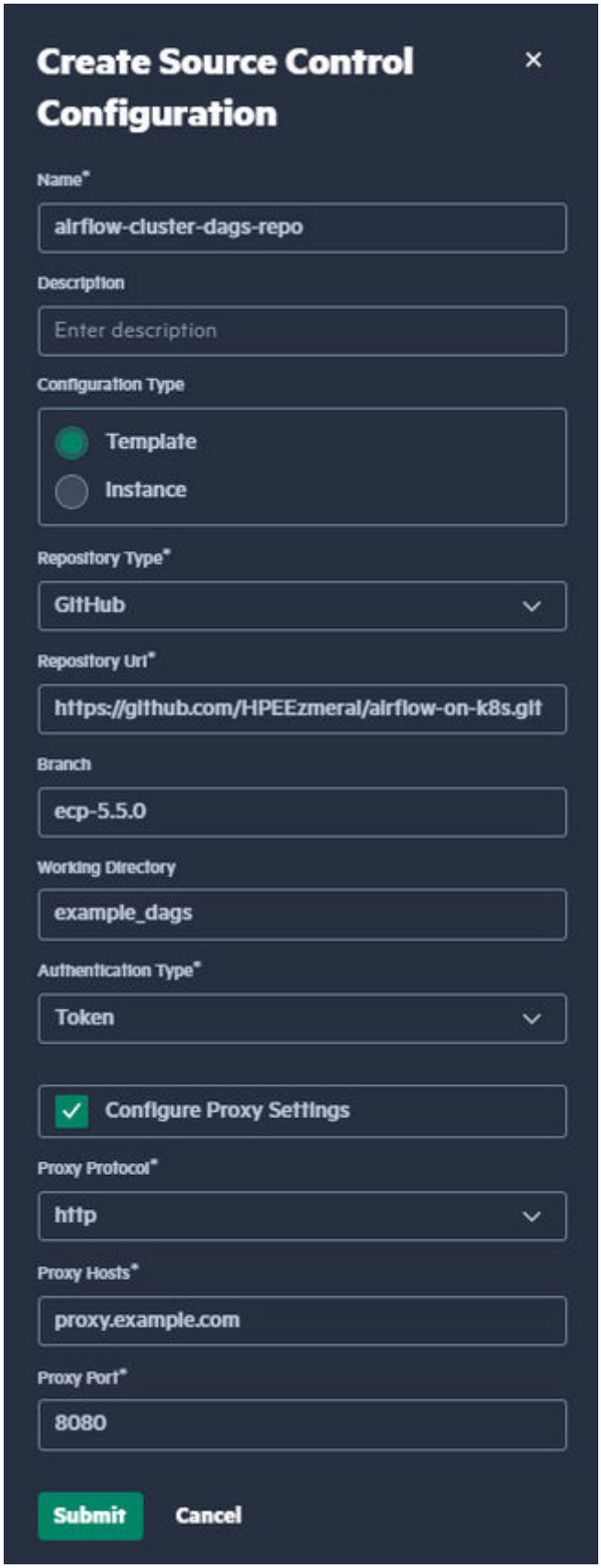
     Create a new tenant with the **ML Ops Project** check box selected. Alternatively, select the **ML Ops Project** check box on an existing tenant.
   - If you are creating an Airflow cluster for Spark in a non-HPE Ezmeral ML Ops project:

     Access the HPE Ezmeral Runtime Enterprise new UI, as described in Submitting and Managing Spark Applications Using HPE Ezmeral Runtime Enterprise new UI on page 254.

     On the **Home** page of the new UI select **View All** on the **Projects** panel. The **Projects** screen opens. Select the name of your project.
2. If your environment has a web proxy, and your HPE Ezmeral Runtime Enterprise tenant or ML Ops project has Istio Service Mesh on page 492 enabled, perform the following:

   To allow the `git clone` function in the Airflow `git-sync` container, create an Istio `ServiceEntry` object with the following web proxy details:

   ```
   cat << EOF | kubectl -n <tenant namespace> apply -f -
   apiVersion: networking.istio.io/v1alpha3
   kind: ServiceEntry
   metadata:
     name: proxy
   spec:
     hosts:
     - web-proxy.corp.hpecorp.net # ignored
     addresses:
     - 16.85.88.10/32
     ports:
     - number: 8080
       name: tcp
       protocol: TCP
     location: MESH_EXTERNAL
   EOF
   ```
3. Log in to HPE Ezmeral Runtime Enterprise as a Tenant Administrator to create Source Control templates. If you already have Source Control templates available, you can log in to HPE Ezmeral Runtime Enterprise as a Project Member.
4. Select the **ML Workbench** tab. The HPE Ezmeral Runtime Enterprise new UI opens on the **Overview** tab of the **Project details** screen in a new browser tab.
5. On the **Source Control Configurations** pane, click the name of a tenant or click **View All**. The **Source Control Configurations** screen opens.
6. Click the **Add Source Control Configuration** button. The **Create Source Control Configuration** form opens.



7. In the form, fill the required fields as follows:

- **Name**: Enter the string `airflow-cluster-dags-repo`. This source control will create a new Airflow cluster instance in this tenant.
- **Configuration Type**:

  NOTE: You must log in to HPE Ezmeral Runtime Enterprise as a Tenant Administrator to create Templates.

  If you are using a **public** Git repository, select **Template**.

  If you are using a **private** Git repository, create a **Template** with the name `airflow-cluster-dags-repo-template`. Then, create an **Instance** with the name `airflow-cluster-dags-repo`, and the `airflow-cluster-dags-repo-template` Source Control as its template.
- **Repository URL**: Enter the public or private Git repository where your DAGs are stored.
- **Branch**: Enter the name of the branch in the Git repository that you want to use.
- **Working Directory**: Enter the path to the directory where DAGs are located in the Git repository.

8. If Git is accessible behind a proxy, select the **Configure Proxy Settings** check box, and fill in the following fields:
   - **Proxy Protocol**: The protocol of the proxy (http or https).
   - **Proxy Host**: The hostname (FQDN) of the proxy server.
   - **Proxy Port**: The port of the proxy server.
9. If the Git repository is private, and you have selected **Configuration Type** as **Instance**, fill in the following fields:
   - **Username**: The username of the user with access to the repository.
   - **Email**: The email of the user with access to the repository.
   - **Token/Password**: The token or password of the user with access to the repository.
10. After filling in all necessary fields, click **Submit**. Wait for about 5 to 10 minutes.
11. Reload the page and return to the **Tenant details** page. The **Workflow Engine** link appears in the **Training and Workflow** area.

Related tasks

Installing Airflow on page 516
Describes how to install Airflow on a Kubernetes cluster in HPE Ezmeral Runtime Enterprise.

Creating an Airflow Cluster Manually on page 520
This procedure describes an alternative method of creating an Airflow Kubernetes cluster. Use this method to perform extra tuning of the Airflow cluster through the command line. However, if no extra tuning is required, use the recommended method described in Creating an Airflow Cluster Automatically on page 517.

Creating an Airflow Cluster Manually

This procedure describes an alternative method of creating an Airflow Kubernetes cluster. Use this method to perform extra tuning of the Airflow cluster through the command line. However, if no extra tuning is required, use the recommended method described in Creating an Airflow Cluster Automatically on page 517.

Prerequisites

- For system, computation, and storage requirements, see Airflow Requirements on page 516.
- **Required access rights:** Platform Administrator or Tenant Administrator/Member
- Airflow is enabled on the Kubernetes cluster, as described in Installing Airflow on page 516.

About this task

Use this method to perform extra tuning of your Airflow cluster. For example, if you are using a proxy server that requires authentication.

Procedure

1. If your environment has a web proxy, and your HPE Ezmeral Runtime Enterprise tenant or ML Ops project has Istio Service Mesh on page 492 enabled, perform the following:

   To allow the `git clone` function in the Airflow `git-sync` container, create an Istio `ServiceEntry` object with the following web proxy details:

   ```
   cat << EOF | kubectl -n <tenant namespace> apply -f -
   apiVersion: networking.istio.io/v1alpha3
   kind: ServiceEntry
   metadata:
     name: proxy
   spec:
     hosts:
     - web-proxy.corp.hpecorp.net # ignored
     addresses:
     - 16.85.88.10/32
     ports:
     - number: 8080
       name: tcp
       protocol: TCP
     location: MESH_EXTERNAL
   EOF
   ```

2. On the Kubernetes master node, open the command line.

3. Configure environment variables.

   NOTE: These environment variables are set only for this shell, and are only needed during bootstrap script execution. It is not necessary to persist them.

   Required environment variable:

`AIRFLOW_GIT_REPO_URL`	URL of the Git repository for your Directed Acyclic Graphs (DAGs). For example: `https://github.com/HPEEzmeral/airflow-on-k8s.git`

   Optional environment variables:

`AIRFLOW_CLUSTER_NAMESPACE`	Name of the namespace for AirflowCluster. This namespace should exist on the cluster.
`AIRGAP_REGISTRY`	If the environment is air gapped, address of the container registry; for example, `localhost:5000/` (the trailing slash is required).

AIRFLOW_GIT_REPO_BRANCH	The branch of the Git repository that will be used to access DAGs. For example: `ecp-5.5.0`
AIRFLOW_GIT_REPO_SUBDIR	Path to the directory where DAGs are placed in the Git repository.
GIT_PROXY_HTTP	If Git repository is located outside of the internal network, address of HTTP proxy for git-sync container.
GIT_PROXY_HTTPS	If Git repository is located outside of the internal network, address of HTTPS proxy for git-sync container.

Default values for environment variables are as follows.

```
AIRGAP_REGISTRY=""
AIRFLOW_CLUSTER_NAMESPACE="default"
AIRFLOW_CLUSTER_IMAGE_TAG="ecp-5.5.0-rc1"
AIRFLOW_BASE_NAMESPACE="airflow-base"
AIRFLOW_GIT_REPO_BRANCH="" #empty string points to main branch of git
repo
AIRFLOW_GIT_REPO_SUBDIR=""
GIT_PROXY_HTTP=""
GIT_PROXY_HTTPS=""
```

4. From the following location, clone the repository branch that corresponds to the release of HPE Ezmeral Runtime Enterprise that your environment is running:

   https://github.com/HPEEzmeral/airflow-on-k8s

5. Install the Airflow cluster using one of the following options:

   - Option 1: Public Git repository shell script

     For example:

     ```
     AIRFLOW_GIT_REPO_URL="https://github.com/HPEEzmeral/
     airflow-on-k8s.git" \
     AIRFLOW_GIT_REPO_SUBDIR="example_dags/"
     AIRFLOW_GIT_REPO_BRANCH="ecp-5.5.0" \
     /bin/sh airflow-on-k8s/bootstrap/airflow-cluster/install.sh
     ```

   - Option 2: Private Git repository shell script

     - If the password (or access token) of the Git repository is already stored in secret by key password within the `AIRFLOW_CLUSTER_NAMESPACE` namespace, additionally pass it the name in `AIRFLOW_GIT_REPO_CRED_SECRET_NAME` variable and pass the user name in `AIRFLOW_GIT_REPO_USER` variable.

       For example:

       ```
       AIRFLOW_GIT_REPO_URL="https://github.com/HPEEzmeral/
       airflow-on-k8s.git" \
       AIRFLOW_GIT_REPO_SUBDIR="example_dags/"
       AIRFLOW_GIT_REPO_BRANCH="ecp-5.5.0" \
       AIRFLOW_GIT_REPO_USER="mapr" \
       AIRFLOW_GIT_REPO_CRED_SECRET_NAME="secret-with-git-creds" \
       /bin/sh airflow-on-k8s/bootstrap/airflow-cluster/install.sh
       ```

- If the password (or access token) is not already stored in secret, pass the user name in the `AIRFLOW_GIT_REPO_USER` variable, then execute the following command. The script generates an appropriate secret, and, after the script runs, passes credentials at the prompt.

  For example:

  ```
  AIRFLOW_GIT_REPO_URL="https://github.com/HPEEzmeral/
  airflow-on-k8s.git" \
  AIRFLOW_GIT_REPO_SUBDIR="example_dags/"
  AIRFLOW_GIT_REPO_BRANCH="ecp-5.5.0" \
  AIRFLOW_GIT_REPO_USER="mapr" \
  /bin/sh airflow-on-k8s/bootstrap/airflow-cluster/install.sh
  ```

Related tasks

Installing Airflow on page 516
Describes how to install Airflow on a Kubernetes cluster in HPE Ezmeral Runtime Enterprise.

Creating an Airflow Cluster Automatically on page 517
Describes how to create an Airflow Kubernetes cluster from a Git repository through the HPE Ezmeral Runtime Enterprise UI. This is the recommended method of Airflow cluster creation.

Accessing Data From Outside Airflow DAGs with DataTap

Describes how to access data from tenant storage in Airflow DAGs using DataTap. This can be done with either Airflow BashOperator or Airflow PythonOperator.

DataTap can be used to access data stored outside DAGs. For example, large datasets, binaries, or other large files which cannot be uploaded to Git repositories can instead be uploaded to DataTap tenant storage. These files can then be accessed in any Airflow DAG.

There are two ways to read and write data from tenant storage in Airflow DAGs using DataTap:

- Airflow BashOperator on page 523
- Airflow PythonOperator on page 523

Airflow BashOperator

You can learn about Airflow BashOperator on the Apache site: Airflow BashOperator (link opens an external site in a new browser tab or window).

To access data from tenant storage in DAGs using DataTap with Airflow BashOperator, proceed as follows:

1. Enter the hadoop CLI command to perform operations with tenant data. Example of `bash_command` argument in the DAG:

   ```
   bash_command='hadoop fs -ls dtap://TenantStorage/' + path
   ```

   NOTE: The full path to tenant storage is:

   ```
   dtap://TenantStorage/
   ```

2. For a full example, see this page.

Airflow PythonOperator

You can learn about Airflow PythonOperator on the Apache site: Airflow PythonOperator (link opens an external site in a new browser tab or window).

To access data from tenant storage in DAGs using DataTap with Airflow PythonOperator, proceed as follows:

1. Use the `pyarrow` Python library to access data in tenant storage:

   ```
   from pyarrow import fs
   ```

   See: Pyarrow Python library (link opens an external site in a new browser tab or window).

2. For a full example, see this page.

Notes about using Airflow

To execute Airflow jobs, you must be an AD/LDAP user that is a member of the tenant where Airflow is installed.

Health Checks

If a database failure occurs, the database pod persists the PersistentVolumeClaim (PVC) and cluster meta information. However, Airflow uses SQLAlchemy, and sometimes the Airflow Scheduler pod loses the connection during database pod failures. One way to automate connection checks is to use Scheduler HealthCheck, where the cluster admin or user restarts or creates a trigger if a connection failure occurs.

See Checking Airflow Health Status in the Apache Airflow documentation (link opens an external web page in a new browser tab or window).

Accessing Logs after Removing Pods

In HPE Ezmeral Runtime Enterprise, any completed pod can be automatically removed within some period of time. After a pod is deleted, you can no longer access logs.

Accessing the Web UI

You can access the Airflow UI as follows:

1. Access the HPE Ezmeral Runtime Enterprise new UI, as described in HPE Ezmeral Runtime Enterprise new UI on page 146.

2. Select **Workflow Engine**:

   - For HPE Ezmeral ML Ops projects, **Workflow Engine** is located on the **Training and Workflow** panel under the **Model Building** section.

   - For non-HPE Ezmeral ML Ops projects, **Workfow Engine** is located on the **Workflow** panel under the **Notebook Servers and Workflow** section.

You can obtain the FQDN of the Airflow web UI as follows:

```
kubectl describe svc airflow-https-svc -n <cluster-namespace>
```

In `Annotations`, obtain the address of the UI as follows:

```
mip-bd-ap05-n2-vm05.mip.storage.hpecorp.net:10007
```

After creating a new Airflow cluster, port numbers of other clusters can be changed. If any issues with the gateway occur, you can port-forward 8080 port of the `af-cluster-airflowui-0` pod in the cluster namespace.

Accessing Data From Outside DAGs with DataTap

See Accessing Data From Outside Airflow DAGs with DataTap on page 523.

Run DAGs with SparkKubernetesOperator

See Using Airflow to Schedule Spark Applications on page 314

List of Airflow Images

Images Used by Airflow

```
gcr.io/mapr-252711/airflow:ecp-5.5.0-rc1
gcr.io/mapr-252711/airflow-operator:ecp-5.5.0-rc1
k8s.gcr.io/git-sync/git-sync:v3.3.4
k8s.gcr.io/volume-nfs:0.8
pbweb/airflow-prometheus-exporter:latest
postgres:9.5
bluedata/hpecp-dtap:1.8.0
```

Python Script to Load Images into Air-Gap Docker Registry

In the following example, replace `localhost:5000` with the address of your air-gap Docker registry.

```
import os

if __name__ == "__main__":
  images = [
    "gcr.io/mapr-252711/airflow:ecp-5.5.0-rc1",
    "gcr.io/mapr-252711/airflow-operator:ecp-5.5.0-rc1",
    "k8s.gcr.io/git-sync/git-sync:v3.3.4",
    "pbweb/airflow-prometheus-exporter:latest",
    "k8s.gcr.io/volume-nfs:0.8",
    "postgres:9.5",
    "bluedata/hpecp-dtap:1.8.0",
  ]
  for x in images:
    os.system("docker pull " + x)
    os.system("docker image tag " + x + " localhost:5000/" + x)
    os.system("docker push localhost:5000/" + x)
```

List of Airflow Providers

Airflow Providers

The capabilities of Airflow can be extended with the use of additional packages, called providers. Providers can obtain operators, hooks, sensor, and transfer operators to communicate with a multiutde of external systems.

All Airflow providers available in HPE Ezmeral Runtime Enterprise are listed in the table below. For more details on the use of a particular package, select the links provided in the table.

Table

Package Name	Version	Descirption
apache-airflow-providers-airbyte	2.1.4	Airbyte
apache-airflow-providers-alibaba	1.1.1	Alibaba Cloud integration (including Alibaba Cloud).
apache-airflow-providers-amazon	3.3.0	Amazon integration (including Amazon Web Services (AWS)).

Table (Continued)

Package Name	Version	Descirption
apache-airflow-providers-apache-beam	3.3.0	Apache Beam.
apache-airflow-providers-apache-cassandra	2.1.3	Apache Cassandra.
apache-airflow-providers-apache-drill	1.0.4	Apache Drill.
apache-airflow-providers-apache-druid	2.3.3	Apache Druid.
apache-airflow-providers-apache-hdfs	2.2.3	Hadoop Distributed File System (HDFS)and WebHDFS.
apache-airflow-providers-apache-hive	2.3.2	Apache Hive
apache-airflow-providers-apache-kylin	2.0.4	Apache Kylin
apache-airflow-providers-apache-livy	2.2.3	Apache Livy
apache-airflow-providers-apache-pig	2.0.4	Apache Pig
apache-airflow-providers-apache-pinot	2.0.4	Apache Pinot
apache-airflow-providers-apache-spark	2.1.3	Apache Spark
apache-airflow-providers-apache-sqoop	2.1.3	Apache Sqoop
apache-airflow-providers-arangodb	1.0.0	ArangoDB
apache-airflow-providers-asana	1.1.3	Asana
apache-airflow-providers-celery	2.1.4	Celery
apache-airflow-providers-cloudant	2.0.4	IBM Cloudant
apache-airflow-providers-cncf-kubernetes	4.0.0	Kubernetes
apache-airflow-providers-databricks	2.6.0	Databricks
apache-airflow-providers-datadog	2.0.4	Datadog
apache-airflow-providers-dbt-cloud	1.0.2	dbt Cloud
apache-airflow-providers-dingding	2.0.4	Dingding
apache-airflow-providers-discord	2.1.4	Discord
apache-airflow-providers-docker	2.6.0	Docker
apache-airflow-providers-elasticsearch	3.0.3	Elasticsearch
apache-airflow-providers-exasol	2.1.3	Exasol
apache-airflow-providers-facebook	2.2.3	Facebook Ads
apache-airflow-providers-ftp	2.1.2	File Transfer Protocol (FTP)
apache-airflow-providers-github	1.0.3	GitHub

last-updated: May 06, 2024

Table (Continued)

Package Name	Version	Descirption
apache-airflow-providers-google	6.8.0	Google services including: • Google Ads • Google Cloud (GCP) • Google Firebase • Google LevelDB • Google Marketing Platform • Google Workspace (formerly Google Suite)
apache-airflow-providers-grpc	2.0.4	gRPC
apache-airflow-providers-hashicorp	2.2.0	Hashicorp including Hashicorp Vault
apache-airflow-providers-http	2.1.2	Hypertext Transfer Protocol (HTTP)
apache-airflow-providers-imap	2.2.3	Internet Message Access Protocol (IMAP)
apache-airflow-providers-influxdb	1.1.3	InfluxDB
apache-airflow-providers-jdbc	2.1.3	Java Database Connectivity (JDBC)
apache-airflow-providers-jenkins	2.1.0	Jenkins
apache-airflow-providers-jira	2.0.4	Atlassian Jira
apache-airflow-providers-microsoft-azure	3.8.0	Microsoft Azure
apache-airflow-providers-microsoft-mssql	2.1.3	Microsoft SQL Server (MSSQL)
apache-airflow-providers-microsoft-psrp	1.1.4	This package provides remote execution capabilities via the PowerShell Remoting Protocol (PSRP).
apache-airflow-providers-microsoft-winrm	2.0.5	Windows Remote Management (WinRM)
apache-airflow-providers-mongo	2.3.3	MongoDB
apache-airflow-providers-mysql	2.2.3	MySQL
apache-airflow-providers-neo4j	2.1.3	Neo4j
apache-airflow-providers-odbc	2.0.4	ODBC
apache-airflow-providers-openfaas	2.0.3	OpenFaaS
apache-airflow-providers-opsgenie	3.0.3	Opsgenie
apache-airflow-providers-oracle	2.2.3	Oracle
apache-airflow-providers-pagerduty	2.1.3	Pagerduty
apache-airflow-providers-papermill	2.2.3	Papermill
apache-airflow-providers-plexus	2.0.4	Plexus
apache-airflow-providers-postgres	4.1.0	PostgreSQL

last-updated: May 06, 2024

Table (Continued)

Package Name	Version	Descirption
apache-airflow-providers-presto	2.2.0	Presto
apache-airflow-providers-qubole	2.1.3	Qubole
apache-airflow-providers-redis	2.0.4	Redis
apache-airflow-providers-salesforce	3.4.3	Salesforce
apache-airflow-providers-samba	3.0.4	Samba
apache-airflow-providers-segment	2.0.4	Segment
apache-airflow-providers-sendgrid	2.0.4	Sendgrid
apache-airflow-providers-sftp	2.6.0	SSH File Transfer Protocol (SFTP)
apache-airflow-providers-singularity	2.0.4	Singularity
apache-airflow-providers-slack	4.2.3	Slack
apache-airflow-providers-snowflake	2.6.0	Snowflake
apache-airflow-providers-sqlite	2.1.3	SQLite
apache-airflow-providers-ssh	2.4.3	Secure Shell (SSH)
apache-airflow-providers-tableau	2.1.7	Tableau
apache-airflow-providers-telegram	2.0.4	Telegram
apache-airflow-providers-trino	2.2.0	Trino
apache-airflow-providers-vertica	2.1.3	Vertica
apache-airflow-providers-yandex	2.2.3	Yandex including Yandex.Cloud
apache-airflow-providers-zendesk	3.0.3	Zendesk

Kubernetes Hosts

The topics in this section describe information and tasks related to Kubernetes Hosts on HPE Ezmeral Runtime Enterprise.

Installing Kubernetes Hosts

The topics in this section describe information and tasks related to installing Kubernetes hosts on HPE Ezmeral Runtime Enterprise.

Kubernetes Worker Installation Overview

Describes how to add a host to HPE Ezmeral Runtime Enterprise as a Kubernetes worker for compute workloads.

Prerequisites

- Hewlett Packard Enterprise recommends enabling platform High Availability before adding a large number of Kubernetes.

- Ensure that hosts conform to the requirements described in Host Requirements on page 813 and Kubernetes Host/Node Requirements on page 833.

  If the `firewalld` service is installed and enabled on the Controller, and the `firewalld` service is installed and enabled on all hosts before they are added to the deployment, the installer for HPE Ezmeral Runtime Enterprise automatically configures firewall rules to open the required ports listed in Port Requirements on page 809 and Kubernetes Port Requirements on page 836.

  CAUTION:

  Numerous configuration changes occur to the host during installation that are required in order for the platform to function. These changes are not completely reversible and may impact any other applications and processes that are currently running on the host. It is strongly recommended that you install HPE Ezmeral Runtime Enterprise on a host that is not being used for any other purpose in order to avoid possible disruptions to your business processes.

  Installing HPE Ezmeral Runtime Enterprise on any host that does not meet all applicable requirements may lead to unpredictable behavior and/or data loss.

- For best results, it is recommended that all compute hosts in a cluster share the same configuration (CPU, RAM, storage, OS, etc.).

- If this host has MIG-enabled GPUs that are supported by HPE Ezmeral Runtime Enterprise, install the NVIDIA driver on the host and configure MIG before adding the host to HPE Ezmeral Runtime Enterprise.

  See Deploying MIG Support on page 840.

- If you want to install the Falco Kernel Module on the host as part of the Falco Container Runtime Security feature, install the module on the host after you install the host OS but before you add the host to HPE Ezmeral Runtime Enterprise.

- See Falco Container Runtime Security on page 499.

About this task

This article describes adding Kubernetes hosts for compute workloads.

- If you visited this article intending to add Data Fabric nodes, see Kubernetes Data Fabric Node Installation Overview on page 531.

- If you visited this article intending to add Shadow Controller or Arbiter hosts, see Enabling Platform High Availability on page 740.

- If you visited this article intending to add a gateway host, see Installing a Gateway Host on page 758.

Procedure

1. Prepare the hosts to be added as Kubernetes hosts.

   - If your environment is running the SSHD service (see Configuration Requirements on page 826), then skip to Kubernetes Host: Add the Public SSH Key on page 537.

   - If your environment does not allow key-based SSH login, then proceed to Agent-Based Kubernetes Host Installation on page 532.

2. If you are adding hosts to expand a Kubernetes cluster that has not been migrated to the Hewlett Packard Enterprise distribution of Kubernetes, create the following touch file on each host:

   ```
   touch /tmp/k8s_docker_override
   ```

   Creating the touch file specifies that the Docker container runtime is used instead of the containerd runtime.

3. In the web interface, select the hosts to add as Kubernetes Workers.

   See Kubernetes Host: Select the Hosts on page 538.

4. Add the Worker hosts.

   See Kubernetes Host: Add the Hosts on page 539.

5. Select the hard drives on the Worker hosts.

   See Kubernetes Host: Select Hard Drives on page 540.

6. Place HPE Ezmeral Runtime Enterprise into Lockdown mode.

   For more information, see Kubernetes Host: Enter Lockdown Mode on page 542.

7. Install the hosts as Kubernetes Workers.

   See Kubernetes Host: Add the Hosts as Workers on page 542.

   HPE Ezmeral Runtime Enterprise verifies that the number of CPU cores in the hosts do not exceed the licensed maximum, and then proceeds with the installation. The UI displays a green **Installing** bar for each of the new hosts.

8. Exit Lockdown mode.

9. On each host, prevent the `yum update` command from updating the Kubernetes repo by setting `enabled=0` in the following file: `/etc/yum.repos.d/bd-kubernetes.repo`

10. Validate that the new Kubernetes Worker has been correctly added and is functioning properly.

    See Kubernetes Host: Validate the Worker Installation on page 543.

Related reference

Deploying MIG Support on page 840
This topic describes how to configure and deploy a supported MIG-enabled GPU on HPE Ezmeral Runtime Enterprise.

Air Gap Tab on page 799
The **Air Gap** tab of the **System Settings** screen enables the Platform Administrator to specify settings to be used when the Kubernetes hosts, clusters, and tenants do not have connectivity to the Internet.

More information

Falco Container Runtime Security on page 499
The Falco Container Runtime Security feature of HPE Ezmeral Runtime Enterprise improves container security and threat detection.

Kubernetes Air-Gap Requirements on page 834

Using the Air Gap Utility on page 869
Describes how to use the air gap utility to download files in an air-gapped HPE Ezmeral Runtime Enterprise environment.

Kubernetes Data Fabric Node Installation Overview

NOTE:

This article describes adding Data Fabric nodes for storage.

- If you visited this article intending to add Kubernetes hosts for compute workloads, then please see Kubernetes Worker Installation Overview.

Before adding one or more Kubernetes Data Fabric nodes, be sure that the nodes conform to the requirements described in Host Requirements and Kubernetes Host Requirements. For best results, it is recommended that all Data Fabric nodes share the same configuration (CPU, RAM, storage, OS, etc.) as other Data Fabric nodes.

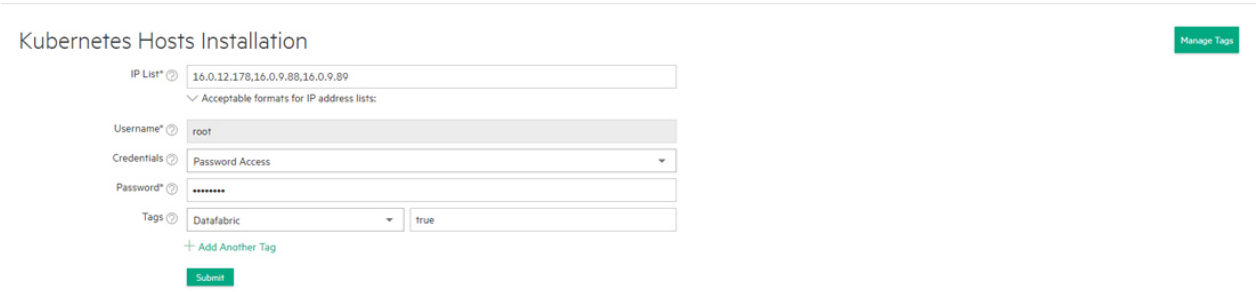
CAUTION: Installing HPE Ezmeral Runtime Enterprise on any node that does not meet all applicable requirements may lead to unpredictable behavior and/or data loss.

CAUTION: Numerous configuration changes occur to the node during installation that are required in order for the platform to function. These changes are not completely reversible and may impact any other applications and processes that are currently running on the node. It is strongly recommended that you install HPE Ezmeral Runtime Enterprise on a node that is not being used for any other purpose in order to avoid possible disruptions to your business processes.

NOTE: Please see Gateway Installation Tab for instructions on installing Gateway hosts.

Adding one or more Data Fabric nodes uses the following basic process:

1. In the **Kubernetes Hosts Installation** screen, be sure to select the `Datafabric` tag and then set the value to either `yes`, `YES`, `true`, or `TRUE` using the **Tags** pull-down menus, as described in The Kubernetes Hosts Installation Screen.
   - Do this for all Worker nodes.
   - Do not set the `Datafabric` tag for the Master nodes that you will use in a Data Fabric cluster.



2. Install HPE Ezmeral Runtime Enterprise on the nodes.
   - If your environment is running the SSHD service (see Configuration Requirements), then skip to Kubernetes Host Step 1: Add the Public SSH Key.
   - If your environment does not allow key-based SSH login, then proceed to Agent-Based Kubernetes Host Installation.
3. In the web interface, select the nodes to add as Kubernetes Workers. See Kubernetes Host Step 2: Select the Host(s).
4. Add the Worker nodes. See Kubernetes Host Step 3: Adding the Host(s).
5. Select the hard drives on the Worker nodes. See Kubernetes Host Step 4: Select Hard Drives.

6. Place HPE Ezmeral Runtime Enterprise into Lockdown mode, as described in Kubernetes Host Step 5: Enter Lockdown Mode.

7. Install the nodes as Kubernetes Workers. See Kubernetes Host Step 6: Add the Host(s) as Worker(s). HPE Ezmeral Runtime Enterprise will validate that the number of CPU cores in the nodes do not exceed the licensed maximum before proceeding with the installation (and displaying the green **Installing** bar for the new nodes).

8. Exit Lockdown mode, and then validate that the new Kubernetes Worker has been correctly added and is functioning properly. See Kubernetes Host Step 7: Validate the Worker Installation.

Agent-Based Kubernetes Host Installation

If your environment does not allow key-based SSH, then you must run the command line agent installation described in this article on each Kubernetes Worker host being added before adding the hosts using the web interface.

NOTE: These instructions assume that the Controller host was installed with the option `--worker-agent-install`. If that was not done and if you do not want to reinstall the Controller host with that option specified, then please contact HPE Technical Support for possible options.

NOTE: If your environment does allow key-based SSH on all of the hosts, then you may bypass this step and proceed directly to Kubernetes Host Step 1: Add the Public SSH Key.

To install the agent on each Kubernetes host:

1. If you encountered any errors while pre-checking and/or installing HPE Ezmeral Runtime Enterprise on the Controller from the command line, then be sure to replicate the same remediation steps on each Worker host you will be adding before proceeding with the installation.

2. Copy the `.erlang.cookie` file from the Controller host to the Kubernetes hosts you are adding. This file is located in the home directory of the user who installed HPE Ezmeral Runtime Enterprise. This step is required to allow secure communications between hosts.

3. Manually copy the HPE Ezmeral Runtime Enterprise Enterprise binary (.bin) from `http://<controller-ip>/repos/common-cp-<os>-release-<version>-<build>.bin` to each Worker host that you will adding, where:

   - `<controller_ip>` is the IP address of the Controller host.
   - `<os>` is the operating system (either `rhel` or `sles`).
   - `<version>` is the .bin version.
   - `<build>` is the specific .bin build number.

   NOTE: The remainder of this article will refer to this .bin file as `<common>.bin`.

4. Make the .bin file executable by executing the command `chmod a+x <common>.bin.`

5. Download the `.parms` file from `http://<controller-ip>/repos/agent-install-worker.params`

6. Modify the relevant settings in `/tmp/agent-install-worker.parms` to the appropriate values. The `.params` file with these edits will be used on every Kubernetes Worker host.

   - **Set the Controller host parameter:** The Controller parameter settings vary based on whether or not platform HA is enabled.

- If platform HA is not enabled, then you must set the `HAENABLED` (platform High Availability Enabled) field to `false` and provide both the Controller IP address and hostname in the `Platform HA not configured` section.

```
##################################################################
###########
                        #                          Platform HA not
configured                            #
                        # Ensure the appropriate parameters are
uncommented and set in this section    #
                        # when Platform HA is not
enabled.                                                  #

##################################################################
###########

                        ## Is PLHA enabled?
                        #HAENABLED=false
```

Note: Uncomment this.

```
## Controller node's IP address.
                        #CONTROLLER=<Controller IP address>
```

Note: Uncomment this and provide the Controller host IP address.

```
## Controller node's FQDN.
                        #CONTROLLER_HOSTNAME=<FQDN of controller>
```

*Note: Uncomment this and provide the Controller hostname. The **Controller hostname** must be **all lower case** set as per the Linux hostname naming convention.*

- If platform HA is enabled, then you must set the `HAENABLED` (Platform High Availability Enabled) field to `false` and provide both the IP address and hostname for the Controller, Shadow Controller, and Arbiter hosts in the `Platform HA configured` section.

  Further, if the deployment uses a Cluster IP address, then you must set `CLUSTERIP` (Cluster IP address); otherwise, you can leave it commented.

```
#######################################################################
###########
                        #                              Platform HA
configured                              #
                        # Ensure the appropriate parameters are
uncommented and set in this section    #
                        # when Platform HA is not
enabled.                                               #

#######################################################################
###########

                        ## Is Platform HA enabled?
                        #HAENABLED=true
```

  Note: Uncomment this.

```
## The cluster IP address.
                        #CLUSTERIP=<Cluster IP address>
```

  Note: Uncomment this if a Cluster IP address is used.

```
## Controller node's IP address. A failover to okay but, his node
must be alive
                        ## for a worker to be added.
                        #CONTROLLER=<Controller IP address>
```

  Note: Uncomment this and provide the Controller IP address.

```
## The original shadow controller node's IP address. This node must
be alive for
                        ## the worker node to be added.
                        #SHADOWCTRL=<Shadow IP address>
```

  Note: Uncomment this and then provide the Shadow IP address.

```
## The arbiter node's IP address. This node must be alive for the
worker node to
                        ## be added.
                        #ARBITER=<Arbiter IP address>
```

  Note: Uncomment this and then provide the Arbiter IP address.

```
## Controller node's FQDN.
                        #CONTROLLER_HOSTNAME=<FQDN of controller>
```

  Note: Uncomment this and then provide the Controller hostname.

```
## Shadow controller node's FQDN.
                        #SHADOW_HOSTNAME=<FQDN of Shadow>
```

*Note: Uncomment this and then provide the Shadow hostname. The **Shadow hostname** must be **all lower case** set as per the Linux hostname naming convention.*

```
## Arbiter node's FQDN.
                        #ARBITER_HOSTNAME=<FQDN of Arbiter>
```

*Note: Uncomment this and then provide the Arbiter hostname. The **Arbiter hostname** must be **all lower case** set as per the Linux hostname naming convention.*

- **Set the installation userid and groupid parameters:** If you have already created a system account on the Controller host, then you will need to set the `BLUEDATA_USER` and `BLUEDATA_GROUP` values accordingly.

```
##########################################################################
#########
                #                          Installation user and
group                           #
                # All nodes in the HPE physical cluster must be
installed as the same user.    #
                # Specify this if the common bundle is not being
executed by the same user as  #
                # the user that will be running the HPE services.
Please refer to the          #
                # System requirements guide for information on
permissions required for a      #
                # non-root user to install and run HPE
software.                                #

##########################################################################
#########

                #BLUEDATA_USER=root
```

Note: Uncomment this and then provide the user id, as appropriate.

```
#BLUEDATA_GROUP=root
```

Note: Uncomment this and then provide the group id, as appropriate.

- **Set other miscellaneous parameters:** Set the following parameters to match the Controller host settings.

```
#######################################################################
#########
                #                           Miscellaneous
parameters                           #

#
        #

#######################################################################
#########

                ## Automount root on the controller node. It must be
the same on the worker too.
                CONTROLLER_AUTOMOUNT_ROOT=/net/
```

Note: Modify this if needed.

```
## Bundle flavor used to install the controller. This may be either
'minimal' or
                ## 'full'
                CONTROLLER_BUNDLE_FLAVOR=minimal
```

Note: Modify this if needed.

```
## Skip configuring NTP? 'true' or 'false'
                #NO_NTP_CONFIG=false
```

Note: Modify this, as appropriate.

```
## If the controller was configured with proxy information, please
specify it


                ## for the worker too.


                #PROXY_URL=
```

Note: Set this if the Controller is configured with a proxy.

```
#NO_PROXY=
```

Set this is the Controller was configured with the `--no-proxy` option during installation.

```
## Controls whether the server should rollback to a clean state when
an error
                ## is encountered during installation. Setting it to
'false' helps with debugging
                ## but the server should be manually cleaned up before
re-attempting the
                ## installation.

                ## Values: 'true' or 'false'.
                #ROLLBACK_ON_ERROR='false'

                # If the controller was configured
```

```
with --dockerrootsize that is different from 20
                # specify it here.
                DOCKER_ROOTSIZE=20
```

Note: Set this, if applicable.

7. Set the Erlang parameter:

```
ERLANG_COOKIE=value stored in <controller-ip>$HOME/.erlang.cookie
```

8. Copy the modified version of the `.parms` file onto every new Kubernetes Worker host.

9. On each Worker host, execute the installer precheck using one of the following commands, where `<A.B.C.D>` is the IP address of the host, and `<name>` is the FQDN of the host:

   - **Kubernetes host:**`/tmp/<precheck>.bin --params /tmp/agent-install-worker.parms --nodetype k8shost --worker <A.B.C.D> --workerhostname <name>`

10. If needed, remediate any issues reported by the installer script, and then re-run the same pre-check script until all tests pass or until you have accounted for any warnings.

11. Run the common install ,bin:

```
<controller-ip>/opt/bluedata/bundles/common-cp-<version>-<build>.bin
```

12. Copy the file `/opt/bluedata/keys/authorized_keys` from the Controller host to the same location on the new Kubernetes Worker host, with the same owner/group, permissions, and SELinux context.

NOTE: Hewlett Packard Enterprise recommends to update to the latest OS packages (e.g. yum update) before installing HPE Ezmeral Runtime Enterprise.

After the installation completes, you should see the message `Successfully prepared server as a HPE CP Kubernetes node.` Proceed directly to Kubernetes Host: Select the Hosts on page 538, as appropriate.

If the installation fails, then erase HPE Ezmeral Runtime Enterprise from the host by executing the command `/tmp/<common>.bin --erase` (or `sudo /tmp/<common>.bin --erase`, or `SUDO_PREFIX="mysudo"; /tmp/<common>.bin --erase`. The instructions contained in Step 1 Troubleshooting on page 860 for the Controller host can also help you remediate problems on this host or hosts.

Using Passwordless SSH

The topics in this section describe information and tasks related to installing Kubernetes hosts on HPE Ezmeral Runtime Enterprise using the passwordless SSH method.

Kubernetes Host: Add the Public SSH Key

About this task

NOTE:

This procedure is only needed if you are adding a Kubernetes Worker host with SSH key-based access.

If you are using the agent installation as described in Agent-Based Kubernetes Host Installation on page 532, skip this procedure and go to Kubernetes Host: Select the Hosts on page 538.

You must upload the public key to the Kubernetes Worker hosts before uploading the corresponding private key to add those hosts via the web interface. The following procedure assumes that you created the keys using a tool like `ssh-keygen`.

CAUTION: The key must be in PEM format.

Procedure

1. Execute the command `ssh-keygen -m PEM -t rsa #` and then follow the onscreen instructions.
2. Copy the `id_rsa.pub` file to the Kubernetes Worker host.
3. Add the public key to the list of authorized keys for the root user. Execute a command similar to the following:

   ```
   cat id_rsa.pub >> /root/.ssh/authorized_keys
   ```

4. Test the key by executing the `ssh -i id_rsa root@<worker>` command from the Controller host (where `<worker>` is the host name or IP address of the Kubernetes Worker host).
   This command should log the root user into the Kubernetes Worker host without being prompted for a password.
5. Proceed to Kubernetes Host: Select the Hosts on page 538.

Kubernetes Host: Select the Hosts

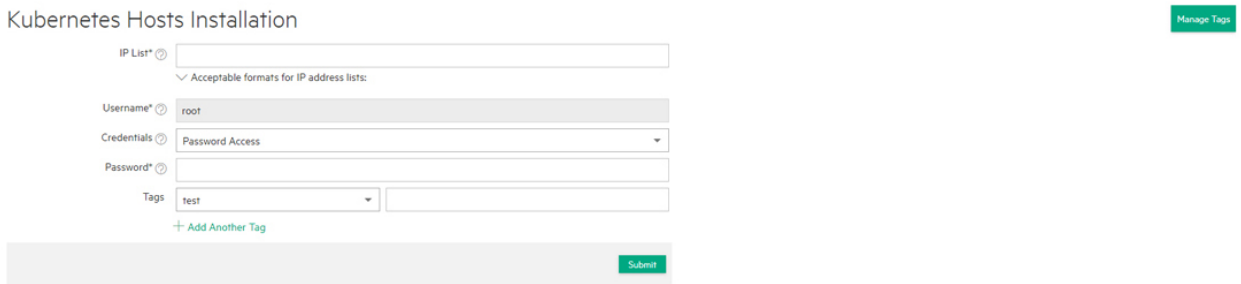
Prerequisites

You will arrive at this step after one of the following:

- Installing the agent on the Kubernetes Worker hosts as described in Agent-Based Kubernetes Host Installation on page 532.
- Adding the public SSH key as described in Kubernetes Host: Add the Public SSH Key on page 537.

About this task

The next step in adding one or more Kubernetes Worker hosts is to select the hosts using the top portion of the **Kubernetes Hosts Installation** screen (see The Kubernetes Hosts Installation Screen on page 551).



Procedure

1. If needed, add one or more tags by clicking the **Manage Tags** button and then adding the tags as described in Adding a New Tag on page 548.

2. Enter the IP addresses of the Kubernetes Worker hosts that you are adding in the **Worker IP** box. You may select one or more hosts as follows:
   - **Single IP address:** To add a single host, enter a properly formatted IP address, such as `10.10.1.1`.
   - **Multiple IP addresses:** Enter the first three octets of the IP addresses, and then separate each digit of the fourth octet with a comma, such as `10.10.1.1,2,5,8`. The preceding example adds four Kubernetes Worker hosts with IP addresses of 10.10.1.1, 10.10.1.2, 10.10.1.5, and 10.10.1.8.
   - **Multiple IP addresses:** Enter multiple IP addresses separated by commas, such as `10.10.1.1`, `10.10.1.2`, `10.10.1.5`, `10.10.1.8`. The preceding example adds four Kubernetes Worker hosts.
   - **IP address range:** Enter an IP address range, such as `10.10.1.1-8`. The preceding example adds eight Kubernetes Worker hosts with IP addresses from 10.10.1.1 to 10.10.1.8.
   - **Combination:** Use a combination of the preceding methods, such as `10.10.1.1`, `10.10.1.2,5,8, 10.10.1.9-12`.

   NOTE: You may only perform one set of Kubernetes host additions at a time. To save time, consider adding all the Kubernetes Worker hosts at once by entering multiple IP addresses as described previously.

3. If needed, assign one or more tags to the hosts you are adding, as described in Assigning Tags to a Host on page 549.

4. Select how to access the Kubernetes Worker hosts. Your available options are the following:
   - **Agent:** If you installed the agent on the Kubernetes Worker hosts as described in Agent-Based Kubernetes Host Installation on page 532, then you will not see any credential or key options and should proceed directly to Kubernetes Host: Add the Hosts on page 539.

   - **Password access:** Select the **Password Access** radio button and then enter the password for the Kubernetes Worker hosts you are adding in the **Password** boxes. The password must be valid for the user name in the **User name** box.
   - **SSH Key:** If the Kubernetes Worker hosts already have a public key installed to enable password-free access (see Kubernetes Host: Add the Public SSH Key on page 537), then you may select the **SSH Key based Access** radio button. Upload the private key by clicking the **Browse** button to open a standard **File Upload** dialog that enables you to browse for and select the key file. If the key requires a passphrase, enter that phrase in the **Passphrase** box. The uploaded private key will only be used for initial host access and will not be permanently stored.

5. Proceed to Kubernetes Host: Add the Hosts on page 539.

Kubernetes Host: Add the Hosts

Prerequisites

You have selected the Kubernetes Worker hosts and entered any required credentials as described in Kubernetes Host: Select the Hosts on page 538.

About this task

The next step is to click the **Submit** button to install HPE Ezmeral Runtime Enterprise on the selected hosts. This action prepares the software for installation on the selected hosts.

Procedure

1. Click the **Submit** button.

   The **Worker(s) Status** table displays the following information for each host that you are adding to the deployment:

   - **IP address:** IP address and (if available), host name of the Kubernetes Worker host.
   - **Status:** Current status of the Kubernetes Worker host, which updates as the installation progresses. Example statuses:
     - **Connecting:** HPE Ezmeral Runtime Enterprise is attempting to connect to the listed Kubernetes Worker hosts.
     - **Running bundle:** HPE Ezmeral Runtime Enterprise has successfully connected to the listed Kubernetes Worker hosts and is preparing the hosts.
     - **Phase 1 of 2 completed:** HPE Ezmeral Runtime Enterprise has completed preparing the listed Kubernetes Worker hosts, which are ready to be added. If you added the hosts by mistake, you may remove them by clicking the **Delete** icon (trash can).
     - **Unlicensed:** If adding the hosts would cause the total number of CPU cores to exceed the number of cores allowed by your license, then this status will appear in an orange bar, and you will not be able to continue installing the host. To resolve this issue, either add a new license that allows the increased number of CPU cores (see License Tab on page 798), or delete the hosts you are trying to add.

   If the host status displays an error, see Troubleshooting Kubernetes Host Installation on page 543

2. After HPE Ezmeral Runtime Enterprise finishes preparing the hosts and before finalizing installation, you can specify the use of each host's drives, or abort the installation on a host.

   Actions:

   - Clicking the **Edit Disks** icon (pencil) for a host opens the **Edit Disk Allocation** popup for that host, which enables you to select one or more drives to add for local persistent and/or ephemeral storage. See Kubernetes Host: Select Hard Drives on page 540.
   - To remove the hosts, click the **Delete** icon (trash can).
   - To view the log from running the installation bundle on the host, click the **Setup Log** icon (down arrow) for that host.

3. Proceed to Kubernetes Host: Select Hard Drives on page 540.

Kubernetes Host: Select Hard Drives

Prerequisites

You have prepared the hosts as described in Kubernetes Host: Add the Hosts on page 539.

About this task

This first phase of Kubernetes Worker host installation prepared the host with the hypervisor agent and Docker. The second phase of the host addition installs the HPE Ezmeral Data Fabric filesystem as a container running the MFS service using the persistent disks. Other system services, such as Nagios and monitoring containers, are installed.

Before proceeding to this second phase, you must allocate the disks on the host for persistent and ephemeral storage.

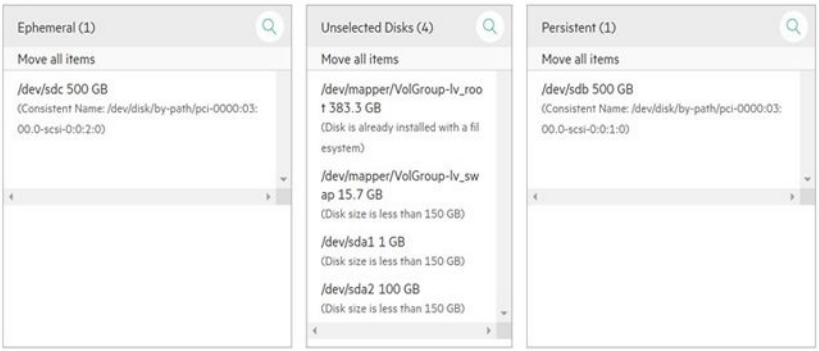
By default, the disks on a Kubernetes hosts are equally split between ephemeral storage and persistent storage. If the host has an odd number of disks, then the ephemeral storage receives the higher number of discs. For example, if the Kubernetes Worker host has five disks, then ephemeral storage receives three disks and persistent storage receives two by default.

If you do not want to install the HPE Ezmeral Data Fabric on Kubernetes-based persistent storage, then you can leave disks in the **Unselected Disks** section. The key use case for this scenario is if you want to install either a third party software-defined storage product or a native Kubernetes orchestrated storage product after creating a Kubernetes cluster.

Procedure

1. Click the **Edit Disks** icon (pencil) for the host you want to edit in the **Kubernetes Host(s) Status** table of the **Kubernetes Hosts Installation** screen.

   The **Edit Disk Allocation** popup for that Kubernetes Worker appears, displaying the suggested disk allocation.



2. Hover the mouse over the disk you want to select.

   One or more arrows appear, depending on whether or not the selected disk is in the **Unselected Disks** area or the **Ephemeral**/**Persistent** areas.

   NOTE: The **Persistent** area is only available if the host has a `Datafabric` tag.

3. Move the mouse to the desired arrow, and then click that arrow.

   The selected drive moves one space in the indicated direction. For example, clicking the left arrow of an unselected disk moves that disk to the **Ephemeral** column.

4. Use the **Posix Client Type** pull-down menu to select the Posix type to use. The available options are **Basic** or **Platinum**.

5. Click the **Set Disk** button to close the popup and add the selected drives.

6. Proceed to Kubernetes Host: Enter Lockdown Mode on page 542.

Kubernetes Host: Enter Lockdown Mode

Prerequisites

You have added the Kubernetes Worker hosts as described in Kubernetes Host Step 4: Select Hard Drives.

Procedure

1. Open the **Quick Access** menu and then select **Enter system lockdown**.

   The **Enter system lockdown mode** dialog appears.

2. Enter a descriptive reason for the lockdown in the **Enter Reason** field.
3. Click the **Submit** button to enter Lockdown mode.
4. Proceed to Kubernetes Host Step 6: Add the Host(s) as Workers.

Kubernetes Host: Add the Hosts as Workers

Prerequisites

Once one or more Kubernetes Worker hosts have been prepared for installation and Lockdown mode is enabled as described in Kubernetes Host: Enter Lockdown Mode on page 542.

Procedure

1. Verify the host fingerprint (MD5 hash). See Public Key Infrastructure on page 134 for information about the PKI.
2. Select the hosts to install in the **Kubernetes Worker(s) Status** table, and then click the **Install** button.

   A confirmation dialog appears.
3. Click **OK** to proceed.

   The status of the selected hosts changes to **storage configuring** and then the **Installing** bar appear in the **Kubernetes Hosts(s) Status** table for the selected Worker hosts while HPE Ezmeral Runtime Enterprise finishes adding them to the deployment.

   This status changes to **configured** after the addition is final.

   If the host status displays an error, see Troubleshooting Kubernetes Host Installation on page 543
4. When the addition completes, exit Lockdown mode by opening the **Quick Access** menu and then selecting **Exit system lockdown**.

   The newly added Kubernetes Worker hosts will appear in the **Kubernetes Hosts Installation** screen. See The Kubernetes Hosts Installation Screen on page 551.
5. Validate that the new Compute hosts have been successfully added, as described in Kubernetes Host: Validate the Worker Installation on page 543.

Kubernetes Host: Validate the Worker Installation

Prerequisites

You have added the hosts as workers as described in Kubernetes Host: Add the Hosts as Workers on page 542.

Perform this procedure for each Kubernetes cluster you created.

About this task

Use the following procedure to validate each Kubernetes cluster you created.

Procedure

1. Access the **Services** tab of the Kubernetes Administrator **Dashboard** screen (see Services Tab) and verify that the new Kubernetes Worker hosts appear and that all services are green.

   Hosts that are in a **ready** state and are not included in an existing Kubernetes cluster should have both the **BD Agent** and **Monitoring Collector** active. If the host has persistent disks, then the **HPE Ezmeral Data Fabric** services must also be enabled and active.

2. If this host contains GPU devices, verify that those devices are available. See Using GPUs in Kubernetes Pods.

3. Create and test a Kubernetes workflow as described in Getting Started with General Kubernetes Functionality and Kubernetes Cluster Usage Examples.

4. Verify that the sample applications are generating the correct output.

5. Access the **Service Endpoints** screen for each cluster you created.

6. In the **Services** column, note the list of services (such as **Hive thrift server**, **Mysql server**, **Spark master**, and/or **SSH**) and the `hostname:port` combination listed for each service.

7. Attempt to access each of the services using the `hostname:port` combination provided.

Troubleshooting Kubernetes Host Installation

Kubernetes Host Installation Fails: Prechecks Errors

To determine which precheck error occurred, do the following:

1. Use SSH to access the host that has the prechecks error.

2. Run the following command:

   ```
   ls -laht /tmp | grep prechecks
   ```

3. Find the most recent prechecks log. The name of the file is in the following format:

   ```
   bd_prechecks.{number}.log
   ```

4. Read through the file to find the error. Prechecks errors have the following format:

   ```
   Checking {description_of_test}: FAILED
   ```

If the error is not one of the following errors, contact Hewlett Packard Enterprise Technical Support.

Error: **`Falco kernel module schema_version must be in the 2.x.y range`**

Symptom	The Kubernetes host installation fails and the following prechecks error is returned: `Falco kernel module schema_version must be in the 2.x.y range`
Cause	The version of the Falco Kernel Module is out of date and does not contain the correct version of the schema.
Action	Update the Falco Kernel Module on this host to version 0.32 or later. Hewlett Packard Enterprise recommends using the latest released version. You must use the Linux `modprobe` tool the Linux `modprobe` tool to install the module.

Error: **`Falco kernel module must be loaded with modprobe to be allowed in ERE`**

Symptom	The Kubernetes host installation fails and the following precheck error is returned: `Falco kernel module must be loaded with modprobe to be allowed in ERE`
Cause	The Falco Kernel Module was installed using a method other than using the Linux `modprobe` tool.
Action	Delete the Falco Kernel Module, then use the Linux `modprobe` tool to reinstall the module. For information about `modprobe`, see the `modprobe(8)` manpage.

Kubernetes Host Installation Fails: Security Error

Symptom	The Kubernetes host installation fails and a security error is returned.
Cause	The local times on the Controller host and the Kubernetes hosts differ significantly.
Action	Set the local time on the Kubernetes host to match the local time on the Controller host. Then begin the installation process again.

Kubernetes Host Installation Fails: Storage Error

Symptom	This environment is an air-gapped environment using a secure container registry, and when you attempt to add a Kubernetes host, the following occurs: • A storage error is returned. • An error message similar to the following is added to the `bds-mgmt.log`: `dictionary update sequence element #3 has length 4; 2 is required`
Cause	The container client certificate uses an RSA key length other than 4086 bits.

Action	Ensure that the certificate key uses RSA 4086. Upload a certificate that uses the correct RSA key length, and then begin the installation again. For more information about the client certificate, see Air Gap Tab on page 799.

Logs for Troubleshooting Kubernetes Hosts

If you experience other issues when installing a Kubernetes host, then access the following logs:

- **Controller host:**
  - **Host Installer log:**

    ```
    /var/log/bluedata/install/addworker.out_.log
    ```

  - **Xtrace file:** This file is a verbose, line-by-line description of the exact commands used by the script to both get data and determine the outcome of each test.

    ```
    /var/log/bluedata/addworker/install.out_.log.xtrace
    ```

- **Kubernetes host:**
  - **Host setup log:**

    ```
    /var/log/bluedata/install/worker_setup_<timestamp>
    ```

  - **Host Xtrace file:** This file is a verbose, line-by-line description of the exact commands used by the script to both get data and determine the outcome of each test.

    ```
    /var/log/bluedata/install/worker_setup_<timestamp>.xtrace
    ```

Using Logs to Troubleshoot Issues

General steps:

1. Begin reading the logs from top to bottom.
2. Stop at the first ERROR that you find. This first error can often cause additional problems downstream. Taking a start-to-finish approach (instead of working your way back from the tail end of the log file) can help you solve one error that in turn resolves a series of cascading errors. If the problem is obvious, then correct the problem and retry the installation.
3. If you are unable to resolve the problems on your own, then contact Hewlett Packard Enterprise for support. You might be asked to provide these installer logs and xtrace files.

Kuberentes Host Tags

The topics in this section describe information and tasks related to Kubernetes Host tags on HPE Ezmeral Runtime Enterprise.

About Tags

Tags are a way to identify hosts with labels that enable HPE Ezmeral Runtime Enterprise features.

This article uses the term **host** to denote Kubernetes hosts and Data Fabric cluster nodes, except if noted.

For reference information about host tags, see Default Host Tags on page 546.

General information about tags:

- There are two main types of tags: default host tags and user-defined tags. Default host tags are included in HPE Ezmeral Runtime Enterprise and are described in Default Host Tags on page 546. You do not have to create default host tags because those tags already exist in the configuration. However, user-defined tags require you to define both the name of the tag and valid values. See Adding a New Tag on page 548.
- You assign tags to a host either when the host is added to HPE Ezmeral Runtime Enterprise, or by updating tags for the existing host.

Working with Tags

This section outlines the general process of creating and using tags. You do not have to create default host tags because those tags already exist in the configuration. The process is as follows:

1. Create a tag using the **Tags** screen, as described in The Tags Screen and Adding a New Tag; tags may also be updated (see Updating Tags for a Host on page 550). Tag names are arbitrary and may include special characters or spaces.
2. Associate the tags with one or more hosts, and then specify the tag values for each host, as described in Assigning Tags to a Host. Values are arbitrary; however, the best practice is to assign consistent values to each tag. For example, you could use the `cpu` tag to:
   - Specify whether a host has high-performance CPUs installed by assigning `yes` and `no` values. In this case, you may not want to assign values regarding a specific CPU type.
   - Specify the specific type of CPU installed by assigning values such as `xeon8180` or `ryzen1950x`. In this case, you may not want to assign values regarding whether a high-performance CPU is installed.
   - You may also assign unique values to each host. For example, you could assign the value `xeon_1` to the `cpu` tag for Host_A, then assign the value `xeon_2` to Host_B, `xeon_3` for Host_C, and so on.

Default Host Tags

This page lists and describes the default system host tags in the HPE Ezmeral Runtime Enterprise. Default host tags are tags that are available by default when HPE Ezmeral Runtime Enterprise is deployed. Some of these host tags enable functions or features or affect how the host can be used. Other default host tags are informational and can be used for things like creating tenant or project constraints.

This article uses the term **host** to denote Kubernetes hosts and Data Fabric cluster nodes, except if noted.

The following is a list of default host tags, including information on allowed tag values, host conditions, and host restrictions:

Datacenter

The `datacenter` tag indicates that the host or node is in a certain datacenter. You can apply the `datacenter` tag with different values to Compute hosts or Data Fabric nodes, and the HPE Ezmeral Runtime Enterprise will attempt to place resources, such as clusters, across various data centers for added redundancy.

- Allowed tag values: This tag must be 63 characters or less. It must be empty or begin and end with an alphanumeric character ([a-z0-9A-Z]) with dashes (-), underscores (_), dots (.), and alphanumerics between.
- Host state conditions or restrictions: none

Datafabric

When set to `yes`, `YES`, `true`, or `TRUE`, the `datafabric` tag denotes that the host is to be used for storage for storage in an **HPE Ezmeral Data Fabric on Kubernetes** cluster.

- Allowed tag values: `true`, `TRUE`, `yes`, `YES`.
- Host state conditions or restrictions:
  - This tag can only be set when the host is added to the platform initially.
  - This tag cannot be deleted or changed.

Istio-ingressgateway

Hosts with the `istio-ingressgateway` tag set to `true` will function as Istio Ingress Gateway in Kubernetes clusters that have Istio enabled. See Creating a New Kubernetes Cluster, Creating a New Data Fabric Cluster, and Istio.

- Allowed tag values: `true`, `false`.
- Host state conditions or restrictions: this tag cannot be set, updated, or deleted when the host is part of a cluster.

Rack

The `rack` tag indicates that the host or node is in a certain rack.

- Allowed tag values: This tag must be 63 characters or less. It must be empty or begin and end with an alphanumeric character ([a-z0-9A-Z]) with dashes (-), underscores (_), dots (.), and alphanumerics between.
- Host state conditions or restrictions: none.

Falco

When set to `true`, the `falco` tag indicates that the Falco Container Security detection service is enabled for this Kubernetes node.

- Allowed tag values: `true`, `false`
- Host state conditions or restrictions:
  - The `falco` tag is currently not enforced on 5.4.x
  - The user may install/uninstall the Falco driver at any host state manually.
  - This tag is set with a value of `true` by bd_mgmt when the Falco driver module is loaded and detected on the host during host install.
  - This tag can only be set/changed after a host is added (phase 1 host installation).
  - This tag can only be set to `true` if the Falco driver module is loaded on the host.
  - This tag can be deleted or set to `false` at any host state when the host is not in the process of being configured.

Related tasks

Assigning Tags to a Host on page 549
Describes adding host tags when adding a host to HPE Ezmeral Runtime Enterprise.

More information

The Tags Screen on page 548

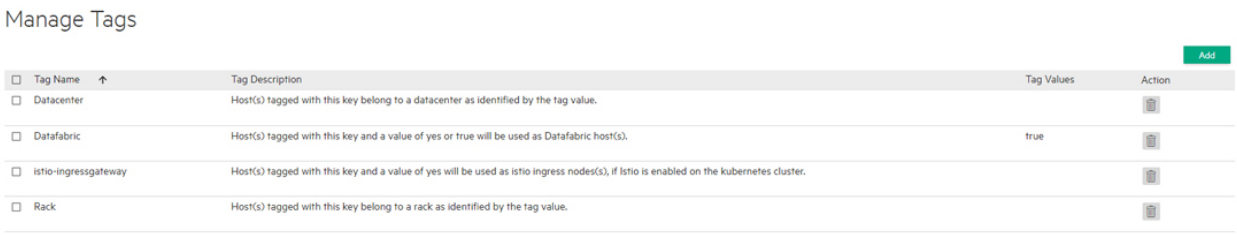
The **Tags** screen enables you to see the tags that are available for labeling hosts in this deployment. You can add or delete tags from the **Tags** screen.

The Tags Screen

The **Tags** screen enables you to see the tags that are available for labeling hosts in this deployment. You can add or delete tags from the **Tags** screen.

This article uses the term **host** to denote Kubernetes hosts and Data Fabric cluster nodes, except if noted.

In either the **Kubernetes Hosts Installation** screen or the **Hosts for High Availability** screen, clicking the **Manage Tags** button opens the **Tags** screen.



The top of this screen contains the following buttons:

Add	Clicking this button opens the **Create Tag** dialog. See Adding a New Tag.
Delete	Selecting one or more tags and then clicking this button deletes the selected tags. See Deleting a Tag.

The table on this screen contains the following information and functions:

Tag Name	The name of the tag.
Description	The description that was provided when the tag was created.
Tag Values	All of the values that have been assigned to this tag. Only one value can be applied per tag per host. For example, if you create a tag called `sample_tag`, then you cannot assign both `yes` and `no` values to a single host.
Delete	Clicking the **Delete** icon (trash can) in the **Action** column deletes the selected tag. **CAUTION:** You cannot undelete a tag.

For information about the default host tags included in HPE Ezmeral Runtime Enterprise, see Default Host Tags on page 546.

Adding a New Tag

Describes adding a new custom host tag to HPE Ezmeral Runtime Enterprise.

About this task

This process creates a new tag but does not assign that tag to a host, nor does it create any values for the tag. To assign a tag and a value to a host, see Assigning Tags to a Host.

This article uses the term **host** to denote Kubernetes hosts and Data Fabric cluster nodes, except if noted.

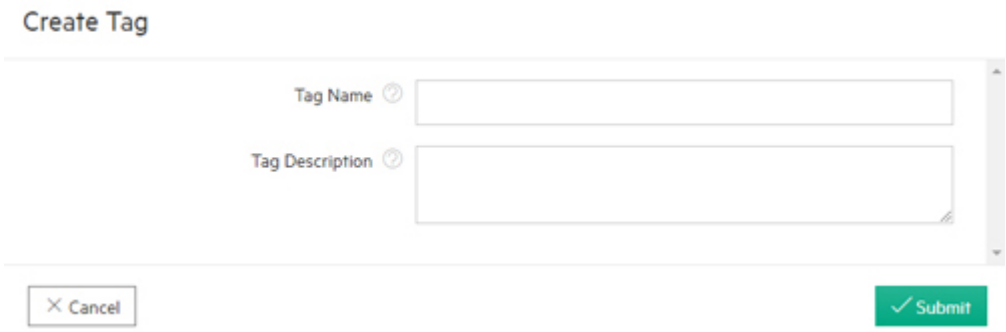
Procedure

1. In the **Installation** screen, click the **Manage Tags** button in the upper right corner.

   The **Tags** screen appears.

2. Click the **Add** button.

   The **Create Tag** dialog appears.



3. Enter a name for the tag in the **Tag Name** field.
4. Enter a description for the tag in the **Tag Description** field.
5. Click **Submit** to save your changes and return to the **Tags** screen, which will display the newly-added tag.

Deleting a Tag

This article uses the term **host** to denote Kubernetes hosts and Data Fabric cluster nodes, except if noted.

To delete a tag, you may either:

- Click the **Delete** icon (trash can) for the tag you want to delete.
- Select one or more tags and then click the **Delete** button.

CAUTION: You cannot undelete a tag.

A confirmation dialog appears. Click **OK** to delete the tag.

Assigning Tags to a Host
Describes adding host tags when adding a host to HPE Ezmeral Runtime Enterprise.

About this task

This procedure describes assigning one or more tags to one or more hosts while you are adding the hosts.

To assign tags to an existing host, see Updating Tags for a Host on page 550.

This article uses the term **host** to denote Kubernetes hosts and Data Fabric cluster nodes, except if noted.

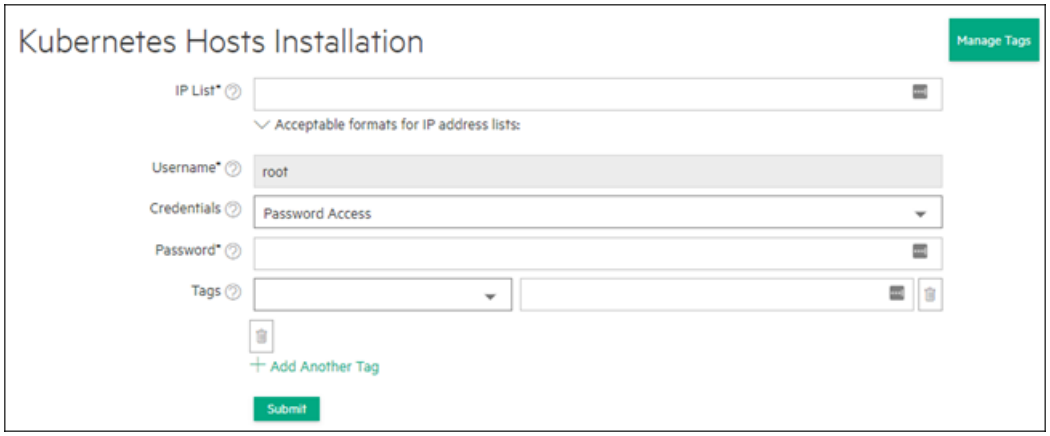
Procedure

1. When selecting the hosts to add, on the host **Installation** screen, click the **Add Tag** link underneath the IP list field.

   If the IP list contains multiple IP addresses, the tags you add are applied to all the hosts in the list.

A row appears with a pull-down menu and a text field. The pull-down menu lists one of the tags that you added in Adding a New Tag on page 548 or one of the default host tags.

If you selected this option by mistake, you may click the **Delete** icon (trash can) to remove the row.



2. Use the pull-down menu to select the desired tag, and then enter a value for the tag in the empty text field.

   The tag name must exist in the configuration. The tag value for tags you value can be any value you like, however you should decide on a uniform set of values (such as `yes` or `no`, a specific CPU type, etc.) for consistency. Default system tags might have rules for values or or a defined set of values.

   NOTE: You may only assign one value per tag per host. For example, you cannot assign the values `yes` and `no` to the tag `sample_tag`.

3. To assign an additional tag to the hosts in the IP list, click the **Add Another Tag** link and repeat Step 2 for the new row that appears.

   To add a new tag to the configuration, click **Manage Tags**.

Updating Tags for a Host
Describes how to change the host tags and values assigned to a host that has already been installed in a HPE Ezmeral Runtime Enterprise deployment.

About this task

NOTE: You cannot use the web interface to update tags for hosts from imported external Kubernetes clusters.

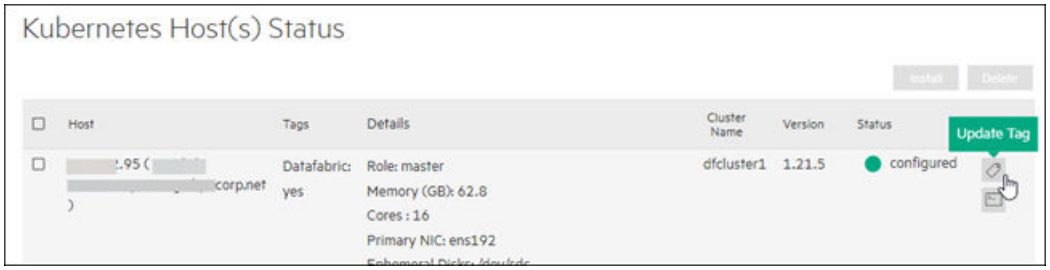
NOTE: Please refer to Default Host Tags on page 546 for information about deleting, adding, or updating default host tags.

You may change the tags and/or values assigned to a host at any time. For example, you may be upgrading from a previous version of HPE Ezmeral Runtime Enterprise, or may decide to implement or modify tags.

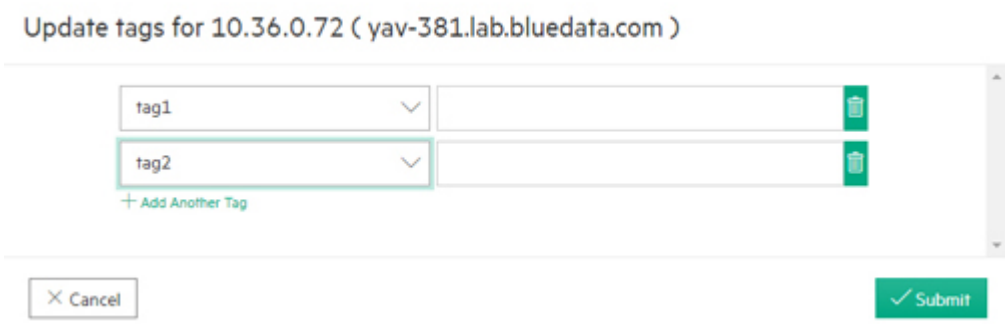
This article uses the term **host** to denote Kubernetes hosts and Data Fabric cluster nodes, except if noted.

Procedure

1. In the **Installation** screen, scroll down to the **Worker(s) Status** table, and then click the **Update Tags** icon (tag) for the desired host.



The **Update Tags** popup appears.



2. Use the pull-down menus and fields to select tags and then enter values.
   - Remove a tag/value by clicking the **Delete** icon (trash can) for that tag.
   - Add a new tag/value by clicking the **Add Another Tag** link.
   - Modify the value for a tag by entering the new value in the text field.

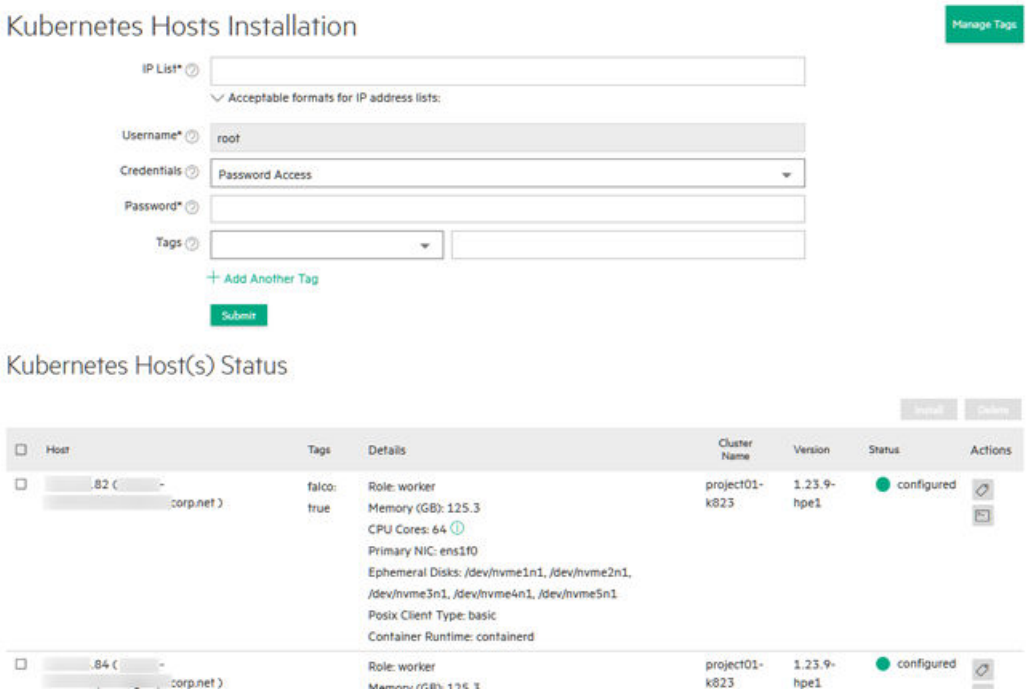
   NOTE: You may only assign one value per tag per host. For example, you cannot assign the values `yes` and `no` to the tag `sample_tag`.

3. Click **Submit** to save your changes and exit the popup.

What to do next

The Kubernetes Hosts Installation Screen

Selecting **Hosts** in the main menu opens the **Kubernetes Hosts Installation** screen, which lists the Kubernetes hosts and nodes and enables you to install, edit, and remove Kubernetes hosts/nodes. You can also manage host/node tags.



This screen lists Kubernetes hosts/nodes only.

- For information about Controller, Shadow Controller, or Arbiter hosts, see The Controllers & HA Screen on page 754.
- For information about Gateway hosts, see The Gateway/Load Balancer Screen on page 755 and Gateway Installation Tab on page 755.

This upper portion of this screen contains the following functions:

Manage Tags

Clicking this button opens the **Tags** screen, which allows you to view, add, and delete host/node tags that then be passed down as first-class Kubernetes labels to the underlying Kubernetes hosts/nodes. See The Tags Screen on page 548.

IP List

Enter the IP addresses for one or more Kubernetes hosts or nodes. Hosts run the containers that form clusters. Nodes act as hosts for Data Fabric clusters. See About HPE Ezmeral Data Fabric on Kubernetes on page 324.

Add Tag

Clicking this link allows you to select an existing tag (see Assigning Tags to a Host on page 549) for the Kubernetes hosts/nodes that you are installing and specify a value.

For example, if you create a tag called `ram` and you are installing a host/node with a large amount of RAM installed, then you could add the tag `ram` to this Kubernetes host/node and then enter a value of `high`. If various Kubernetes hosts/nodes reside in different racks, then you could use a tag called `rack` to specify the Kubernetes host/node location, such as `rack_a`, `rack_b`, or `rack_c`.

To select a tag, use the pull-down menu to select the tag to add, and then type the desired value in the text field. If you add a tag by mistake, click the **Delete** icon

(trash can) to remove the tag. You can also add one or more additional tags by clicking the **Add Another Tag** link and repeating this process for each tag you want to assign to the Kubernetes hosts/nodes. You may only assign one value per tag.

NOTE:

If you are adding a Data Fabric node, then:

- You must select the `Datafabric` tag and then set the value to `yes` . See Kubernetes Data Fabric Node Installation Overview on page 531.
- Do not select this tag if you are adding one or more Kubernetes Worker hosts.
- You must install Data Fabric nodes separately from Kubernetes Worker hosts.

Credentials

This is where you add either a valid user name and password or SSH key to access the Kubernetes hosts/nodes being added.

Submit

Clicking this button begins the process of adding the specified Kubernetes hosts/nodes. Some features require preparation of the hosts before the hosts are added to HPE Ezmeral Runtime Enterprise.

See Kubernetes Worker Installation Overview on page 528 or Kubernetes Data Fabric Node Installation Overview on page 531, as appropriate.

The lower portion of this screen contains the **Install** and **Delete** buttons, and the **Workers Status** table.

Install

Selecting one or more hosts/nodes in the following table and then clicking this button installs the selected hosts/nodes as Kubernetes Worker hosts or Data Fabric nodes, if they have not already been installed. See Kubernetes Worker Installation Overview on page 528 or Kubernetes Data Fabric Node Installation Overview on page 531, as appropriate.

Delete

Selecting one or more Kubernetes hosts/nodes in the following table and then clicking this button removes the selected Kubernetes hosts. See Decommissioning/Deleting a Kubernetes Host on page 555.

Workers Status

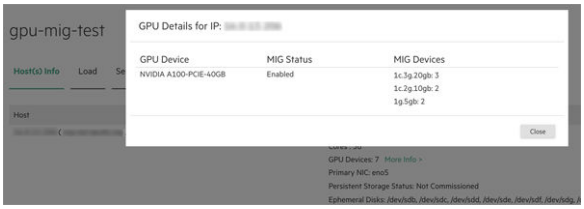
The table displays the following information and functions for each Kubernetes host/node:

- **Host:** IP address and hostname of the Kubernetes host/node.
- **Tags:** Lists any tags assigned to the Kubernetes host/node and the value assigned to each tag.
- **Details:** Lists information about the host, such as the CPU cores, number of GPU devices, RAM, primary NIC, persistent storage status, the paths to the ephemeral and persistent storage, the Posix client type, and the container runtime. To display the number of logical CPU cores, physical CPU

cores and sockets, hover over the information icon next to the **CPU Cores** entry.

If the host is running the Hewlett Packard Enterprise distribution of Kubernetes, the container runtime is `containerd`. If the host is part of a Kubernetes cluster that was created on a previous version of HPE Ezmeral Runtime Enterprise and has not been migrated to use the Hewlett Packard Enterprise distribution of Kubernetes, the container runtime is `Docker`.

If a GPU device supports MIG, when you click the **More Info** link, **GPU Details** dialog shows information about the MIG configuration. For example:



If the GPU device does not support MIG, the **GPU Details** dialog lists the GPU devices, but shows `N/A` in **MIG Status** and in **MIG Devices**.

- **Cluster Name:** Name of the Kubernetes cluster to which this host/node is attached. The message **Not Assigned** appears if the host/node is not assigned to a Kubernetes cluster.

- **Version:** Kubernetes version assigned to this Kubernetes host/node. The message **Not Assigned** appears if the host/node is not assigned to a Kubernetes version.

  If the Kubernetes version number includes the phrase `-hpe<n>`, where <n> is a number, the host is running a Kubernetes version that is distributed by Hewlett Packard Enterprise, which uses the containerd runtime.

- **Status:** Status of the host/node. This column will say **ready** for all fully-installed Kubernetes hosts/nodes. See Kubernetes Host: Add the Hosts on page 539 and Kubernetes Host: Add the Hosts as Workers on page 542 for the statuses that appear during the Kubernetes host/node installation process.

- **Actions:**

  - **Decommission:** Clicking the **Decommission** icon (broken line) decommissions the Kubernetes host/node and prevents it from running clusters or pods. See Decommissioning/Deleting a Kubernetes Host on page 555.

- **Delete:** Clicking the **Delete** icon (trash can) for a decommissioned Worker host removes that host. See Decommissioning/Deleting a Kubernetes Host on page 555.
- **Update Tags:** Clicking the **Update Tags** icon (tag) for a host opens the **Update Tags** popup, which enables you to add, edit, and remove tags for that host. See Updating Tags for a Host on page 550.

Decommissioning/Deleting a Kubernetes Host

NOTE: This article uses the term host to denote both Kubernetes hosts and Data Fabric cluster nodes, except if noted.

This article describes the following:

- Decommissioning a Kubernetes Worker Host.
- Deleting a Kubernetes Worker Host.

Decommissioning a Kubernetes Worker Host

Clicking the **Decommission** icon (barred circle) in the **Actions** column of the **Kubernetes Host(s) Status** table in the **Kubernetes Hosts Installation** screen (see The Kubernetes Hosts Installation Screen on page 551) removes the persistent storage service and vacates any data volumes to other nodes so that the host can be removed. You may then delete the Kubernetes host, as described in Deleting a Kubernetes Worker Host on page 555, below.

NOTE: You can only decommission one Kubernetes Worker host at a time, and the deployment cannot have any fewer than four (4) commissioned Kubernetes Worker hosts.

Deleting a Kubernetes Worker Host

Deleting a Kubernetes host completely removes it from the deployment. To delete one or more Kubernetes Worker hosts:

1. Remove the affected Kubernetes Worker hosts from any existing Kubernetes clusters.
2. Access the **Kubernetes Hosts Installation** screen (see The Kubernetes Hosts Installation Screen on page 551).
3. Decommission the Kubernetes Worker hosts by clicking the **Decommission** icon (broken line) for each Kubernetes Worker host being deleted.
4. In the **Kubernetes Host(s) Status** table, either:
   - Remove a single Kubernetes Worker host by clicking the **Delete** icon (trash can) for the host you want to remove.
   - Remove multiple Kubernetes Worker hosts by selecting the affected hosts and then clicking the **Delete** button above the table.
5. The **Status** of the affected Kubernetes Worker hosts shows **deleting**.

The selected Worker hosts are removed.

Monitor the deletion process and address any reported problem. The log is located on the affected host, in the `/tmp/worker_setup_<timestamp>` folder.

In the unlikely event that Kubernetes Worker deletion fails, you can delete the Kubernetes Worker host manually by executing the following command on that Worker host:

- If the Kubernetes Worker host was installed without using the agent:

```
/opt/bluedata/bundles/<common-epic.bin> -ef
```

- If the agent was used when installing the Kubernetes Worker host:

```
/opt/bluedata/bundles/<common-epic.bin> -ef --onworker --node-type
worker --worker <worker-ip>
```

After deletion has completed, execute the following commands to verify that HPE Ezmeral Runtime Enterprise has been removed from that host:

- `bdconfig -sysinfo`

  The system should return `command not found`.

- `rpm -qa | grep -E "bluedata|hpe"`

  The system should return an empty response.

If you plan to re-use this host as a new Worker host for either Big Data/AI/ML or Kubernetes, then you will need to ensure that Docker storage is cleaned-up. Follow these steps:

1. Make sure that `/var/lib/docker` is empty.
2. Make sure that `/etc/sysconfig/docker-storage` has the value `DOCKER_STORAGE_OPTIONS=`

To reinstall the host immediately, you will need to begin again from Kubernetes Host: Add the Hosts on page 539. Otherwise, you must return to Kubernetes Worker Installation Overview on page 528 and restart the installation process from the beginning.

Downloading Kubernetes Usage Details

Platform Administrators can download scripts to view Kubernetes usage details in HPE Ezmeral Runtime Enterprise.

The following two scripts allow Platform Administrators to download Kubernetes usage details:

- **k8susage.py:** Collects the usage metrics from Kubernetes clusters and stores the results in a comma-delimited (.csv) file. Historical usages are collected for the specified time period and aggregated over the specified time interval (aggregation interval) or the current usage (over a 2 minute interval) is collected. Individual pod metrics and pod counts are summed up for each aggregation interval following the hierarchy Pod > Node > Namespace > All namespaces. This script collects the following metrics:
  - CPU cores limits and requests
  - Memory limits and requests
  - Ephemeral storage capacity and usage
  - Number of running and pending pods
  - Tenant storage usage (now option only)
- **k8scsv.py:** Collect utilization metrics from Kubernetes clusters and stores the results in a comma-delimited (.csv) file. Historical usages are collected for the specified time period and aggregated

over the specified time interval (aggregation interval) or the current usage (over a 2-minute interval) is collected. Individual pod metrics, except the CPU and memory limit percentages are summed up for each aggregation interval following the hierarchy Pod > Node > Namespace > All Namespaces. For the CPU and memory limit metrics, the per aggregation interval averages are computed for the Node, Namespace, and All Namespaces. This script collects the following metrics:

- CPU
  - Pod used Nanocores
  - Pod usage as a percentage of the total node CPU
  - Pod usage as a percentage of the defined limit for the pod containers (or total node CPU if unlimited)
- Memory
  - Pod total memory usage
  - Pod memory usage as a percentage of the total node allocable memory
  - Pod memory usage as a percentage of the defined limit for the pod containers (or total node allocable memory if unlimited)
  - Pod total network received (Rx) and transmitted (Tx) bytes.
  - Pod network received (Rx) and transmitted (Tx) bytes per aggregation interval (not for now option)

Location

These scripts are located on the Controller host, in the following directory:

```
/opt/bluedata/common-install/scripts/monitoring
```

Usage

You must have Platform Administrator privileges to execute these scripts.

- **k8susage.py:**

  Either:

  ```
  python k8susage.py -c <controller> -f <credentials_file> -s
  <start_time> -e <end_time>
  ```

  or

  ```
  python k8susage.py -c <controller> -f <credentials_file> -n
  ```

  For example:

  ```
  python k8susage.py -f cred.json -c 10.1.32.120 -s 2021/11/01-08:00:00 -e
  2021/11/10-10:11:45
  ```

- **k8scsv.py:**

  Either:

  ```
  python k8scsv.py -c <controller> -f <credentials_file> -s <start_time> -e <end_time>
  ```

  or

  ```
  python k8scsv.py -c <controller> -f <credentials_file> -n
  ```

Where:

- `<controller>` is one of the following:
  - The IP address of the Controller host
  - Cluster IP address (if platform HA is enabled)
  - IP address of a Gateway host.
- `<credentials_file>` is the path to a JSON file that contains the username and password to use to connect to the Controller host.

  This JSON file stores the username under the key `user` and the password under the key `password`.

  For example:

  ```
  { "user": "MyUserName", "password": "MyPassword123" }
  ```

Options

Flag	Description	Required
-h, --help	Show help message and exit.	
-a, --all-namespaces	Include all namespaces and not just tenant associated namespaces.	
-c CONTROLLER, --controller=CONTROLLER	Controller IP address.	Yes
-e END, --end=END	End time as either YYYY/mm/dd-HH:MM:SS or number of hours in the past.	Yes, with the -s option.
-f CREDFILE, --file=CREDFILE	Credentials file.	Yes
-i INTERVAL, --interval=INTERVAL	Interval (e.g. 10m for 10 minutes, 1h for 1 hour, etc., min: 2m).	
-k K8SCLUSTER, --k8scluster=K8SCLUSTER	Filter by this Kubernetes cluster name.	
-l LOGFILE, --logfile=LOGFILE	Log file (default is ./script_name.log)	
-n, --now	Collect latest usage over a 2-minute interval.	Yes, without the -s -e options.
-o OUTPUTFILE, --outputfile=OUTPUTFILE	Output file (default is ./script_name.csv).	

Flag	Description	Required
-p, --list-pods	List pods in the .csv file.	
-q, --quiet	No output to console.	
-s START, --start=START	Start time as either YYYY/mm/dd-HH:MM:SS or number of hours in the past.	Yes, with the -e option.
-t TENANT, --tenant=TENANT	Filter by this tenant name.	
-x, --https	Connect to the Controller host via https.	

CSV File Columns

The .csv files include the following columns:

- **k8susage.csv:**
  - Time period query:

    ```
    Time Window Start | Time Window End | Selector | Cluster Name | Tenant
    Name | Namespace | Node Name | Pod Name | CPU Limit Cores | CPU Request
    Cores | Memory Limit (B) | Memory Request (B) | Ephemeral Storage
    Capacity (B) | Ephemeral Storage Available (B) | Ephemeral Storage Used
    (B) | Running Pods | Pending Pods
    ```

  - Now query:

    ```
    Timestamp | Selector | Cluster Name | Tenant Name | Namespace | Node
    Name | Pod Name | CPU Limit Cores | CPU Request Cores | Memory Limit
    (B) | Memory Request (B) | Ephemeral Storage Capacity (B) | Ephemeral
    Storage Available (B) | Ephemeral Storage Used (B) | Tenant Storage
    Used (MB) | Running Pods | Pending Pods | Quota CPU Cores | Quota
    Memory (GB) | Quota Ephemeral Storage (GB) | Quota Tenant Storage (GB)
    | Cluster CPU Cores | Cluster Memory (GB) | Cluster Ephemeral Storage
    (GB) | Cluster Tenant Storage (GB)
    ```

- **k8scsv.csv:**
  - Time period query:

    ```
    Time Window Start | Time Window End | Selector | Cluster Name | Tenant
    Name | Namespace | Node Name | Pod Name | CPU Usage Nanocores | CPU
    Usage Node (%) | CPU Usage Limit (%) Memory Usage (B) | Memory Usage
    Node (%) Memory Usage Limit (%) | Network Rx (B) | Network Tx (B) |
    Network Rx Int (B) | Network Tx Int (B)
    ```

  - Now query:

    ```
    Timestamp | Selector | Cluster Name | Tenant Name | Namespace | Node
    Name | Pod Name CPU Usage Nanocores | CPU Usage Node (%) | CPU Usage
    Limit (%) Memory Usage (B) | Memory Usage Node (%) Memory Usage Limit
    (%) | Network Rx (B) | Network Tx (B)
    ```

In the .csv files, The **Selector** column specifies the **Row** data:

- **Total:** Total sum/average for the period.

- **Tenant:** Total sum/average for the tenant/namespace.
- **Node:** Total sum/average for the node.
- **Pod:** Individual pod.

Limitations

- The scripts attempt to match pods in tenant-associated namespaces with the same name on dierent clusters to the correct tenant based on the cluster nodes assigned to the cluster during the aggregation interval. Pending pods that are not assigned to a node cannot be matched.
- The scripts query the change history of the clusters for added/deleted hosts and try to match nodes found in the aggregation data to a specific cluster. Deleted hosts cannot be matched.
- Historical data for deleted tenants/namespaces, nodes and pods can be retrieved using the `--all-namespaces` option, but cannot be matched to clusters or tenants.

Kubernetes Application Administration

The topics in this section describe information and tasks related to Kubernetes application administration on HPE Ezmeral Runtime Enterprise.

Applications Overview

Kubernetes Cluster Member users can launch pods using the **Kubernetes Applications** screen. This screen has three tabs:

- **KubeDirector:** Allows you to launch pods using KubeDirector applications by clicking the **Launch** button for the application you want to deploy. See KubeDirector Tab.
- **Kubectl:** Allows you to onboard (upload) Kubectl applications using a filesystem mount interface. See Kubectl Tab.
- **Service Endpoints:** Allows you to access exposed application endpoints. See Service Endpoints Tab.
- **Virtual Endpoints:** Allows you to access exposed virtual application endpoints. See Virtual Endpoints Tab.

The following articles describe how to deploy and onboard applications:

- **Deploying Applications:** Describes how to deploy KubeDirector applications. See Deploying Applications.
- **Onboarding Applications:** Describes how to onboard Kubectl applications. See Onboarding Applications.

For Additional Information

This includes additional information authoring and deploying custom applications using the open-source BlueK8s project, which includes the KubeDirector operator. (Link downloads a .zip file that extracts to an .md text file.)

The Kubernetes Applications Screen

Selecting **Applications** in the main menu opens the **Kubernetes Applications** screen, which allows you to launch pods and upload/download files to/from the Kubernetes cluster. This screen is not available in HPE Ezmeral Runtime Enterprise Essentials.

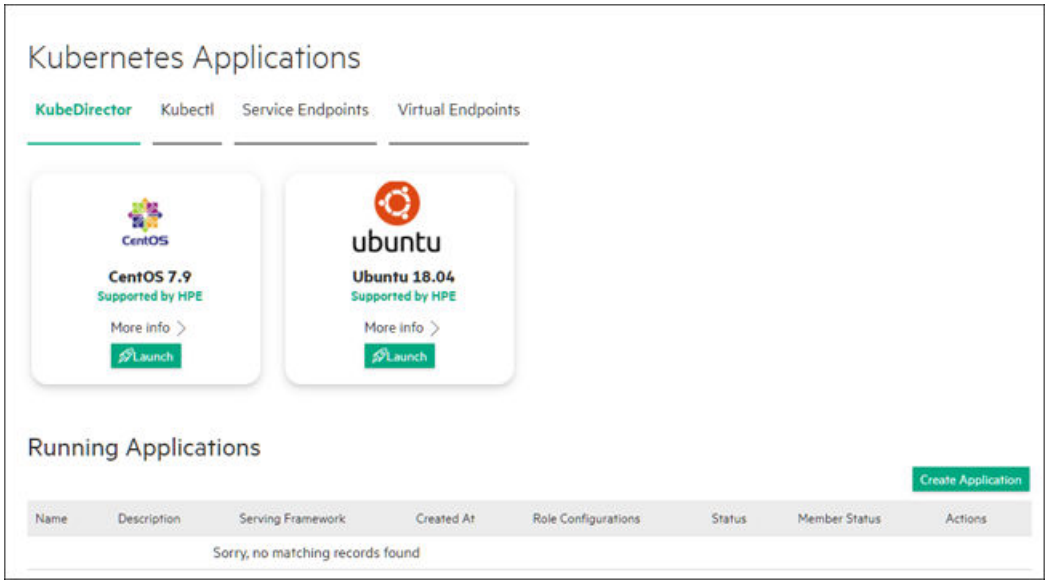
This screen is divided into two tabs:

- **KubeDirector**: Allows you to launch and delete pods. See KubeDirector Tab.

- **Kubectl**: Allows you to upload and download files to/from the cluster. See Kubectl Tab.
- **Service Endpoints**: Allows you to access exposed application endpoints. See Service Endpoints Tab.
- **Virtual Endpoints**: Allows you to access virtual endpoints. See Virtual Endpoints Tab.

KubeDirector Tab

The **KubeDirector** tab presents a list of available **KubeDirector** applications and allows you to launch pods using those applications. You can also delete running pods.



The top of this screen contains one tile for each available KubeDirector application.

The lower portion of this screen contains the **KubeDirector Running Applications** table, which lists the following information for each pod in the current cluster:

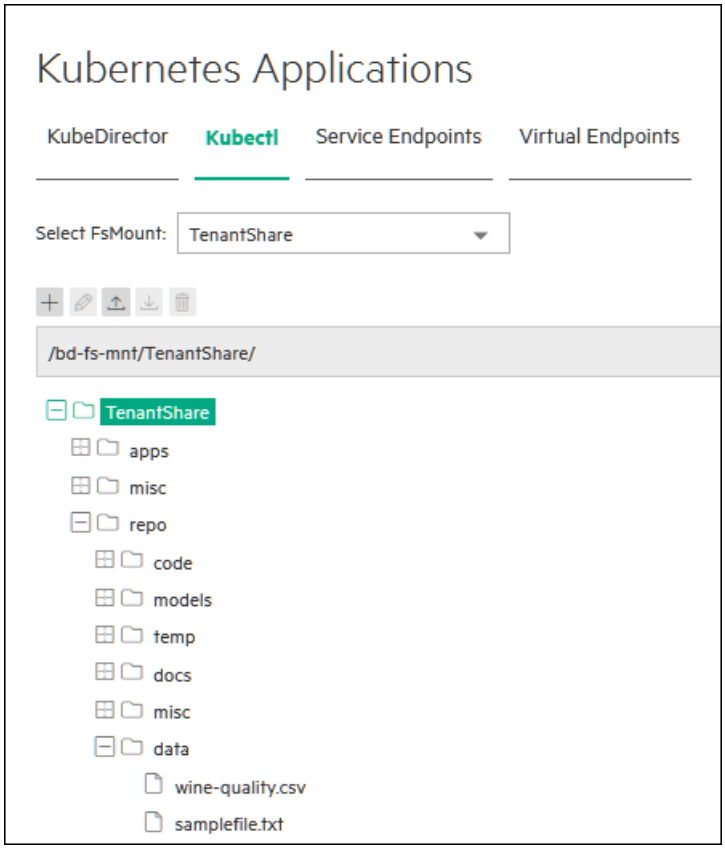
- **Name**: Name of the application/pod.
- **Created at**: Date and time that the application/pod was created.
- **Role Configuration**: Name of each virtual node/container role in the cluster (e.g. controller or worker) and the number of virtual nodes/containers in each role.
- **Status**: Status of the pod.
- **Member Status**: Status of the Kubernetes Tenant Member users who can access this application.
- **Action**: Clicking the **Delete** icon (trash can) deletes the selected application/pod.

Clicking the **Launch** button for an application opens the **Launch Kubernetes Applications** screen. See Deploying Applications.

Kubectl Tab

NOTE: This tab is not available for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

The **Kubectl** tab allows you to onboard and deploy Kubectl applications using any of the FS mounts that have been created for the current cluster.

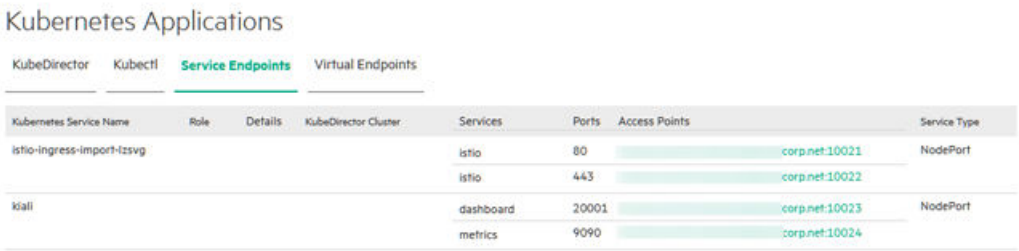


The top of this tab contains the **Select FS Mount** pull-down menu, which allows you to select the FS mount you want to use to access the application. See About FS Mounts.

You can upload and download files as described in Uploading and Downloading Files. See Onboarding Applications for an overview of deploying Kubectl applications.

Service Endpoints Tab

The **Service Endpoints** tab displays the available service endpoints.



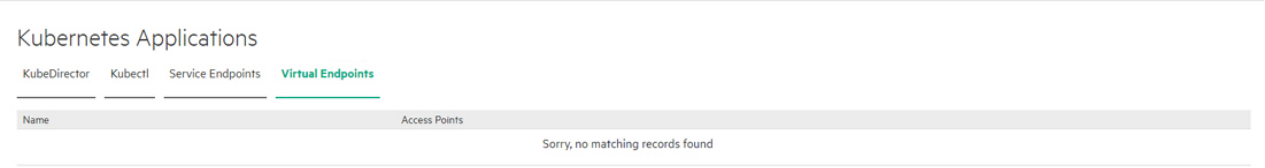
This tab provides the following information for each of the virtual nodes/containers in the current Kubernetes cluster:

- **Name**: Name of the virtual node/container.
- **Details**: Information about the virtual node:
  - **Application Name**: Name of the pod to which the virtual node/container belongs.
  - **Role**: Role of the virtual node/container within its pod, such as **controller** or **worker**.

- **Services**: List of the services that are running on each virtual node/container.
- **Port**: Gateway host port for the service. You can access the service by accessing that port at the IP address of the virtual node/container.
- **Gateway Mappings**: Gateway hostname and port to which each service is mapped. Clicking a link opens the specified service in a new browser tab/window.

Virtual Endpoints Tab

The **Virtual Endpoints** tab displays the available virtual service endpoints.



This tab provides the following information for each of the virtual service endpoints in the current Kubernetes cluster:

- **Name**: Name of the virtual service endpoint.
- **Access Points**: Links to use to access each available virtual service endpoint.

Deploying KubeDirector Applications

Deploy KubeDirector applications into HPE Ezmeral Runtime Enterprise using the **KubeDirector** tab of the **Kubernetes Applications** screen.

This article describes how to deploy KubeDirector applications using the **KubeDirector** tab of the **Kubernetes Applications** screen (see KubeDirector Tab).

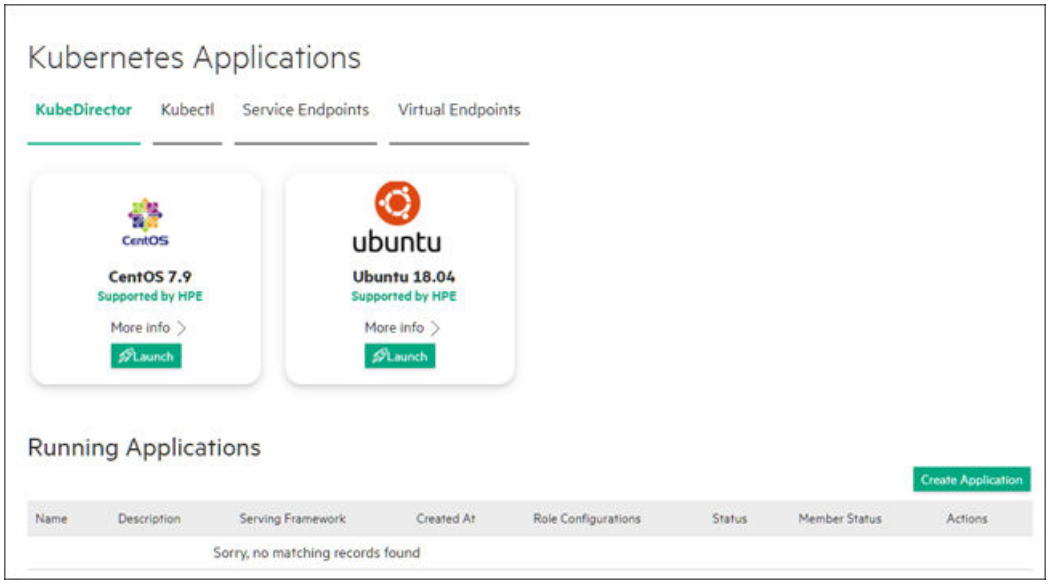
For instructions on onboarding a Kubectl application, see Onboarding Applications.

If your application will use GPU resources, see also Using GPUs in Kubernetes Pods.

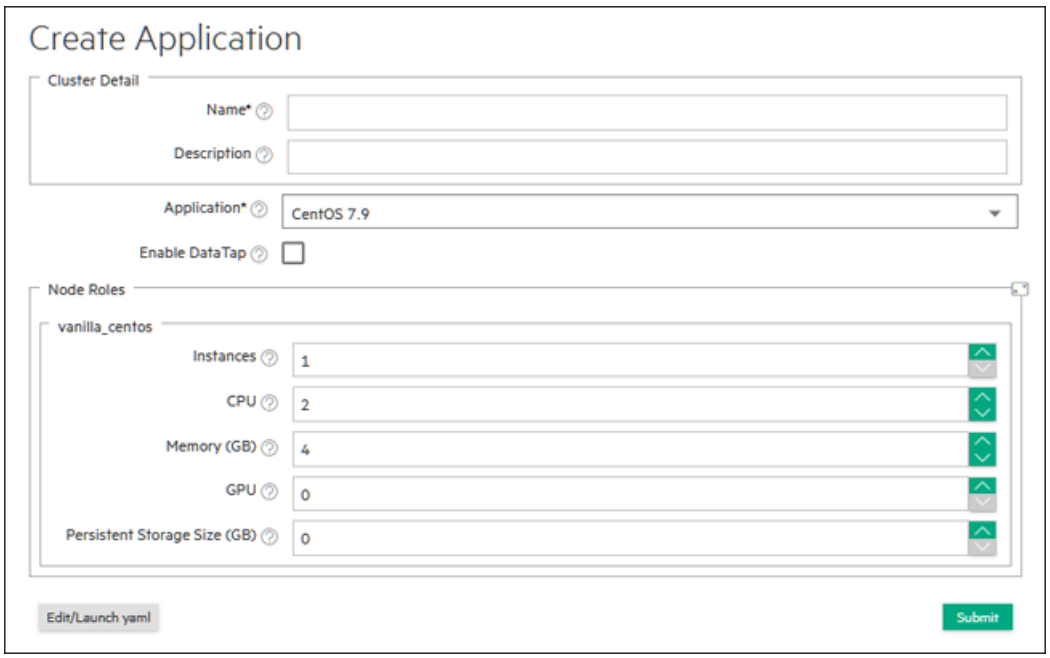
Deploying KubeDirector Applications

To deploy a KubeDirector application:

1. Log into the web interface as a Member of the Kubernetes cluster within which you want to launch the application.
2. In the main menu, select **Applications** to open the **Kubernetes Applications** screen. See The Kubernetes Applications Screen.
3. If needed, select **KubeDirector** to open the **KubeDirector** tab.
4. Find the tile that corresponds to the application you want to deploy, and then click the **Launch** button on that tile.



The **Create Application** screen appears. The content of the screen varies depending on which application you are creating. The following screen is an example of the screen for a CentOS 7.9 application instance:



5. Complete the information on the form.

   To enable DataTap for this application, **Enable DataTap**. For information about DataTap, see About DataTaps on page 122.

6. If needed, open the YAML file for editing by clicking **Edit/Launch yaml**. If you choose to edit the YAML file, you can edit some or all of the following information, as appropriate.

   - **apiVersion:** API to use for the pod that will be created to deploy this application.
   - **Metadata name:** Name of the pod that will be created to deploy this application.
   - **Metadata namespace:** Kubernetes namespace to use. Leave this at the default setting unless you are an advanced Kubernetes user.

NOTE: You can create KubeDirector applications that HPE Ezmeral Runtime Enterprise automatically applies to future tenants.

If you create a KubeDirector application in one of the following namespaces, HPE Ezmeral Runtime Enterprise adds the application to the list of KubeDirector applications used in the creation of future tenants:

- If your platform has a Spark or ML Ops license, KubeDirector applications added to the following namespaces are automatically applied to future tenants:
  - `kd-apps`
  - `kd-mlops`
  - `kd-spark`
- If your platform does not have a Spark or ML Ops license, only KubeDirector applications added to the `kd-apps` namespace are automatically applied to future tenants.

- **Spec name:** Name of the application being launched. Do not modify this value.
- **appCatalog:** Source of the application being launched inside this pod. Do not modify this value.
- **Roles:** Each application requires virtual nodes/containers with one or more roles. The correct name and number of roles appears in the **Launch Kubernetes Applications** screen by default, and the following information appears for each role:
  - **id:** name of the role. Do not modify this value.
  - **members:** number of virtual containers to create with this role. For example, `members: 3` means that three virtual nodes will be created for a role.
  - **Resource requests:** Requested amount of memory and CPU resources to use for each container of the current role that is being created in this pod. For example, `memory: "4Gi"` and `cpu: "2"` means that each container will request 4GB of RAM and two CPU cores.
  - **Resource limits:** Maximum amount of memory and CPU resources that can be used by each container of the current role that is being created in this pod. For example, `memory: "4Gi"` and `cpu: "2"` means that each container can use a maximum of 4GB of RAM and two CPU cores.

  If the application will use GPU resources, additional information is required. See Deploying Applications That Use GPU Resources on page 565

7. Click **Submit**.

HPE Ezmeral Runtime Enterprise returns you to the **Kubernetes Applications** screen. The new pod that you just created appears in the **KubeDirector Running Applications** table. When the **Status** of this pod changes to **ready**, then you may access the service endpoints within that pod using either the command line or the **Service Endpoints** tab (see Service Endpoints Tab).

CAUTION:

Improper parameter modification may cause pod creation to fail or other undesirable behavior.

Deploying Applications That Use GPU Resources

If you specify a nonzero value for GPU on any HPE Ezmeral Runtime Enterprise UI screen that creates an application (such as when you create a notebook), the required environment variable settings are

added to the YAML file automatically. If you edit the YAML file manually, you must ensure that the NVIDIA_DRIVER_CAPABILITIES environment variable is configured.

- Set the NVIDIA_DRIVER_CAPABILITIES environment variable to `'compute,utility'` for every role that uses a GPU.
- Specify GPU resources in the resource requests and limits.

  For examples of specifying MIG resources, see Using GPUs in Kubernetes Pods on page 727.

For example:

```
apiVersion: "kubedirector.hpe.com/v1beta1"
kind: "KubeDirectorCluster"
metadata:
  name: "centos7gpu"
  namespace: "gput"
  labels:
    description: ""
spec:
  app: "centos7x"
  namingScheme: "CrNameRole"
  appCatalog: "local"
  roles:
    -
      id: "vanilla_centos"
      members: 1
      env:
        -
          name: "NVIDIA_DRIVER_CAPABILITIES"
          value: "compute,utility"
      resources:
        requests:
          cpu: "2"
          memory: "4Gi"
          nvidia.com/gpu: "1"
        limits:
          cpu: "2"
          memory: "4Gi"
          nvidia.com/gpu: "1"
```

Deploying Applications in an Air-Gap Environment

If Kubernetes is deployed in an air-gap configuration (see Kubernetes Air-Gap Requirements and Air Gap Tab), then you will need to manually edit `KubeDirectorApp` resources, as follows:

1. Retrieve the desired image from the bluedata Docker hub, and then upload that image to your locally-accessible repo. There are a few cases in which the path in the repository does not match the common name. For example:

Common Name	Path Name
Jupyter notebook	`/kd-notebook`
Training engine (controller)	`/kd-training`
Training engine (REST server)	`/kd-api-serving`
Deployment Engine	`/kd-api-serving`

2. In the `kd-apps` namespace, change the `defaultImageRepoTag` and (if present) `imageRepoTag` parameters to identify the image on a locally-accessible repo.

For example, for the `spark221e2` KubeDirector App, execute the command `kubectl -n kd-apps edit kdapp spark221e2`, and then modify the `defaultImageRepoTag` and `imageRepoTag` parameters.

NOTE: Modifying the Kubernetes cluster in this fashion will affect the default KubeDirector app catalog for each subsequently-created tenant in that cluster.

Enabling SSH Access to KubeDirector Application Pods

To enable SSH access to KubeDirector application pods:

1. Execute the following command to initialize and open a web terminal to connect to the pod:

   ```
   kubectl exec -it <pod name> - - /bin/bash
   ```

   For example:

   ```
   #> kubectl exec -it kdss-66ddf-0 - - /bin/bash
   pod_terminal#>passwd bluedata     ### provide password or add user using
   adduser -p command
   pod_terminal#>exit
   ```

2. SSH from your web terminal or other client in the usual way by connecting to the access point displayed on the **Service Endpoints** tab of the **Kubernetes Applications** screen. See Service Endpoints Tab. For example:

   ```
   ssh bluedata@<access point> -p <port number>
   #> ssh bluedata@vm188.mycorp.net -p 10011
   ```

3. Provide your password to complete the connection.

Onboarding Applications from an FS Mount

Deploy Kubernetes YAML applications onto HPE Ezmeral Runtime Enterprise from a filesystem mount using **Kubectl** tab of the **Kubernetes Applications** screen.

Prerequisites

Required access rights: Kubernetes Tenant Member

About this task

NOTE:

This function is not available for external Kubernetes clusters. See Importing an External Kubernetes Cluster.

Use this procedure to deploy a YAML-based Kubernetes application that you upload to a file system mount.

For examples of creating custom YAML applications, see: Sample YAML Reference Programs

To deploy a KubeDirector application you use a different procedure. See Deploying KubeDirector Applications on page 563.

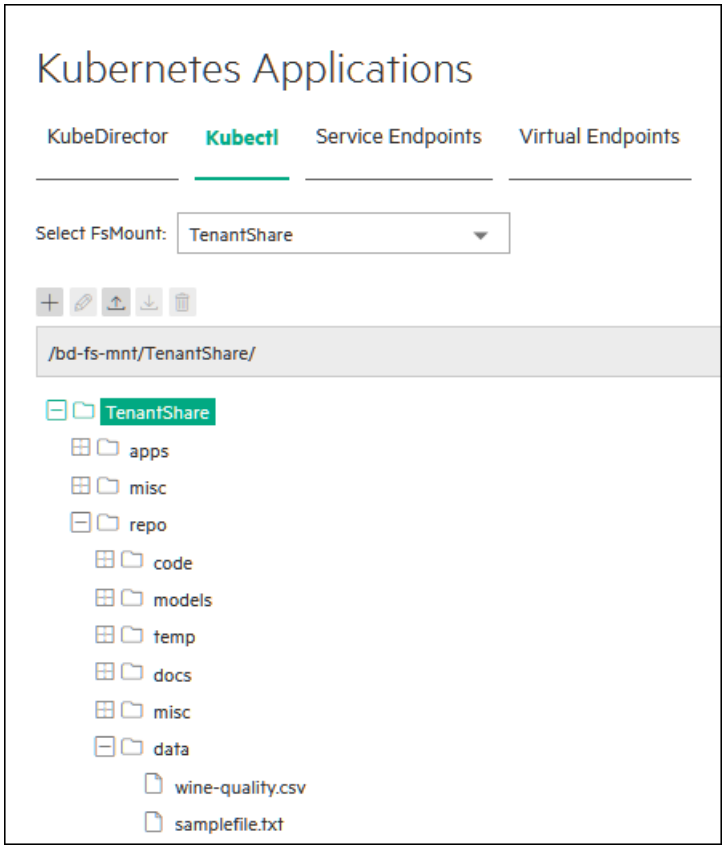
Procedure

1. Log into the web interface as a member of the Kubernetes cluster in which you want to onboard the application.

2. In the main menu, select **Applications** to open the **Kubernetes Applications** screen.

   See the Kubernetes Applications Screen.

3. If needed, select **Kubectl** to open the **Kubectl** tab.



4. From the **Select FS Mount** menu, select the FS mount to use.

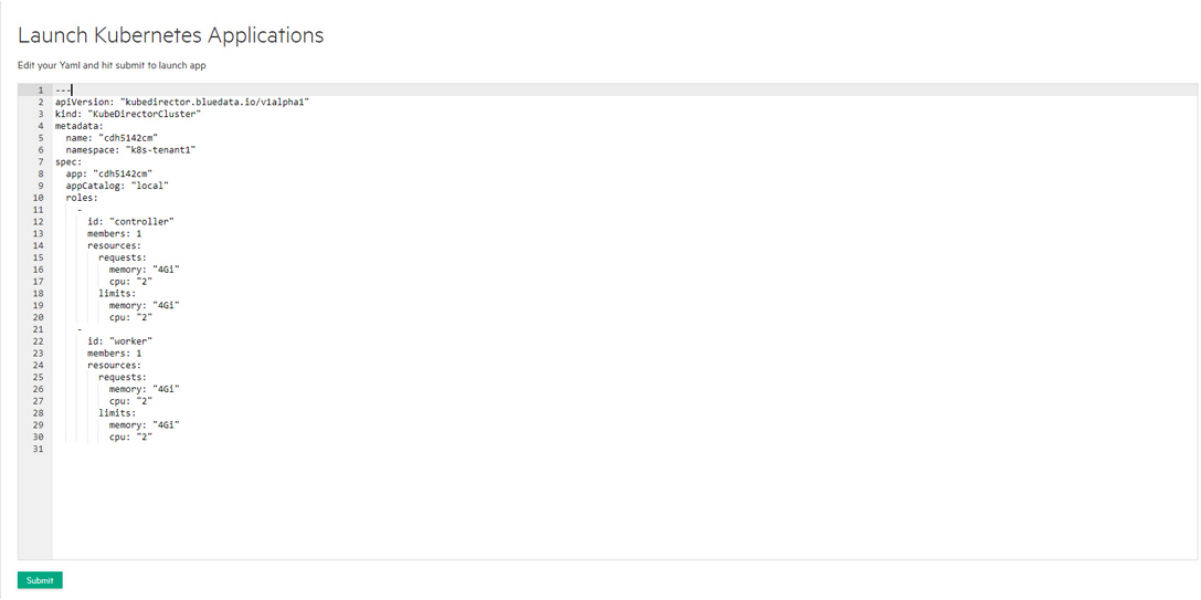
   For more information about filesystem mounts, see About FS Mounts and Creating a New FS Mount.

5. If needed, upload the desired YAML file to the FS mount.

   You may choose to place this file in the `apps` folder or anywhere else. For information about uploading and downloading files, see Uploading and Downloading Files.

6. Select the YAML file, and then click **Open**.

   The **Launch Kubernetes Applications** screen appears.



7. You may freely edit the YAML file as desired.

   Any edits will affect the `kubectl` operation, but will not change the YAML file on disk.

   If your application will use GPU resources, see also Using GPUs in Kubernetes Pods.

8. Click the appropriate button to proceed:
   - **Apply**: Executes the command `kubectl apply` on the YAML file.
   - **Create**: Executes the command `kubectl create` on the YAML file.
   - **Delete**: Executes the command `kubectl delete` on the YAML file.

Updating KubeDirector Applications

Upgrading HPE Ezmeral Runtime Enterprise to a newer version automatically updates KubeDirector applications; however, you may need to upgrade KubeDirector applications without upgrading your HPE Ezmeral Runtime Enterprise deployment. To do this:

1. Execute the following command to delete the old application version:

```
kubectl delete kubedirectorapps.kubedirector.hpe.com
<old_application_name>
```

2. Execute the following command to create or update new applications:

```
kubectl apply -f cr-app-<new_application_name>.json
```

Platform Administration

Tasks and reference information for Platform Adminstrators (Site Administrators) managing the HPE Ezmeral Runtime Enterprise deployment.

The Platform Administrator (Site Admin) manages the site as a whole, and aspects of the HPE Ezmeral Runtime Enterprise deployment that are larger in scope than Kubernetes clusters.

Platform Administrator Overview

Platform Administrators can manage Big Data tenants, AI/ML projects, users, and the infrastructure that supports those tenants and projects. They can also manage the global settings that affect the entire deployment. The articles that describe managing tenants, projects, and hosts/applications that support them appear in the following categories:

- Interface
- Hosts

See also:

- Global Settings Overview for a list of articles that describe managing the "global" settings affect the entire deployment.
- Kubernetes Overview for a list of articles that describe managing Kubernetes within the deployment.

Interface

These articles describe the Platform Administrator interface:

- Dashboard - Platform Administrator on page 570
- Toolbar & Main Menu - Platform Administrator on page 575
- Global Settings Overview - These options are where the Platform Administrator manages settings that affect the entire deployment.

Hosts

These articles describe using the **Installation** screen to manage hosts for Big Data tenants and AI/ML projects.

- About Tags
- To install a Kubernetes Worker, see Kubernetes Worker Installation Overview.
- For help installing a Gateway host, see Gateway Installation Tab.
- Decommissioning/Deleting a Kubernetes Host.

Dashboard - Platform Administrator

The Platform Administrator **Dashboard** screen by either selecting **Dashboard** in the main menu.
The Platform Administrator **Dashboard** screen presents a high-level overview of current activity. (See Dashboard - Kubernetes Administrator for information about the Kubernetes dashboard.)

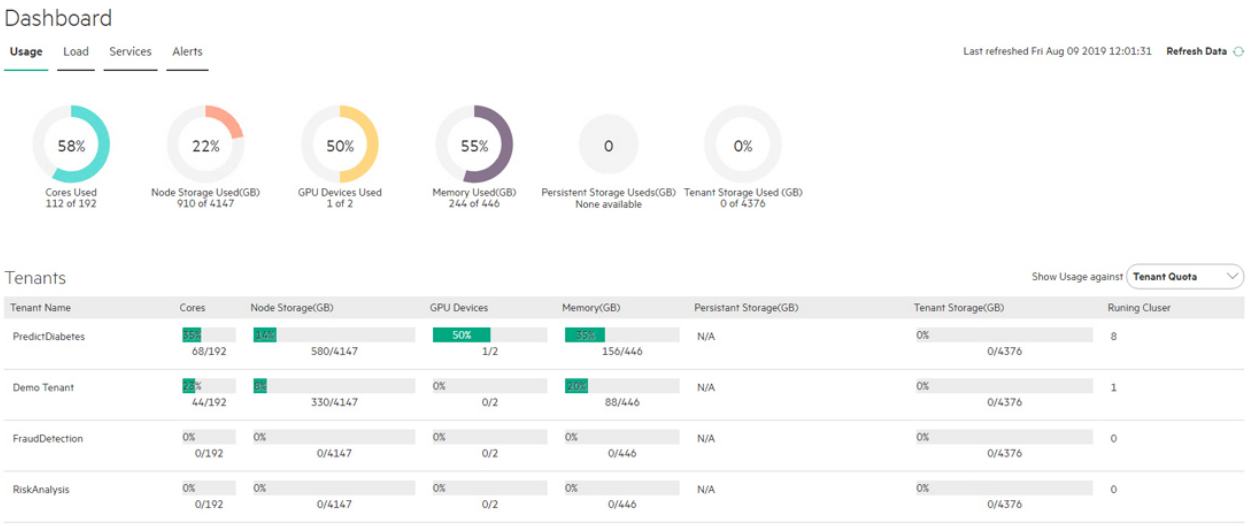
The top of this screen contains the **Refresh Data** function, which displays the date and time of the most recent **Dashboard** refresh. Clicking the **Refresh Data** button refreshes the data on this screen.

The following tabs are available:

- **Usage:** This tab displays usage information on a per-tenant basis. See Usage Tab.
- **Load:** This tab displays load statistics for on-premises CPU, memory, and network resources. See Load Tab.
- **Services:** This section displays the health status for each component service for each host. See Services Tab.
- **Alerts:** This tab displays any alert messages generated by the system. See Alerts Tab.

Usage Tab

The **Usage** tab displays usage statistics for the Big Data tenants and AI/ML projects.



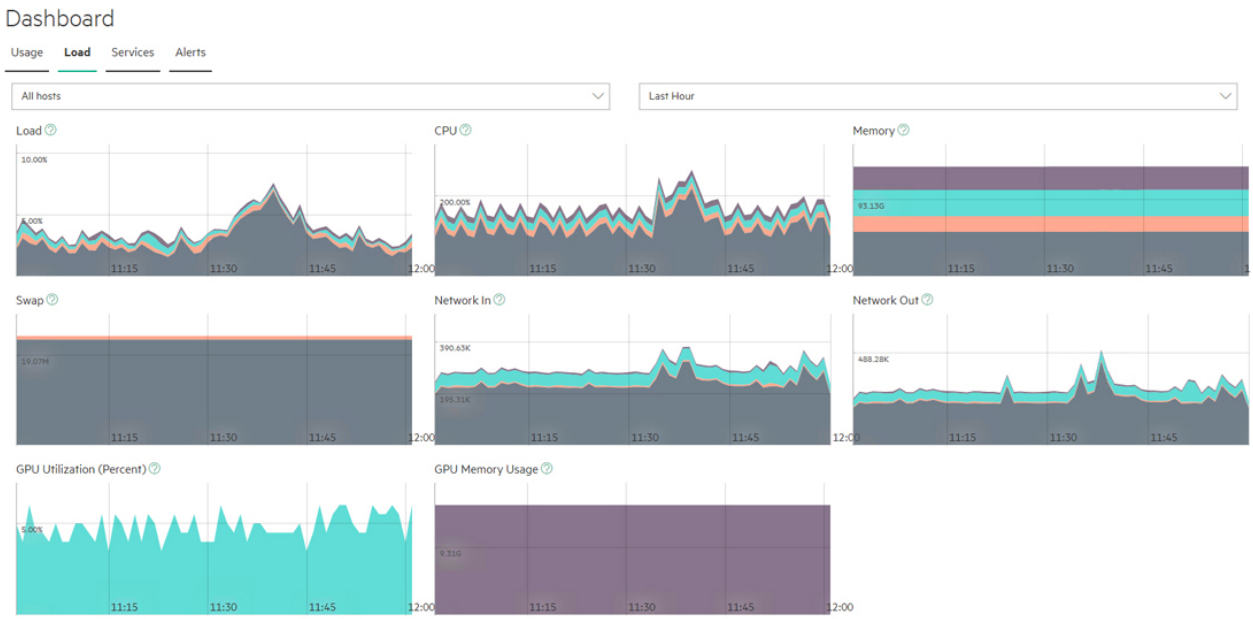
The top of the **Usage** tab displays dials showing the following aggregate information for all of the tenants/projects:

- **Cores Used:** Percentage of available virtual CPU cores being used by all of the tenants.
- **Node Storage Used (GB):** Percentage of available node storage being used by all of the nodes.
- **GPU Devices Used:** Number of GPU devices being used and the total number of GPUs, if any. This only appears if the deployment is running RHEL/CentOS 7.x.
- **Memory Used (GB):** Percentage of available RAM being used by all of the tenants.
- **Persistent Used (GB):** Percentage of available persistent storage used and the total available persistent storage, in GB.
- **Tenant Storage Used (GB):** Percentage of available tenant storage being used by all of the tenants.

The bottom of this tab contains a table that lists all of the tenants and projects. This table displays the **Tenant Name**, **Cores Used**, **Node Storage Used (GB)**, **GPUs Used**, **Memory Used (GB)**, **Persistent Used (GB)**, **Tenant Storage Used (GB)**, and the number of **Running Clusters** being used by that tenant. This number is expressed as **x** of **y**, where **x** is the allotted number and **y** is either the Tenant Quota or total System Resources, depending on your **Show Usage against** menu selection.

Load Tab

The **Load** tab displays a series of dials and charts. Hovering the mouse over a bar opens a popup with more detailed information for the selected time.



This tab shows the following information for the selected time period:

- **Load:** One-minute average system load percentage for the selected host(s) over the selected time period.
- **CPU:** Percentage of host CPU utilization across all user space processes that are currently running for the selected host(s) over the selected time period. On multi-core systems, the percentages can be greater than 100%.
- **Memory:** Current use of host memory across all cluster processes for the selected host(s) over the selected time period.
- **Swap:** Amount of swap-file usage over the selected time period for the selected host(s) over the selected time period, in GB.
- **Network In:** Amount of incoming host network bandwidth being used by the selected host(s) over the selected time period.
- **Network Out:** Amount of outgoing host network bandwidth being used by the selected host(s) over the selected time period.

The following additional information applies to tenants with GPUs enabled:

- **GPU Utilization (percent):** Selecting **All hosts** in the left pull-down menu displays aggregate GPU utilization in percent per host. Selecting an individual host displays per-GPU utilization for that host.
- **GPU Memory Usage:** Selecting **All hosts** in the left pull-down menu displays aggregate GPU memory usage in percent per host. Selecting an individual host displays per-GPU memory usage for that host.

You may select the host(s) you want to view and also adjust the time period for which results appear using the pull-down menus at the right side of the **Load** tab. The available options are:

- Last Hour (default)
- 6 Hours
- Day

- Week

Services Tab

The **Services** tab displays the status of services for each host being used for Big Data tenants and/or AI/ML projects.

This tab displays information such as (but not necessarily limited to) the following for each BD/AI/ML host:

- **Host Name:** Name of the host.
- **Virtual Node Count:** Number of virtual nodes running on that host.
- **BlueData:** This group displays the following information:
  - **Management:** Status of the management service, which handles back-end administration tasks.
  - **Data Server:** Status of the data service agent, which acts as an intermediary between the file system and other entities. This service establishes communication between a host's virtual nodes and the Caching Node service. On the Controller host, the Data Server also receives DataTap browsing queries from the Management Service.
  - **Caching Node:** Status of the data service, which communicates with the storage services referenced by DataTaps. This service provides an accelerated I/O channel between those storage services and the applications running in virtual nodes. On the Controller host, this service also provides the back end for DataTap browsing.
  - **Hypervisor Controller:** Status of the hypervisor controller, which manages virtual nodes (containers) along with the Hypervisor Agent service.
  - **Hypervisor Agent:** Status of the hypervisor agent, which manages virtual nodes (containers) along with the Hypervisor Controller service.
  - **Monitoring Collector:** Status of the monitoring engine that collects performance, usage, and other metrics.
  - **Monitoring Database:** Status of the database that stores monitoring information.
- **HA:** This group only appears if High Availability is enabled. When enabled, this group displays the following information:
  - **HA Engine:** Health of the High Availability Engine, which is the central High Availability state transitions executing unit. The HA Engine performs various tasks specific to High Availability in response to requests from other services and must be running on both the Controller and Shadow Controller hosts.

- **HA Status:** Status of the High Availability service. The node for which this service appears is functioning as the Controller host. If this dot is green, then the High Availability host is functioning normally and all hosts (Controller, Shadow Controller, and Arbiter) are up. This dot appears as yellow if one of these three hosts has failed to indicate that the High Availability cluster has been degraded and that the deployment is not protected against any further host failure. If the dot is red, then High Availability protection is not currently functional.
- **Pacemaker:** Health of the High Availability cluster polling service. This service periodically polls the High Availability cluster and, in the event of a failure, triggers failover state transition in the HA Engine. This service must be running on both the Controller and Shadow Controller hosts.
- **Cluster Management:** Health of the cluster manager daemon that helps the Pacemaker service perform its periodic polling and trigger failover/failback. This daemon must be running on both the Controller and Shadow Controller hosts.

NOTE: Only one host (either the Controller or Shadow Controller) functions as the Controller at any given time.

- **Infrastructure:** Describing the individual items in this group is beyond the scope of this manual. This group includes the **HA Proxy** service. This is the service that runs on Gateway host(s). If this service is down, then end users will not be able to access virtual cluster service endpoints.
- **Actions:** The **Actions** column of the table includes a **Check Now** icon (folder) for each host. Clicking this icon refreshes the status of all listed services for the selected host.

The status of a service can be either **OK** (green dot), **CRITICAL** (red dot), or **DISABLED** (intentionally not running; gray dot). Hovering the mouse over the status button opens a popup with additional information. In general:

- The Controller host must not display any red dots. If the Controller host has one or more error(s), then HPE Ezmeral Runtime Enterprise may not function properly.
- If all of the dots for a Worker host are red, then that host will not be able to provide resources. This situation typically occurs because the host has been powered off, has lost network connectivity, or because HPE Ezmeral Runtime Enterprise is not properly installed.
- A Worker host with some red and some green dots may cause some operations to fail, unless the errors are transient conditions caused by the host powering on or regaining network connectivity.

Please generate a support bundle and then contact Hewlett Packard Enterprise Technical Support if a host that is reporting service errors meets all of the following criteria:

- HPE Ezmeral Runtime Enterprise is completely installed.
- The host is powered on.
- The host has network connectivity.

See The Support/Troubleshooting Screen on page 922 and Generating a Support Bundle on page 926.

Alerts Tab

The **Alerts** tab displays any alert messages from the Caching Node, Data Server, and Management services.

The following alerts appear in this tab:

- **Notifications:** Routine messages. A green dot appears next to each routine notification.
- **Error:** A minor error has occurred. A gray dot appears next to each error notification.
- **Warning:** A serious error has occurred. An orange dot appears next to each warning notification.
- **Critical:** A critical error has occurred. A red dot appears next to each critical notification.

NOTE: The presence of non-routine alerts does not mean that HPE Ezmeral Runtime Enterprise will not function normally.

Toolbar & Main Menu - Platform Administrator

Describes the toolbar and navigation sidebar available to Platform Administrators in HPE Ezmeral Runtime Enterprise.

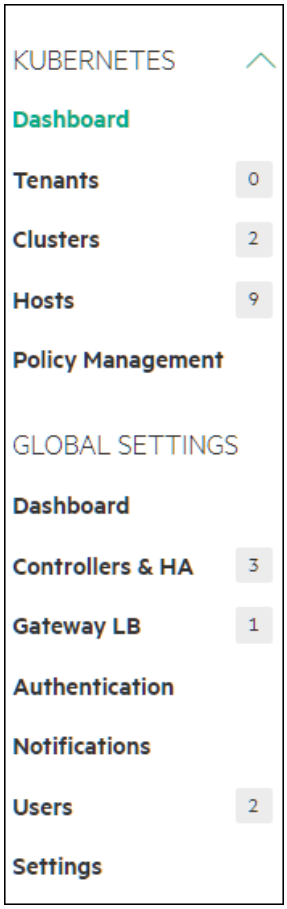
This article describes the UI items for Platform Administrators.

Toolbar

The layout of the Toolbar is the same as described in Navigating the GUI on page 143. For information about the content of the **Quick Access** menu for Platform Administrators, see Quick Access Menu - Platform Administrator on page 577.

Main Menu - Platform Administrator

The Platform Administrator **Main Menu** appears as shown in the following image.



For Platform Administrators, the **Main Menu** includes the following:

- **KUBERNETES** section:

Dashboard	Opens the **Kubernetes Dashboard** screen.
Tenants	Displays the number of Kubernetes tenants and opens the **Kubernetes Tenants** screen, which enables you to add, modify, and delete tenants and/or projects. When Platform Administrators open the screen for a particular tenant or project, they act as a Tenant Administrator or Project Administrator, and can perform tenant administration or project administration tasks. See Kubernetes Tenant Administration on page 450.
Clusters	Displays the number of Kubernetes clusters, and opens the **Kubernetes Clusters** screen, which enables you to add, manage, edit, and delete Kubernetes clusters. See The Kubernetes Clusters Screen on page 457.
Hosts	Displays the number of hosts and opens the **Kubernetes Hosts Installation** screen, which enables you to manage Kubernetes hosts. To manage High Availability and Gateway LB hosts, see Global Settings Overview.
Policy Management	Opens the **Git Repositories for policies** screen, which enables you to manage policies stored in a Git repository and apply them to Kubernetes

	clusters automatically. This feature is not available in HPE Ezmeral Runtime Enterprise Essentials. See Centralized Policy Management on page 336.

- **GLOBAL SETTINGS** section:

Dashboard	Opens the deployment **Dashboard** screen.
Controllers & HA	Opens the **Controllers & HA** screen, which enables the Platform Administrator to configure platform High Availability (HA) and add the hosts that will become the Shadow Controller and Arbiter hosts.
Gateway LB	Opens the **Gateway LB** screen, which enables the Platform Administrator to add Gateway hosts and manage Gateway settings.
Authentication	Opens the **User Authentication** screen, which enables the Platform Administrator to configure user authentication settings.
Notifications	Opens the **Notification Settings** screen, which enables the Platform Administrator to configure the deployment to deliver Nagios alerts.
Users	Opens the **User Management** screen, which enables the Platform Administrator to view the current user sessions in the deployment, add or delete users amd groups, and to manage user and group role assignments.
Settings	Opens the **System Settings** screen, which enables the Platform Administrator to manage platform-wide configuration settings, such as tenant storage, air gap configuration, software updates, and licenses.

Quick Access Menu - Platform Administrator

For Platform Administrators, the following items appear in the **Quick Access** menu:

Create Tenant	Opens the **Create New Tenant** screen, which allows you to create a new tenant or AI/ML project.
Add User	Opens the **Create New User** screen, which enables you to add a new user to the local user database. See Creating a New User (Local) on page 776.
Assign Users	Opens the **Assign Users** screen, which enables you to grant roles to users. See Assigning/Revoking User Roles (Local) on page 771 or Assigning/Revoking User Roles (LDAP/AD/SAML) on page 774, as appropriate.
Enter/Exit system lockdown	When enabled, Lockdown mode prevents users from making any changes to the deployment. See Lockdown Mode on page 916.
User Info	Opens the **Current User Information** dialog, which lists your role, current project, and username.

User Guide	Opens this *User and Administrator Guide*.
Support	Opens the **Support/Troubleshooting** screen, which enables the Platform Administrator to generate, download, and delete support bundles, perform configuration checks, and search platform logs. See The Support/Troubleshooting Screen on page 922.
Privacy	Opens the **Hewlett Packard Enterprise Privacy Statement** web page in a new browser tab or window.
Version	Displays version and build information about the HPE Ezmeral Runtime Enterprise deployment.

Related reference

Users and Roles on page 130

HPE Ezmeral Runtime Enterprise new UI on page 146
Introduces the HPE Ezmeral Runtime Enterprise UI that is the primary interface used to access machine learning (ML Ops) projects, and tenants that use analytics applications, such as Spark.

HPE Ezmeral Data Fabric Introduction

HPE Ezmeral Data Fabric is a platform for data-driven analytics, ML, and AI workloads. The patented file-system architecture was designed and built for performance, reliability, and scalability. HPE Ezmeral Runtime Enterprise supports multiple implementations of HPE Ezmeral Data Fabric.

The HPE Ezmeral Data Fabric platform serves as a secure data store and provides file storage, NoSQL databases, object storage, and event streams. The patented file-system architecture was designed and built for performance, reliability, and scalability.

HPE Ezmeral Runtime Enterprise can connect to following implementations of HPE Ezmeral Data Fabric:

IMPORTANT: Even though multiple Data Fabric storage deployments might be available, ONLY ONE Data Fabric deployment can be registered as tenant storage.

HPE Ezmeral Data Fabric on Bare Metal	**HPE Ezmeral Data Fabric on Bare Metal** is an implementation of HPE Ezmeral Data Fabric that is on physical or virtual machines that are not part of the HPE Ezmeral Runtime Enterprise deployment. HPE Ezmeral Data Fabric on Bare Metal is the supported implementation of HPE Ezmeral Data Fabric for production deployments of HPE Ezmeral Runtime Enterprise. To register HPE Ezmeral Data Fabric on Bare Metal as tenant storage, see HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579
HPE Ezmeral Data Fabric on Kubernetes	HPE Ezmeral Data Fabric on Kubernetes is an implementation of HPE Ezmeral Data Fabric in a Kubernetes cluster. HPE Ezmeral Data Fabric on Kubernetes is available for use in non-production deployments of HPE Ezmeral Runtime Enterprise, but it is not supported for production environments. To register HPE Ezmeral Data Fabric on Kubernetes, see Registering HPE Ezmeral Data Fabric on Kubernetes as Tenant Storage.

Embedded Data Fabric	**Embedded Data Fabric** is an implementation of HPE Ezmeral Data Fabric that is locally **Embedded** and runs on HPE Ezmeral Runtime Enterprise hosts. **Embedded Data Fabric** is not supported on HPE Ezmeral Runtime Enterprise 5.5.0 and later. To migrate an existing Embedded Data Fabric deployment from a prior release of HPE Ezmeral Runtime Enterprise, contact Hewlett Packard Enterprise Technical Support.

More Information

Videos and tutorials: HPE Developer site for Ezmeral Data Fabric

HPE Ezmeral Data Fabric as Tenant/Persistent Storage

For an implementation of HPE Ezmeral Data Fabric to be used as tenant/persistent storage, it must be registered in HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise supports the use of exactly one implementation of HPE Ezmeral Data Fabric as tenant/persistent storage, as follows:

HPE Ezmeral Data Fabric on Bare Metal	HPE Ezmeral Data Fabric on Bare Metalis external to HPE Ezmeral Runtime Enterprise. You register this implementation after you have installed and verified HPE Ezmeral Runtime Enterprise. See HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579
HPE Ezmeral Data Fabric on Kubernetes	If you implement **HPE Ezmeral Data Fabric on Kubernetes**, you register the Data Fabric as tenant/persistent storage as a step during the Data Fabric cluster creation process. See Registering HPE Ezmeral Data Fabric on Kubernetes as Tenant Storage.
Embedded Data Fabric	If your deployment had an existing Embedded Data Fabric before you upgraded to HPE Ezmeral Runtime Enterprise version 5.4.0 or later, that implementation was registered as tenant/persistent storage during the Platform Controller Setup portion of the HPE Ezmeral Runtime Enterprise installation procedure.

Registering HPE Ezmeral Data Fabric on Bare Metal as Tenant Storage

This procedure describes registering HPE Ezmeral Data Fabric on Bare Metal as Tenant Storage. An HPE Ezmeral Data Fabric on Bare Metal cluster is external to the HPE Ezmeral Runtime Enterprise installation. After you have installed or upgraded to HPE Ezmeral Runtime Enterprise 5.5.0 or later, you can register the same HPE Ezmeral Data Fabric on Bare Metal cluster as Tenant Storage by multiple HPE Ezmeral Runtime Enterprise instances.

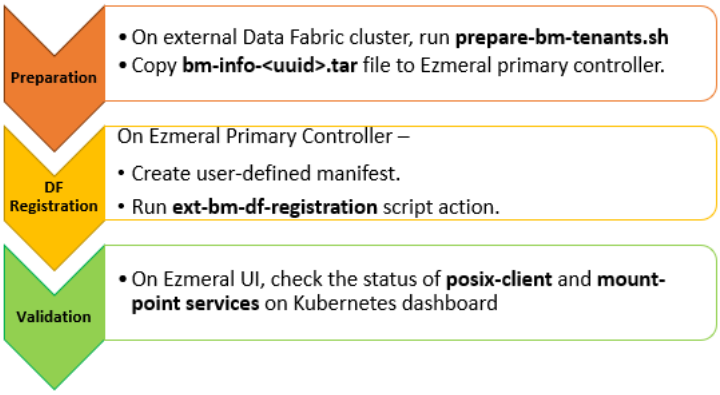
Prerequisites

NOTE: You must read all sections before proceeding to perform the procedure.

- The user who performs this procedure must have Platform Administrator access to HPE Ezmeral Runtime Enterprise.
- Activity must be quiesced on the relevant clusters in the HPE Ezmeral instance.
- The HPE Ezmeral Runtime Enterprise deployment must not have configured tenant storage.

- An HPE Ezmeral Data Fabric on Bare Metal cluster must have been deployed. See HPE Ezmeral Data Fabric Documentation for more details on a HPE Ezmeral Data Fabric on Bare Metal cluster.
- When deploying the Data Fabric on Bare Metal cluster:
  - Keep the **UID** for the `mapr user` at the default of 5000.
  - Keep the **GID** for the `mapr group` at the default of 5000.
  - The Data Fabric (DF) cluster on Bare Metal must be a SECURE cluster.
  - Data At Rest Encryption (DARE) must have been enabled on the DF cluster on Bare metal. If deploying a new DF cluster on Bare metal, enable DARE during the installation. To enable DARE on an existing Data Fabric cluster on Bare metal, see Enabling Encryption of Data at Rest.
  - For compatibility information, see Support Matrixes on page 54.
- Data Fabric volumes which match per-tenant volume names, must not exist on the Data Fabric on Bare Metal cluster. For more information, see Administering volumes

About this task



An HPE Ezmeral Runtime Enterprise can connect to multiple Data Fabric storage deployments; however, only one Data Fabric deployment can be registered as tenant storage.

- If you have an HPE Ezmeral Data Fabric on Bare Metal cluster outside the HPE Ezmeral Runtime Enterprise, and if you want to configure HPE Ezmeral Data Fabric on Bare Metal as tenant storage, continue with this procedure.
- If you have already registered another Data Fabric instance as tenant/persistent storage, do not proceed with this procedure. Contact Hewlett Packard Enterprise Support if you want to use a different Data Fabric instance as tenant storage.

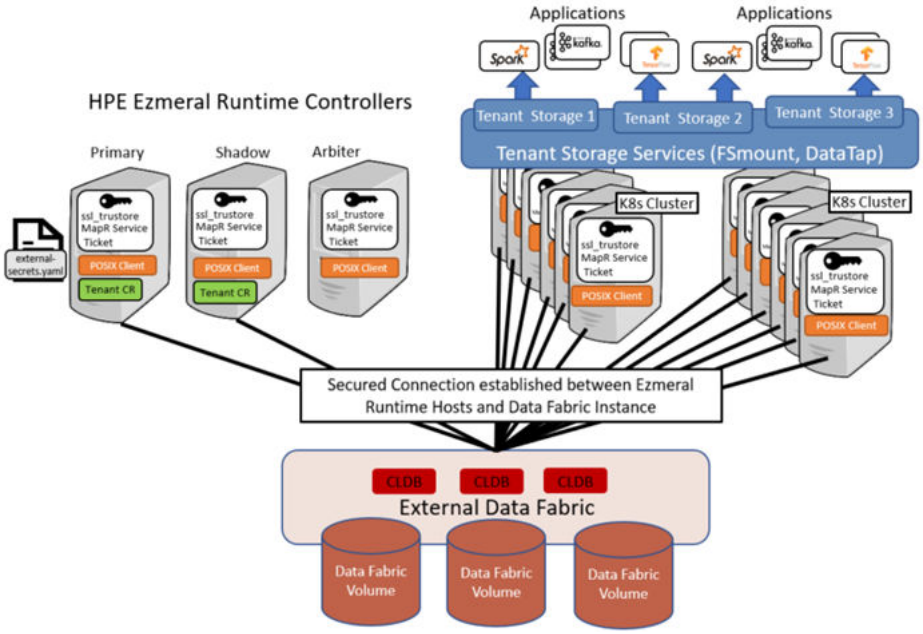
NOTE: After you have installed or upgraded to HPE Ezmeral Runtime Enterprise 5.5.0 or later:

- It is no longer necessary to dedicate an HPE Ezmeral Data Fabric on Bare Metal cluster to one HPE Ezmeral Runtime Enterprise installation.
- Multiple HPE Ezmeral Runtime Enterprise installations may register the same HPE Ezmeral Data Fabric on Bare Metal cluster as the backing for their tenant storage.
- On each HPE Ezmeral Runtime Enterprise installation, all tenants will have their tenant storage backed by the same registered HPE Ezmeral Data Fabric on Bare Metal cluster.

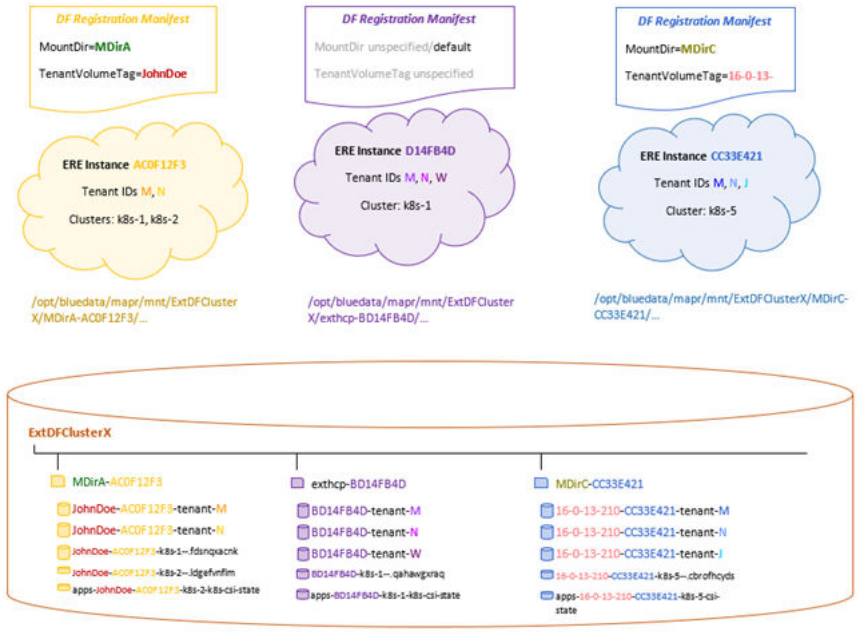
The Registration procedure described herein must be run on each HPE Ezmeral Runtime Enterprise installation.

This procedure may require 10 minutes or more per EPIC or Kubernetes host [Controller, Shadow Controller, Arbiter, Master, Worker, and so on], as the registration procedure configures and deploys Data Fabric client software on each host.

After Data Fabric registration is completed, the configuration will look as follows:



The following image shows an example of a configuration in which multiple HPE Ezmeral Runtime Enterprise installations have registered the same Bare Metal Data Fabric cluster as their tenant storage.



Registration Steps - A Short Summary:

This section provides a quick reference for the steps required for registration. For detailed instructions, refer to the Procedure section:

- Log in as mapr user, to a node of the HPE Ezmeral Data Fabric on Bare Metal cluster, on which the CLDB and Apiserver services are running, and:
  - `mkdir <working-dir-on-bm-df>/`

.

- On the Primary Controller of HPE Ezmeral Runtime Enterprise installation, do the following:
  - ```
    scp /opt/bluedata/common-install/scripts/mapr/gen-external-secrets.sh
    mapr@<cldb_node_ip_address>:<working-dir-on-bm-df>/
    ```
  - ```
    scp /opt/bluedata/common-install/scripts/mapr/prepare-bm-tenants.sh
    mapr@<cldb_node_ip_address>:<working-dir-on-bm-df>/
    ```
  - ```
    mkdir /opt/bluedata/tmp/ext-bm-mapr/
    ```
- Create a user-defined manifest for the procedure:
  - If you are not specifying any keys (i.e. to generate default values for all keys):

    ```
    touch /opt/bluedata/tmp/ext-bm-mapr/ext-dftenant-manifest.user-defined
    ```
  - Else, specify the following parameters:
    - ```
      cat << EOF > /opt/bluedata/tmp/ext-bm-mapr/
      ext-dftenant-manifest.user-defined
      EXT_MAPR_MOUNT_DIR="/
      <user_specified_directory_in_mount_path_for_volumes>"
      TENANT_VOLUME_NAME_TAG="<user_defined_tag_to_be_included_in_tenant_volu
      me_names>"
      EOF
      ```
- On the CLDB node of the HPE Ezmeral Data Fabric on Bare Metal cluster:
  - `cd <working-path-on-bm-df>/`
  - `./prepare-bm-tenants.sh`
- On the Primary Controller of HPE Ezmeral Runtime Enterprise:
  - Move or remove any existing `bm-info-*.tar` from `/opt/bluedata/tmp/ext-bm-mapr/`
  - `scp mapr@<cldb_node_ip_address>:< working-dir-on-bm-df>/ bm-info-*.tar /opt/bluedata/tmp/ext-bm-mapr/`
  - `cd /opt/bluedata/tmp/ext-bm-mapr/`
  - `LOG_FILE_PATH=<log_file_path> /opt/bluedata/bundles/hpe-cp-*/ startscript.sh --action ext-bm-df-registration`

Procedure

1. **Preparation (On HPE Ezmeral Data Fabric on Bare Metal Cluster):**
   a) Verify that the HPE Ezmeral Data Fabric on Bare Metal cluster is in good state.

b) Before starting the Registration procedure, on the HPE Ezmeral Runtime Enterprise Primary Controller , make sure that the `prepare-bm-tenants` is run already on the required HPE Ezmeral Data Fabric on Bare Metal cluster. The `prepare-bm-tenants` and `gen-external-secrets.sh` scripts are available on the Ezmeral Primary Controller, under `opt/bluedata/common-install/scripts/mapr/` and may be copied to the external HPE Ezmeral Data Fabric on Bare Metal cluster.

c) **NOTE:** You can run `prepare-bm-tenants` on the HPE Ezmeral Data Fabric on Bare Metal cluster on behalf of a single HPE Ezmeral Runtime Enterprise instance, or multiple HPE Ezmeral Runtime Enterprise instances simultaneously.

To run the `prepare-bm-tenants` script, do the following:

1. With Administrator credentials (such as the *mapr* user), log in to a node of the external HPE Ezmeral Data Fabric on Bare Metal cluster, on which the CLDB and Apiserver services are running.

2. Copy the `prepare-bm-tenants.sh` and `gen-external-secrets.sh` scripts to a CLDB node of the external HPE Ezmeral Data Fabric on Bare Metal cluster, placing both scripts in the same working directory.

3. Ensure the `prepare-bm-tenants.sh` file has executable permission and execute the script.

   Upon successful execution of the `prepare-bm-tenants.sh` script:

   - A file named `bm-info-<8_byte_uuid>.tar` is created in the same directory (A *uuid* is generated during each run of the `prepare-bm-tenants` step).

   - The `bm-info-<8_byte_uuid>.tar` file contains information on the Data Fabric cluster and other results of the `prepare-bm-tenants` step. The `bm-info-<8_byte_uuid>.tar` file must be placed on the HPE Ezmeral Runtime Enterprise Primary Controller, under `/opt/bluedata/tmp/ext-bm-mapr/`, before proceeding to the next step.

2. **Before Registration (On HPE Ezmeral Runtime Enterprise Primary Controller)**:

   Perform the following steps on the HPE Ezmeral Runtime Enterprise Primary Controller host.

   a) EnsurethatHPE Ezmeral Runtime Enterprise isnotcurrentlyinSiteLockdown.

   b) On the HPE Ezmeral Runtime Enterprise Primary Controller host, make sure that the path `/opt/bluedata/tmp/ext-bm-mapr/` is created.

   c) Ensure that `bm-info-<8_byte_uuid>.tar` file is placed under `/opt/bluedata/tmp/ext-bm-mapr/`. Also, ensure that you do not have more than one `bm-info-<uuid>.tar` file under `/opt/bluedata/tmp/ext-bm-mapr/`.

   d) Create a new manifest file named `ext-dftenant-manifest.<user-defined>` under `/opt/bluedata/tmp/ext-bm-mapr/` on the HPE Ezmeral Runtime Enterprise primary Controller host.

   e) Enter the following information in `/opt/bluedata/tmp/ext-bm-mapr/ext-dftenant-manifest.user-defined` :

```
EXT_MAPR_MOUNT_DIR="/<directory_in_mount_path_for_volumes>"
TENANT_VOLUME_NAME_TAG="<user_defined_tag_to_be_included_in_tenant_volum
e_names>"
```

- The `EXT_MAPR_MOUNT_DIR` is an optional parameter. This value must begin with a **/**. It must not equal **/** or /mapr. If you do not specify any value, a default value of `/exthcp-<bdshared_global_uniqueid>` is generated. The `bdshared_global_uniqueid` is automatically generated for the HPE Ezmeral installation.
- The `TENANT_VOLUME_NAME_TAG` is an optional parameter, and it will be included as part of the name for every tenant volume (for the HPE Ezmeral instance) created on the Data Fabric cluster. If specified, the value must only contain characters that are allowed in a volume name, and must not contain the period (.) character.
- The `TENANT_VOLUME_NAME_TAG` specified in `ext-dftenant-manifest.user-defined` influences the tenant volume names for tenants created after the Registration.

3. **Registration**

   The `ext-bm-df-registration` action represents the overall Registration procedure for External HPE Ezmeral Data Fabric on Bare Metal.

   a) To complete the registration procedure, initiate the `ext_register_dftenants` action, using the following command:

   ```
   LOG_FILE_PATH=<path_to_log_file> /opt/bluedata/bundles/hpe-cp-*/
   startscript.sh --action ext-bm-df-registration
   ```

   The LOG_FILE_PATH specified must be a path that exists on all the HPE Ezmeral hosts.

   b) When prompted, enter the Platform Administrator username and password. HPE Ezmeral Runtime Enterprise uses this information for REST API access to its management module.

   NOTE: The `ext-bm-df-registration` action validates the contents of `bm-info-<8_byte_uuid>.tar`, and finalizes the `ext-dftenant-manifest`. The following keys-values will be automatically added to the manifest:

   ```
   CLDB_LIST="<comma-separated;FQDN_or_IP_address_for_each_CLDB_node>"
   CLDB_PORT="<port_number_for_CLDB_service>"
   SECURE="<true_or_false>" (Default is true)
   CLUSTER_NAME="<name_of_DataFabric_cluster>"
   REST_URL="<REST_server_hostname:port>" (or space-delimited list of
   <REST_server_hostname:port> values)
   TICKET_FILE_LOCATION="<path_to_service_ticket_for_HCP_admin>"
   SSL_TRUSTSTORE_LOCATION="<path_to_ssl_truststore>"
   EXT_SECRETS_FILE_LOCATION="<path_to_external_secrets_file>"
   ```

   The`ext-bm-df-registration` actionfailsifvolumes,whichmatchper-tenantvolumenames, existalready ontheexternal HPE Ezmeral Data Fabric on Bare Metal cluster.

   The result of the `ext-bm-df-registration` action is the following:

   - The Data Fabric client is deployed on the client ERE hosts.
   - For each existing tenant, a Data Fabric volume is created on the HPE Ezmeral Data Fabric on Bare Metal cluster.
   - For a new tenant (created in the future), a tenant volume will be created automatically, on the HPE Ezmeral Data Fabric on Bare Metal cluster.
   - Tenant volume names are in the form of `<user-defined-prefix>-<bdshared_global_uniqueid>-tenant-<tenant-id>`, where:

- The `user-defined-prefix` is the value of TENANT_VOLUME_NAME_TAG, if it was specified in `ext-dftenant-manifest.user-defined`.
- *bdshared_global_uniqueid* is an identifier generated automatically for the HPE Ezmeral installation.
- *tenant-id* is a unique identifier for the relevant HPE Ezmeral tenant on the HPE Ezmeral instance.

- **Tenant Storage** is configured to use the HPE Ezmeral Data Fabric on Bare Metal cluster, for all future tenants. And:
  - **TenantStorage** and **TenantShare** are created for all existing tenants on the Data Fabric cluster.
  - Both **TenantShare** and **TenantStorage** are available for all tenants.
- The Registration action also reconfigures the following services:
  - Nagios, to track Data Fabric related client and mount services on the appropriate HPE Ezmeral Runtime Enterprise hosts.
  - WebHDFS, to enable browser-based file system operations, such as upload, mkdir.

Future Kubernetes clusters created in the HPE Ezmeral Runtime Enterprise will have persistent volumes located under `<df_cluster_name>/<ext_mapr_mount_dir>-<bdshared_global_uniqueid>/`

The registered HPE Ezmeral Data Fabric on Bare Metal cluster will be the backing for Storage Classes of future Kubernetes Compute clusters, that are created in the HPE Ezmeral Runtime Enterprise.

The registration procedure does not modify the Storage Classes for Compute clusters, which existed before the registration.

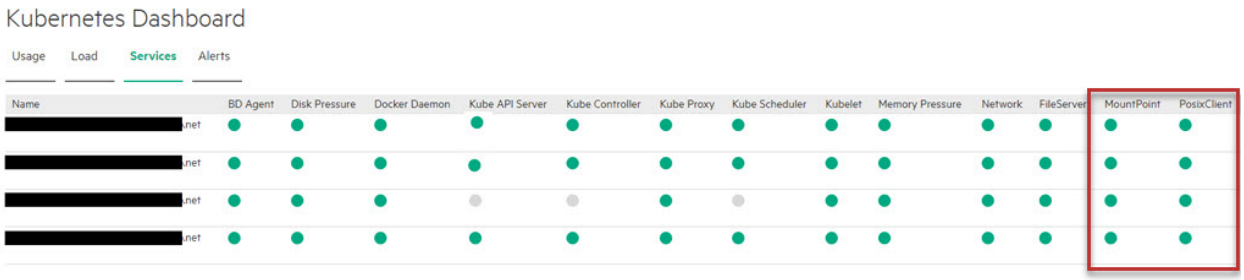
4. **Validation:**

   To confirm that the Registration is completed, check the following:

   a) Check the output and log of the `ext-bm-df-registration` action .

   b) On the HPE Ezmeral Runtime Enterprise Web UI, view the **Tenant Storage** tab on the **System Settings** page. Check that the information displayed on the screen is accurate for the HPE Ezmeral Data Fabric on Bare Metal cluster.

   c) On the HPE Ezmeral Runtime Enterprise, view the **Kubernetes** and **EPIC** Dashboards, and ensure that the POSIX Client and Mount Path services on all hosts are in normal state.



   d) On the HPE Ezmeral Runtime Enterprise web UI, as an authenticated user, check that you are able to browse Tenant Storage on an existing tenant. You can also try uploading a file to a directory under Tenant Storage, and reading the uploaded file. See Uploading and Downloading Files on page 367 for more details.

Registering HPE Ezmeral Data Fabric on Kubernetes as Tenant Storage

This procedure describes registering HPE Ezmeral Data Fabric on Kubernetes as Tenant storage.

Prerequisites

NOTE: Please read the complete procedure before you start this registration process.

- The HPE Ezmeral Runtime Enterprise deployment must not have configured tenant storage. In the HPE Ezmeral Runtime Enterprise (ERE) Web UI, make sure that **Tenant Storage** is set to **None**, in **Settings** screen.
- Make sure that the HPE Ezmeral Data Fabric on Kubernetes cluster does not have pre-existing Data Fabric volumes named in the `tenant-<id>` format. For more information, see Administering Volumes. You can also run the following command inside the `admincli-0` pod:

  ```
  maprcli volume list -columns volumename | grep tenant
  ```

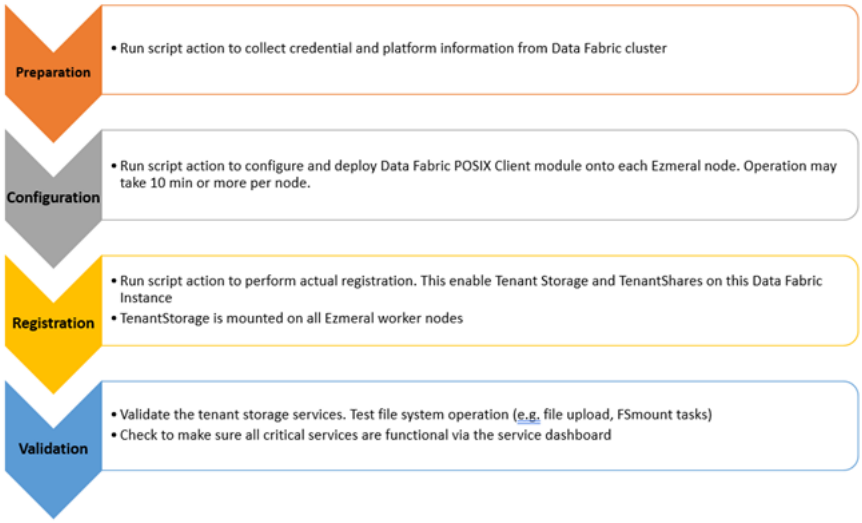
  If any Data Fabric volume exists, you can conclude that the Data Fabric cluster is already registered as tenant storage. Contact Hewlett Packard Enterprise Support for technical assistance.
- An HPE Ezmeral Data Fabric on Kubernetes cluster must have been created. See HPE Ezmeral Data Fabric Documentation for more details on a HPE Ezmeral Data Fabric on Kubernetes cluster.
- Before proceeding to register HPE Ezmeral Data Fabric on Kubernetes, you must have created the Data Fabric cluster by performing upto **Step 5: Summary** of the procedure Creating a New Data Fabric Cluster.
- This procedure must be performed by the user who installed HPE Ezmeral Runtime Enterprise.
- This procedure may require 10 minutes or more per EPIC or Kubernetes host (Controller, Shadow Controller, Arbiter, Master, Worker, and so on).
- This procedure must be performed on the primary Controller host.

  If Platform HA is enabled, in the ERE web UI, you can check **Controllers** page to confirm which controller is set *Primary.*

  CAUTION: You will not be able to delete this HPE Ezmeral Data Fabric on Kubernetes cluster after you have completed this step.

last-updated: May 06, 2024

About this task

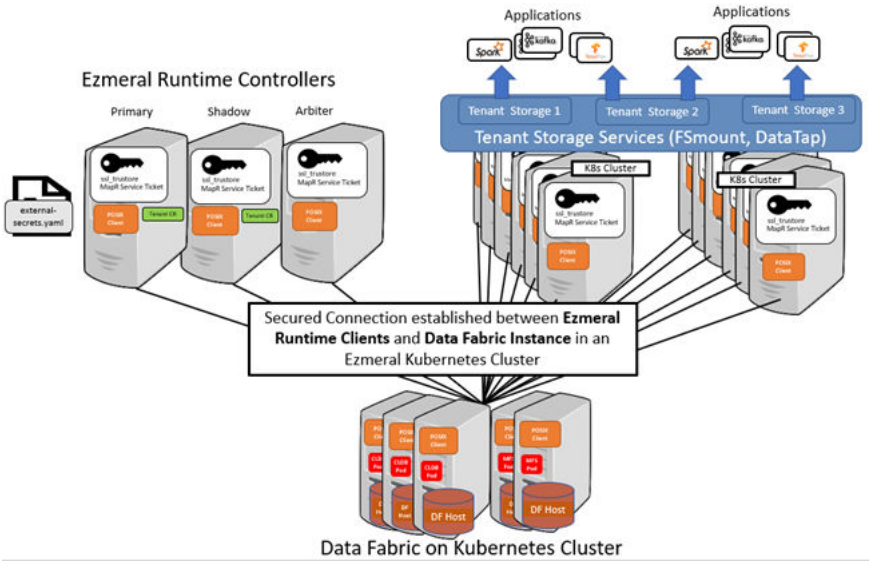


HPE Ezmeral Runtime Enterprise can connect to multiple Data Fabric storage deployments; however, only one Data Fabric deployment can be registered as tenant storage.

- If you have an HPE Ezmeral Data Fabric on Kubernetes cluster outside the HPE Ezmeral Runtime Enterprise, and if you want to configure HPE Ezmeral Data Fabric on Kubernetes as tenant storage, continue with this procedure.
- If you have already selected another Data Fabric instance for tenant/persistent storage, do not proceed with this procedure. Contact Hewlett Packard Enterprise Support if you want to use a different Data Fabric instance as tenant storage

This procedure may require 10 minutes or more per EPIC or Kubernetes host [Controller, Shadow Controller, Arbiter, Master, Worker, and so on], as the registration procedure configures and deploys Data Fabric client software on each host.

After Data Fabric registration is completed, the configuration will look as follows:



Procedure

1. **Preparation:**
   a) Have the Platform Administrator username and password ready.

b) Verify that all cluster nodes are up and running, and that the system is not in a degraded state.

c) Obtain the IP address of a cluster master node by executing the following command:

```
bdconfig --getk8shosts
```

This command returns a table with information for all nodes; you need the `IPADDR` value for any node on the relevant cluster that displays `K8S_MASTER` as `True`. **If the cluster has more than one master node, then you can pick IPADDR from any of the master nodes to be used as the Kubernetes Master Node IP in the next step. You do not need to repeat Step d. multiple times for each Master Node IP.**

d) Execute the information from the HPE Ezmeral Data Fabric on Kubernetes cluster and create a manifest file at `/opt/bluedata/tmp/<MASTER_NODE_IP>/dftenant-manifest`:

```
LOG_FILE_PATH=/tmp/<log_file>
MASTER_NODE_IP="<Kubernetes_Master_Node_IP_Address>" /opt/bluedata/
bundles/hpe-cp-*/startscript.sh --action prepare_dftenants
```

where:

`LOG_FILE_PATH` is an optional parameter that can help confirm or troubleshoot functionality. If this is not provided, then the file `/tmp/bds<datetime>.log` will be created.

The `MASTER_NODE_IP` was obtained in Step c. above.

The contents of the manifest created are:

```
CLDB_LIST="<comma-separated;FQDN_or_IP_address_for_each_CLDB_node>"
CLDB_PORT="<port_number_for_CLDB_service>"
SECURE="<true_or_false>" (Default is true)
CLUSTER_NAME="<name_of_DataFabric_cluster>"
REST_URL="<REST_API_URL_as_hostname:port>"
EXT_MAPR_MOUNT_DIR="<directory_in_mount_path_for_volumes>"
(Default is /exthcp)
TICKET_FILE_LOCATION="<path_to_ticket_for_HCP_admin>"
SSL_TRUSTSTORE_LOCATION="<path_to_ssl_truststore>"
HCP_ADMIN_USER="<name_of_HCP_admin_user>" (Default is mapr)
EXT_SECRETS_FILE_LOCATION="<path_to_external_secrets_file_for_Spark_cl
uster>"
FORCE_ERASE="<true_or_false" (Default is true)
RESTART_CNODE="<true_or_false"> (Default is true)
```

The `dftenant-manifest` is needed for cluster registration (next section).

a) Proceed to Registration.

2. **Configuration**

Deploy a Data Fabric client on all hosts by executing the following command on the primary controller host:

```
LOG_FILE_PATH=/tmp/<log_file>
MASTER_NODE_IP="<Kubernetes_Master_Node_IP_Address>" /opt/bluedata/
bundles/hpe-cp-*/startscript.sh --action configure_dftenants
```

The `ext_configure_dftenants` action deploys HPE Ezmeral Data Fabric client modules (such as the POSIX Client), on HPE Ezmeral Runtime Enterprise hosts.

3. **Registration**

To complete the registration procedure, initiate the `ext_register_dftenants` action, using the following command:

```
LOG_FILE_PATH=/tmp/<log_file>
MASTER_NODE_IP="<Kubernetes_Master_Node_IP_Address>" /opt/bluedata/
bundles/hpe-cp-*/startscript.sh --action register_dftenants
```

When prompted, enter the Site Administrator username and password. HPE Ezmeral Runtime Enterprise uses this information for REST API access to its management module.

The results of the `register_dftenants` action are the following:

- `register_dftenants` creates a volume, on the HPE Ezmeral Data Fabric on Kubernetes cluster, for each existing HPE Ezmeral Runtime Enterprise tenant. For a new tenant (created in the future), a tenant volume gets created automatically, on the HPE Ezmeral Data Fabric on Kubernetes cluster. The name of the volume in will be `tenant-<ID>`, where <ID> is the number of the tenant.
- The `register_dftenants` action reconfigures **Tenant Storage** to use the HPE Ezmeral Data Fabric on Kubernetes cluster, for all future tenants. And:
  - **TenantStorage** and **TenantShare** will be created for all existing tenants on the Data Fabric cluster.
  - For AI/ML tenants, the project repository will be changed to use a Data Fabric volume. However, data from the existing project repository will not be migrated.
  - Both **TenantShare** and **TenantStorage** will be available for all tenants.
- The `register_dftenants` action also reconfigures the following services:
  - Nagios, to track Data Fabric related client and mount services on the appropriate HPE Ezmeral Runtime Enterprise hosts.
  - WebHDFSs, to enable browser-based file system operations, such as upload, mkdir, and so on

The file systems on the per-tenant volumes on the Data Fabric cluster are mounted, by the Data Fabric client on each node, under `/opt/bluedata/mapr/mnt/<cluster_name>/<ext_mapr_mount_dir>/<tenant-id>/` , where:

- `<cluster_name>` is the name of the HPE Ezmeral Data Fabric on Kubernetes cluster.
- `<ext_mapr_mount_dir>` is specified in the `ext-dftenant-manifest`. See Step 1.c.
- `<tenant-id>` is the unique identifier for the relevant tenant.

Future Kubernetes clusters created in the HPE Ezmeral Runtime Enterprise will have persistent volumes located in:

```
/opt/bluedata/mapr/mnt/<datafabric_cluster_name>/<ext_mapr_mount_dir>/
```

The registered HPE Ezmeral Data Fabric on Kubernetes cluster will be the backing for Storage Classes of future Kubernetes Compute clusters, that are created in the HPE Ezmeral Runtime Enterprise The registration procedure does not modify the Storage Classes for Compute clusters, which existed before the registration.

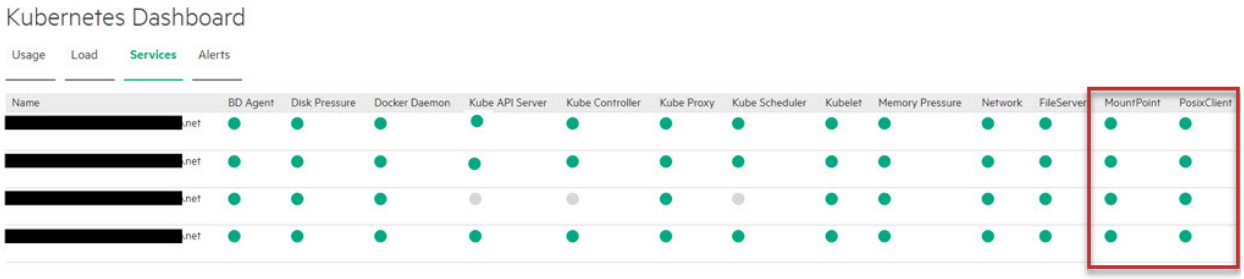
4. **Validation:**

   To confirm the success of the Registration, check the following

a) Check the output and/or logs of the `ext_configure_dftenants` and `ext_register_dftenants` actions.

b) On the HPE Ezmeral Runtime Enterprise Web UI, view the **Tenant Storage** tab on the **System Settings** page. Check that the information displayed on the screen is accurate for the HPE Ezmeral Data Fabric on Kubernetes cluster.

c) On the HPE Ezmeral Runtime Enterprise, view the **Kubernetes** and **EPIC** Dashboards, and check that the POSIX Client and Mount Path services on all hosts are in normal state.



d) On the HPE Ezmeral Runtime Enterprise Web UI, verify that, you are able to browse Tenant Storage on an existing tenant. If wanted, try uploading a file to a directory under Tenant Storage and reading the uploaded file. See Uploading and Downloading Files on page 367 for more details.

5. Proceed to Step 7: Fine-Tuning the Cluster of the procedure Creating a New Data Fabric Cluster.

HPE Ezmeral Data Fabric on Kubernetes Administration

You administer HPE Ezmeral Data Fabric on Kubernetes and Embedded Data Fabric as part of your HPE Ezmeral Runtime Enterprise environment. The external "bare metal" implementation of HPE Ezmeral Data Fabric is administered through its own tools and has its own documentation. (Not available in HPE Ezmeral Runtime Enterprise Essentials.)

The administration procedures in this section apply to the following implementations of HPE Ezmeral Data Fabric, except where noted:

- HPE Ezmeral Data Fabric on Kubernetes
- Embedded Data Fabric(supported on migrated HPE Ezmeral Runtime Enterprise deployments only)

For information about bare-metal implementations of HPE Ezmeral Data Fabric, see the HPE Ezmeral Data Fabric Documentation.

This feature is not available in HPE Ezmeral Runtime Enterprise Essentials.

About HPE Ezmeral Data Fabric on Kubernetes

NOTE: In this article, the term tenant refers to HPE Ezmeral Data Fabric tenants (formerly "MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

HPE Ezmeral Data Fabric on Kubernetes enables you to run HPE Ezmeral Data Fabric services on top of Kubernetes as a set of pods by:

- Creating Data Fabric clusters for storing data.
- Creating tenants for running Spark jobs inside pods.

This feature is not available in HPE Ezmeral Runtime Enterprise Essentials.

Creating a Data Fabric cluster and installing the tenant components runs HPE Ezmeral Data Fabric on Kubernetes as a fully native Kubernetes application. Deploying a Data Fabric cluster offers the following benefits:

- Independent and elastic storage and compute scaling.
- Simplified installation, upgrades, and scaling for easier "Day 0" and "Day 2" use.
- Pre-wired for data-intensive workloads, such as Spark and KubeFlow.
- Built for security, including user authentication and data encryption both at rest and in transit.

A tenant within a Kubernetes cluster is a workspace that contains compute runtime pods (such as Spark applications) that access and/or process data from the Data Fabric cluster, with no requirement for an internal Data Fabric cluster in the same Kubernetes environment. You can configure tenants to access data on external storage clusters that reside on bare-metal and other environments. Each tenant can connect to different storage clusters, but a single tenant cannot connect to multiple storage clusters.

The following installation scenarios are available:

- **Scenario 1:** Dedicated Data Fabric cluster. This scenario uses a dedicated Kubernetes cluster with the sole function of running HPE Ezmeral Data Fabric to provide data services. See Scenario 1, below.
- **Scenario 2:** Co-located Data Fabric cluster. This scenario co-locates with a Compute cluster. HPE Ezmeral Data Fabric runs alongside other workloads sharing a single Kubernetes cluster. See Scenario 2, below.

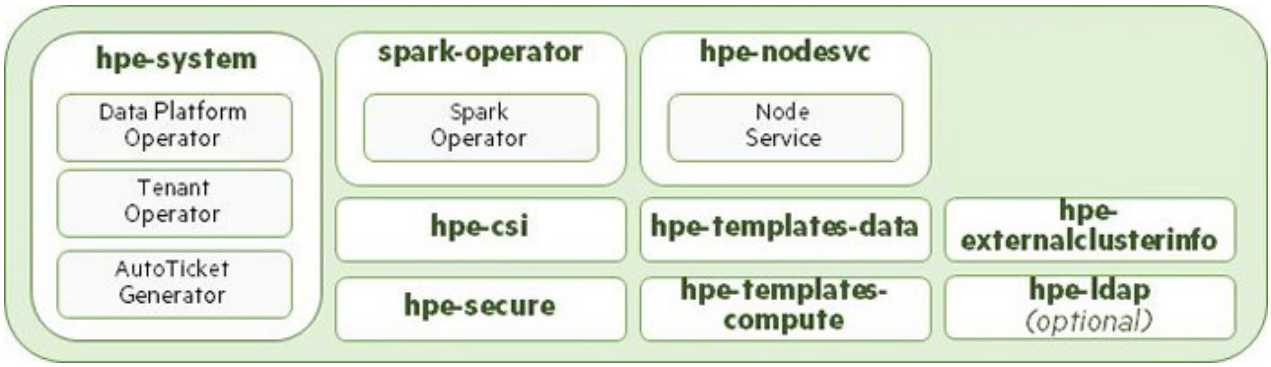
HPE Ezmeral Data Fabric on Kubernetes Configurations

You can configure HPE Ezmeral Data Fabric on Kubernetes with any of the following configurations:

- Requirements for HPE Ezmeral Data Fabric on Kubernetes — Recommended Configuration on page 595: Explained in this topic.
- Footprint-Optimized Configuration. See Requirements for HPE Ezmeral Data Fabric on Kubernetes — Footprint-Optimized Configurations on page 598

Namespaces Created for HPE Ezmeral Data Fabric

HPE Ezmeral Data Fabric uses namespaces to separate and isolate resources and applications. The following illustration shows the namespaces created as part of an installation by the bootstrap utility. This example shows the infrastructure namespaces that do not reflect any installed Data Fabric clusters or tenants.

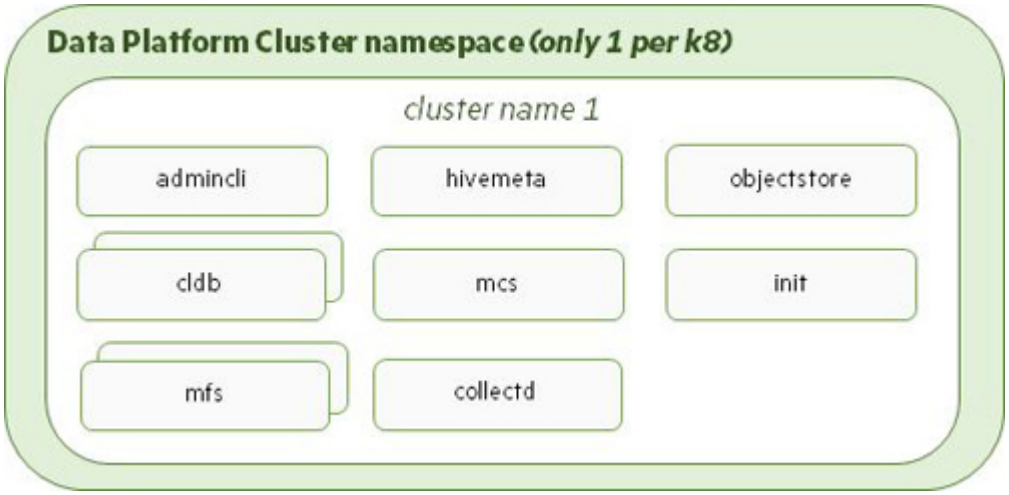


The bootstrap utility creates the following namespaces in each Kubernetes environment:

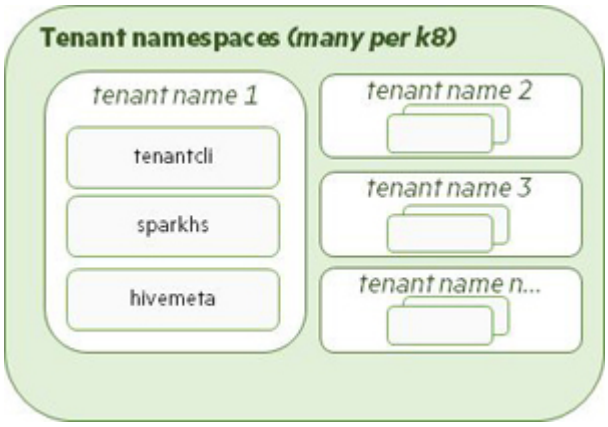
- **`hpe-system`** - This namespace is created for Data Fabric and Tenant operators, and the Autoticket Generator pod, in order to reduce the surface area for security vulnerabilities. Operators running in this namespace have privileges that Operators running in other namespaces (such as the Spark operator) do not have.
- **`spark-operator`** - This namespace is created for the Spark operator.
- **`hpe-csi`** - This namespace is created for running the HPE Ezmeral Data Fabric Container Storage Interface (CSI) version 0.3 pods, described here. CSI is an industry-standard interface that enables containerization of volume plug-ins that are agnostic to the underlying node for volume plug-in driver binaries. The CSI driver also provides POSIX support for an ObjectStore pod to connect to the Data Fabric cluster, as described in MapR Container Storage Interface Storage Plugin Overview and CSI Examples. (These links open external websites in a new browser tab/window.)
- **`hpe-nodesvc`** - This namespace contains a `daemonset` of `noderservice` pods that are responsible for labeling and annotating nodes for use with HPE Ezmeral Data Fabric.
- **`hpe-templates-data`** - This namespace contains a set of default config maps and secrets used by the pods created by the Data Fabric operator. These config maps contain the configuration files used by storage cluster services contained in these pods. For example, the `cldb-cm` config map contains the `cldb.conf` file used by a CLDB pod to read configuration settings.
- **`hpe-templates-compute`** - This namespace contains a set of default config maps and secrets used by the pods created by the Tenant operator. The config maps contain the various configuration files used by tenant services contained in these pods.
- **`hpe-externalclusterinfo`** - This namespace contains the information about external, existing storage clusters including information about the locations of various cluster components like CLDB and Zookeeper nodes, as well as secrets used by external tenants to connect to storage clusters.
- **`hpe-ldap`** - This optional namespace contains an `openLDAP` pod and service. During bootstrapping, if the authentication choice made is to use the `EXAMPLE openLDAP` service, then this will be the namespace in which it is generated.

The following namespaces are also used:

- **Data Fabric Cluster:** The Data Fabric operator creates the Data Fabric Cluster namespace and specifies the name of this namespace in the Data Fabric Custom Resource. The Data Fabric cluster runs in this name space, and pods for every required cluster component are created in this namespace. The following illustration shows the namespace and pods generated when the Data Fabric operator detects a new Data Fabric Custom Resource (CR) file created in the Kubernetes environment:



- **Tenant:** The Tenant operator creates tenant namespaces to run compute applications (e.g. Spark). Other tenant services, such as Hive Metastore, a Tenant CLI, and Spark History Server can also run in these namespaces. Multiple tenant namespaces can exist within the Data Fabric Cluster namespace. These namespaces are created when the Tenant operator detects new Tenant Custom Resource (CRs) file created in the Kubernetes environment.



Data Fabric Operators and Custom Resources

Native Kubernetes only has the notion of pods and pod lifecycles. Complex multi-tiered applications such HPE Ezmeral Data Fabric require higher-level management. Kubernetes operators are a standard Kubernetes design pattern that simplify starting complex Kubernetes applications and also manage the entire application lifecycle, including complex upgrades. For more information, see Operators (link opens an external website in a new browser tab/window). An Operator consists of two components:

- **Custom Resources (CRs):** See Custom Resources, below.
- **Controllers:** A Controller builds what is specified in the applicable CR.

HPE Ezmeral Data Fabric includes the following Kubernetes operators:

- **Data Fabric:** Creates Data Fabric clusters.
- **Tenant:** Creates tenant namespaces for running Spark applications. The tenant references either:
  - The internal Data Fabric cluster residing in the same Kubernetes environment.
  - A different storage cluster external to the Kubernetes cluster.
- **Spark Operator:** Starts Spark jobs inside existing tenants. A Spark job creates a Spark cluster on the fly. A Spark driver pod launches a set of Spark executors that execute the specified job.

Custom Resources

A Custom Resource (CR) Kubernetes component is a valid instance of a Custom Resource Definition (CRD) that adds new types to Kubernetes via a YAML file that contains settings for customized application installation in the Kubernetes environment, as described here (link opens an external website in a new browser tab/window). The following sample CRs are available:

- Data Fabric
- Tenant

You may either:

- Customize and deploy the included sample CRs for Data Fabric clusters and tenants.
- Create and deploy your own custom CRs.

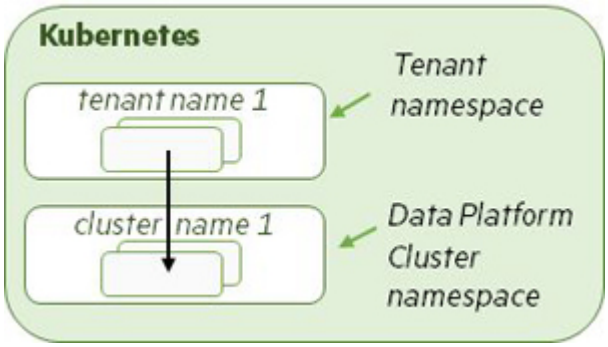
Deploying Data Fabric Clusters and Tenants

There are two ways to deploy Data Fabric Clusters:

- Scenario 1
- Scenario 2

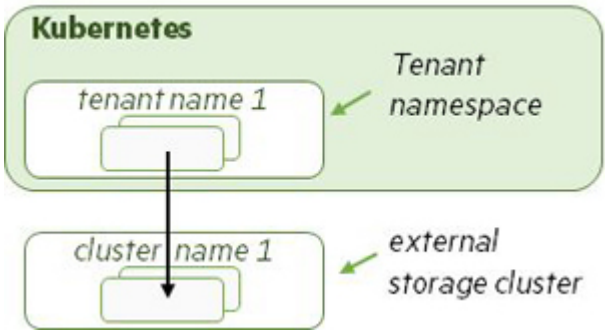
Scenario 1: Tenants Using Internal Data Fabric Clusters

You can deploy both a Data Fabric cluster and multiple Tenants in the same Kubernetes environment within HPE Ezmeral Runtime Enterprise. The following illustration depicts both a Data Fabric cluster and a tenant for running Spark applications created in the same Kubernetes environment. Pods are created for both the Data Fabric cluster and tenant components, and tenant containers access data in the internal Data Fabric cluster.



Scenario 2: Tenants Using External Storage Clusters

Having an available installation of an HPE Ezmeral Data Fabric storage cluster allows you to create a Kubernetes tenant that uses this external storage. You can also configure Spark applications from an HPE Ezmeral Runtime Enterprise tenant to connect and access the external storage cluster, such as HPE Ezmeral Data Fabric on bare metal. A Tenant namespace and various support files are created to hold this external connectivity information. The following illustration depicts applications in a tenant connecting to and accessing data in a storage cluster located either on premises or in another supported environment.



A Tenant can connect to only one storage cluster, but different Tenants can connect to different storage clusters. And multiple Tenants can connect to the same storage cluster.

NOTE: Each tenant can only connect to one storage cluster, but multiple tenants can connect to either different storage clusters or a storage cluster that is shared by multiple tenants. External storage clusters must be visible from the pods. You can test this by opening a shell to a running pod and then pinging the nodes in the external storage cluster.

NOTE: You may manually create one or more tenants, as described in Manually Creating a New HPE Ezmeral Data Fabric Tenant on page 703.

Container Storage Interface

HPE Ezmeral Data Fabric incorporates an optional Container Storage Interface (CSI) Storage Plugin that exposes HPE Ezmeral Data Fabric to containerized workloads. If the CSI is installed, the bootstrap utility offers the option either to install or not to install CSI. CSI is installed by default, but HPE Ezmeral Data Fabric can operate without it. See Using the CSI on page 634 and Container Storage Interface (CSI) Storage Plugin Overview (link opens an external website in a new browser tab/window).

Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only)

Describes the requirements for HPE Ezmeral Data Fabric on Kubernetes, which is only supported for non-production environments. For all production requirements, recommendation is to use HPE Ezmeral Data Fabric on Bare Metal.

There are two types of configurations of HPE Ezmeral Data Fabric on Kubernetes:

Recommended Configuration	Beginning HPE Ezmeral Runtime Enterprise 5.5.0, recommendation is to use HPE Ezmeral Data Fabric on Bare Metal for use in production environments.
Footprint-Optimized Configuration	Footprint-optimized configurations implement HPE Ezmeral Data Fabric on Kubernetes on a smaller set of nodes. Footprint-optimized configurations are intended for non-production environments, such as development, testing, and proof-of-concept demonstrations. Footprint-optimized configurations are not supported on production environments. For information about the different footprint-optimized configurations for non-production environments, see Requirements for HPE Ezmeral Data Fabric on Kubernetes — Footprint-Optimized Configurations on page 598.

Requirements for HPE Ezmeral Data Fabric on Kubernetes — Recommended Configuration

Describes the minimum system requirements for using HPE Ezmeral Data Fabric on Kubernetes in HPE Ezmeral Runtime Enterprise.

NOTE: In this article, the term tenant refers to HPE Ezmeral Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

You can configure HPE Ezmeral Data Fabric on Kubernetes with any of the following configurations:

- Recommended Configuration: Explained in this topic.
- Footprint-Optimized Configuration. See Requirements for HPE Ezmeral Data Fabric on Kubernetes — Footprint-Optimized Configurations on page 598

Recommended Configuration

Table

Configuration	Recommended Minimum CPU Cores	Recommended Minimum RAM
3 Masters + 5 Workers **NOTE:** 5 Worker nodes are minimum requirement. HPE recommends using 6 or more Worker nodes as it increases the High Availability (HA) of the cluster.		
Masters (Requirements are the same as the general requirements for Master Hosts)	4 per node (12 cores total)	32GB per node (96GB total)

last-updated: May 06, 2024

Table (Continued)

Configuration	Recommended Minimum CPU Cores	Recommended Minimum RAM
Workers	32 per node (160 cores total)	64GB per node (320GB total)
1 or more Compute nodes (required if running compute jobs on the cluster.)	32 per node (32 cores total)	64 GB per node (64GB per node)

The minimum deployment that includes **HPE Ezmeral Data Fabric on Kubernetes** is a single Kubernetes cluster.

The following diagram shows the hosts in a minimum production deployment of HPE Ezmeral Runtime Enterprise with **HPE Ezmeral Data Fabric on Kubernetes**. The minimum deployment is a single Kubernetes cluster.

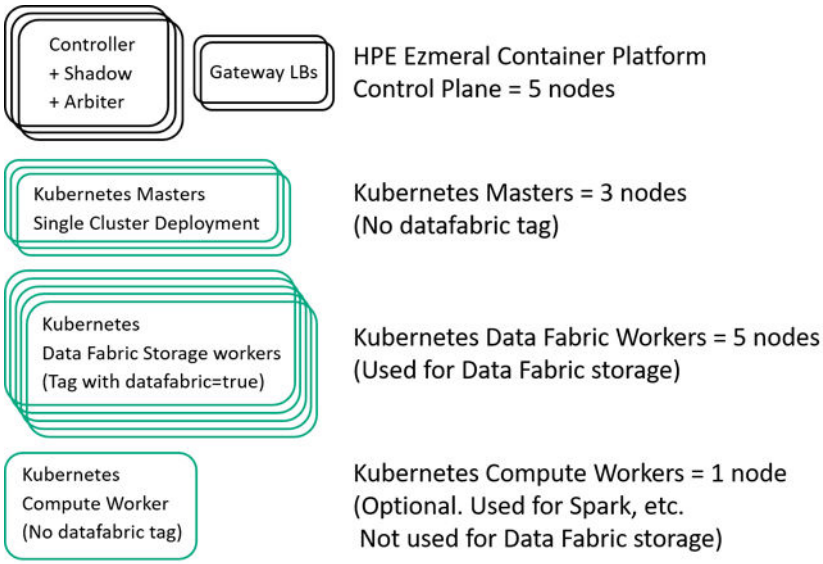


Figure 8: Minimum Production Deployment of HPE Ezmeral Data Fabric on Kubernetes

In contrast, if the HPE Ezmeral Runtime Enterprise deployment has separate clusters for **HPE Ezmeral Data Fabric on Kubernetes** and for compute, Kubernetes Masters and Workers are required for each Kubernetes cluster. The following diagram shows an example of a multiple cluster deployment.

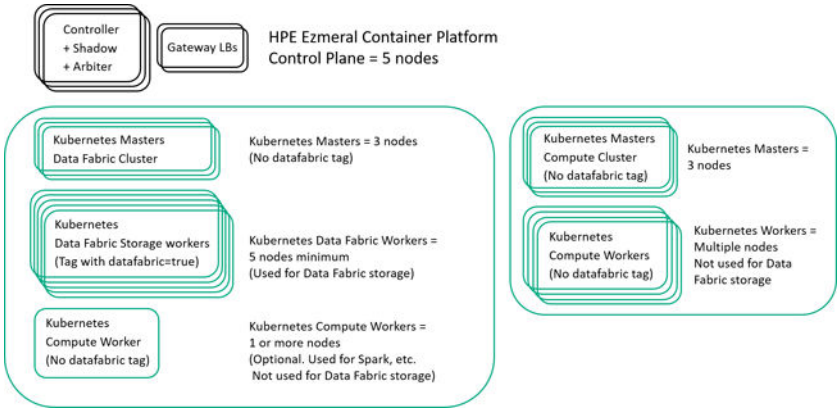


Figure 9: Separate Data Fabric and Compute Clusters

In production deployments of HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes has the following minimum requirements:

- **Master Nodes:**

  The number of master nodes depends on the number of Kubernetes clusters in this deployment of HPE Ezmeral Runtime Enterprise. Each Kubernetes cluster requires a minimum of three (3) Kubernetes Master nodes for HA.

  For example, if this is a single-cluster Kubernetes deployment of HPE Ezmeral Runtime Enterprise, a minimum total of three (3) Kubernetes Master nodes are required.

  You must select an odd number of Master nodes in order to have a quorum (e.g. 3, 5, 7, etc.). Hewlett Packard Enterprise recommends selecting three or more Master nodes to provide High Availability protection for the Data Fabric cluster.

  Master nodes orchestrate Kubernetes cluster and are not used for data storage. Master nodes cannot use the `Datafabric` tag.

- **Worker Nodes:** The minimum is five Worker nodes, each with the `Datafabric` tag set to `true` or `yes`. These Worker nodes are used for Data Fabric storage.

  Worker nodes that will be used for data storage must have the `Datafabric` tag set to `yes` (`Datafabric=yes`). The `Datafabric` tag may also be set to `YES`, `true`, or `TRUE`.

  NOTE: You must have at least five (5) Data Fabric Worker nodes for storage (tagged with `Datafabric=true` or `yes`) in a Data Fabric cluster to ensure High Availability protection. You may reduce resource requirements by turning off monitoring services, if they are not needed. See User-Configurable Configuration Parameters.

  If you want to install application add-ons, such as Spark, in the same cluster, you need at least one (1) additional Worker node that does **not** have the `Datafabric` tag. This Worker node is called a Compute node within a Data Fabric cluster.

  For example, if you want to use an application such as Spark, Airflow, or Kubeflow in the same cluster as the `Datafabric` nodes, you will need at least one Compute node, for a total of six (6) Worker nodes at minimum. Tenants using the node only require enough compute resources to run Spark service containers.

Ephemeral Storage Requirements

Each host must have a minimum of 500GB of ephemeral storage available to the OS.

Persistent Storage Requirements

For persistent storage, minimum requirement is One disk (hard disk, SSD, or NVMe drive) per Data Fabric node. However, HPE recommends having three or more disks on each Data Fabric node, to allow at least one full storage pool per node, for production environments.

If a Data Fabric node has single disk, and the disk fails, data recovery may be slower and impact the performance, as replication copies are stored on other nodes. Hence, HPErecommends multiple disks in a Data Fabric node, which allows a full storage pool in a node.

NVMe support on EC2 instances depends on the release of HPE Ezmeral Runtime Enterprise (see On-Premises, Hybrid, and Multi-Cloud Deployments on page 102).

Other Requirements

- **Container Runtime:** Docker.
- **Kubernetes:** See the Kubernetes Version Requirements on page 832.

- **SELinux Support:** Nodes that are part of HPE Ezmeral Data Fabric on Kubernetes require SELinux to be run in "permissive" (or "disabled") mode. If you need to run SELinux in "enforcing" mode, contact your Hewlett Packard Enterprise support representative.
- **CSI Driver:** For usage considerations, see Using the CSI on page 634. To deploy the CSI driver on a Kubernetes cluster, the Kubernetes cluster must allow privileged pods. Shared Docker mounts must be allowed for mount propagation. See the Kubernetes CSI documentation and OS Configurations for Shared Mounts (links open external websites in a new browser tab/window).

Requirements for HPE Ezmeral Data Fabric on Kubernetes — Footprint-Optimized Configurations
Describes available footprint-optimized configurations of HPE Ezmeral Data Fabric on Kubernetes and the requirements for deploying a footprint-optimized configuration in non-production environments.

Footprint-Optimized Configurations

Footprint-optimized configurations implement HPE Ezmeral Data Fabric on Kubernetes on a smaller set of nodes and support a subset of the features (services) of the high-performance production configuration. Footprint-optimized configurations are intended for non-production environments such as for development, testing, and proof-of-concept demonstrations.

Footprint-optimized configurations are not supported for use in production environments.

There are two supported footprint-optimized configurations of HPE Ezmeral Data Fabric on Kubernetes:

Combined Masters-Workers Configuration	The smallest configuration has three nodes, each of which performs Kubernetes master functions and Data Fabric storage functions. Optionally, you can add Data Fabric worker nodes or compute nodes to this configuration. To run compute jobs, at least one compute node is required. For information about the requirements for this configuration, see Combined Masters-Workers Configuration on page 598.
Dedicated Control Plane Configuration	This configuration is similar to the high-performance production configuration. This configuration has three dedicated control plane (master) Kubernetes nodes, and a minimum of three Data Fabric worker nodes. Optionally, you can add Data Fabric worker nodes or compute nodes to this configuration. To run compute jobs, at least one compute node is required. For information about the requirements for this configuration, see Dedicated Control Plane Configuration on page 599.

Combined Masters-Workers Configuration

This configuration is the smallest configuration. The smallest configuration is a total of three nodes, each of which performs Kubernetes master functions and Data Fabric storage functions. Optionally, you can add Data Fabric worker nodes or compute notes to this configuration. To run compute jobs, at least one compute node is required.

This configuration is not supported for use in production environments, even if you add worker nodes. Migration of this configuration to the high-performance production configuration is not supported.

Table

Configuration	Recommended Minimum CPU Cores	Recommended Minimum RAM
3 Masters/Workers		

Table (Continued)

Configuration	Recommended Minimum CPU Cores	Recommended Minimum RAM
Masters/Workers All nodes tagged `Datafabric=yes`	32 per node (96 cores total)	64GB per node (192GB total)
One Compute Node Required if running compute jobs on the cluster. No `Datafabric` tag.	32 per node (32 cores total)	64 GB per node (64GB total)

See Requirements for HPE Ezmeral Data Fabric on Kubernetes — Recommended Configuration on page 595 for information about the following:

- Ephemeral Storage Requirements
- Persistent Storage Requirements
- Other Requirements, such as CSI drivers

Dedicated Control Plane Configuration

This configuration is similar to the high-performance production configuration. This configuration has three dedicated control plane (master) Kubernetes nodes, and a minimum of three Data Fabric worker nodes. Optionally, you can add Data Fabric worker nodes or compute nodes to this configuration. To run compute jobs, at least one compute node is required.

With enough additional worker nodes, for a total of five or more worker nodes, this configuration can be converted into the high-performance production configuration. Changes to the CR are required. Contact your Hewlett Packard Enterprise representative.

Table

Configuration	Recommended Minimum CPU Cores	Recommended Minimum RAM
3 Masters + 3 Workers		
Masters Requirements are the same as the general requirements for Kubernetes Master hosts. No `Datafabric` tag.	4 per node (12 cores total)	32GB per node (96GB total)
Workers All nodes tagged `Datafabric=yes`.	32 per node (96 cores total)	64GB per node (192GB total)
1 Compute Node Required if running compute jobs on the cluster. No `Datafabric` tag.	32 per node (32 cores total)	64 GB per node (64GB total)

See Requirements for HPE Ezmeral Data Fabric on Kubernetes — Recommended Configuration on page 595 for information about the following:

- Ephemeral Storage Requirements
- Persistent Storage Requirements

- Other Requirements, such as CSI drivers

Limitations of Footprint-Optimized Configurations

- No Hive Metastore
- No Monitoring or Metrics Capabilities - Monitoring capabilities are not available as Grafana and openTSDB are not available. However, Metrics and Monitoring pods may be added to the cluster if enough resources are available.

Footprint-Optimized Configuration CR

Footprint-optimized configurations use a different CR than the high-performance production configurations. The CR used for footprint-optimized configurations configure pods differently and omit monitoring and metrics services. For a sample CR, see the following:

https://github.com/HPEEzmeral/df-on-k8s/blob/main/examples/p1.5.0/3node/3node_core_objectstore_gateway.yaml

Data Fabric Cluster Administrator Username and Password

This topic defines the HPE Ezmeral Data Fabric cluster administrator and provides information about the default username (`mapr`) and password for the Data Fabric cluster administrator in HPE Ezmeral Runtime Enterprise deployments.

The Data Fabric cluster administrator is the user that HPE Ezmeral Data Fabric cluster services run as on each node.

In HPE Ezmeral Runtime Enterprise:

- The default username of the Data Fabric cluster administrator is `mapr`.
- The default password of the Data Fabric cluster administrator is generated automatically.

  To get the password, enter the following command:

  ```
  kubectl get secret system -n <data-fabric-cluster-namespace> -o yaml |
  grep MAPR_PASSWORD | head -1 | awk '{print $2}' | base64 --decode
  ```

HPE Ezmeral Data Fabric Control System (MCS) is available for HPE Ezmeral Data Fabric on Kubernetes.

The port for MCS is not fixed though in HPE Ezmeral Data Fabric. The following example demonstrates how to determine the port. First, enter the following commands in your terminal:

```
kubectl get services -n <data-fabric-cluster-namespace> | grep mcs
```

The output of this example is:

```
mcs-svc NodePort 10.105.167.223 <none> 8443:30452/TCP
```

The port number to use in this example is: 30452

Then use an IP address of one of the HPE Ezmeral Data Fabric cluster nodes for the url: https://16.0.14.189:30452 (this would be the login URL in this example).

Login with the username: `mapr`

NOTE:

For information about passwords for monitoring services such as Kibana and Grafana, see Managing HPE Ezmeral Data Fabric on Kubernetes on page 627.

Using Self-Signed Certificates with the Data Fabric Cluster

NOTE: In this article, the term tenant refers to Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

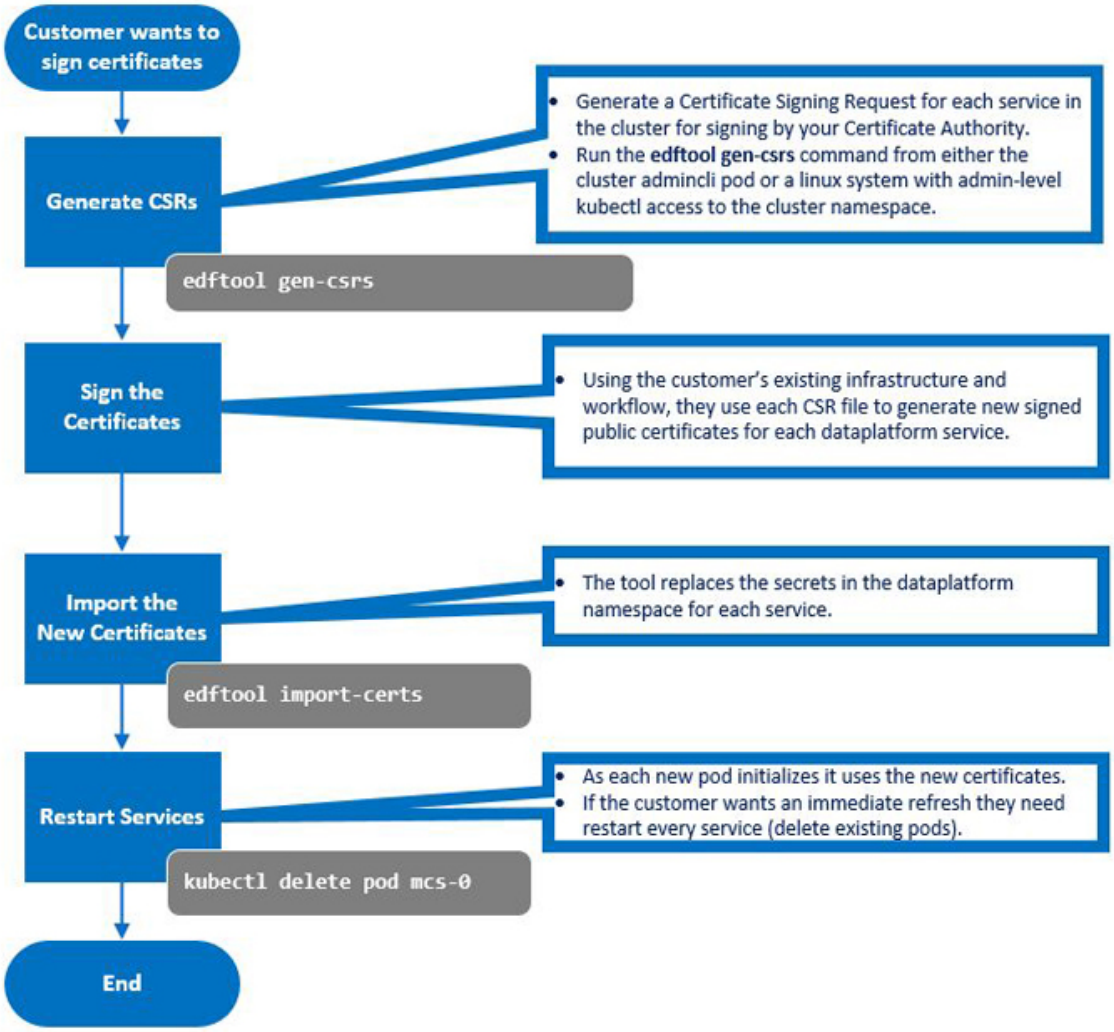
An initialization script generates private keys and self-signed certificates when a Data Fabric cluster is created. The `edftool` includes several commands that allow these certificates to be self-signed using the following workflow:

1. Generate Certificate Signing Requests (CSRs) for the Data Fabric cluster services.
2. Custom-sign the certificates.
3. Import the signed public certificates back into the cluster and insert them into the `ssl_keystore` and `ssl_truststore` for each service.
4. Restart each service.

Workflow

An initialization pod generates self-signed certificates for the services supported by the Data Fabric when you create the Data Fabric and apply the example CRs. This information is saved as secrets in the `data-platform` namespace. Each service pulls its specific private and public keys from those secrets during cluster deployment.

Self-signed certificates may not meet environment security requirements, such as requiring certificates signed by a Certificate Authority (CA). The following diagram illustrates how you can generate new public certificates, sign them, and import them into the cluster:



Use the `edftool` to generate CSRs for each service:

NOTE: The `edftool` uses a `.cnf` template file to generate the CSRs. By default, this template is stored in `/opt/mapr/kubernetes/template.cnf` . If needed, you can specify a different template file by executing the `edftool gen-csrs` command.

NOTE: Certificates cannot be updated individually. They must be updated for all the Data Fabric services at once.

1. Execute the following command on either the Kubernetes cluster or the client where the `edftool` is installed:

```
kubectl exec -it -n <pod-namespace> admincli-0 -- /bin/bash
```

2. Execute the following command to allow the files to be created:

```
cd /tmp
```

3. Execute the following command:

```
edftool gen-csrs
```

The tool examines the current secrets, looks up each service, and determines the current signing status. For example:

```
mtomas@mark-vbox2: ~/.../bootstrap
Available Commands:
  cluster-trust Setup trust between two clusters
  export-certs  Export the public certs of each service
  export-keys   Export the private keys of each service
  gen-csrs      Generate certificate signing requests for each service
  help          Help about any command
  import-certs  Import new certs (newly signed?)

Flags:
  -h, --help   help for edftool

Use "edftool [command] --help" for more information about a command.
sh-4.4$ edftool gen-csrs
The list below are the subjects for signing requests.   Each request will use the CNF template file /opt
/mapr/kubernetes/template.cnf for configuration:
--'admincli-svc' subject=CN = *.admincli-svc.heisenberg.svc.cluster.local
--'collectd-svc' subject=CN = *.collectd-svc.heisenberg.svc.cluster.local
--'elasticsearch-svc' subject=CN = *.elasticsearch-svc.heisenberg.svc.cluster.local
--'hivemeta-svc' subject=CN = *.hivemeta-svc.heisenberg.svc.cluster.local
--'httpfs-svc' subject=CN = *.httpfs-svc.heisenberg.svc.cluster.local
--'kibana-svc' subject=CN = *.kibana-svc.heisenberg.svc.cluster.local
--'mcs-svc' subject=CN = *.maprwebserver-svc.heisenberg.svc.cluster.local
--'grafana-svc' subject=CN = *.grafana-svc.heisenberg.svc.cluster.local
--'opentsdb-svc' subject=CN = *.opentsdb-svc.heisenberg.svc.cluster.local
--'zk-svc' subject=CN = *.zk-svc.heisenberg.svc.cluster.local
--'cldb-svc' subject=CN = *.cldb-svc.heisenberg.svc.cluster.local
Would you like to proceed? y/n:
```

4. Enter yes at the following prompt to continue the process:

```
Would you like to proceed? y/n:
```

5. Execute the `ls` command to verify CSR generation. For example:

```
mtomas@mark-vbox2: ~/.../bootstrap
--'cldb-svc' subject=CN = *.cldb-svc.heisenberg.svc.cluster.local
Would you like to proceed? y/n:
y
success.  next step is to use your CA to sign the CSRs and then update them with import-certs.sh-4.4$
sh-4.4$
sh-4.4$ ls
hcpdpt-20200709T1531.log     heisenberg-elasticsearch-svc.csr  heisenberg-mcs-svc.csr
hcpdpt-20200709T1608.log     heisenberg-grafana-svc.csr        heisenberg-opentsdb-svc.csr
heisenberg-admincli-svc.csr  heisenberg-hivemeta-svc.csr       heisenberg-zk-svc.csr
heisenberg-cldb-svc.csr      heisenberg-httpfs-svc.csr         k8_patch_cluster_heisenberg.sh
heisenberg-collectd-svc.csr  heisenberg-kibana-svc.csr
sh-4.4$ ls -l
total 76
-rw-r--r-- 1 mapr mapr  8883 Jul  9 15:32 hcpdpt-20200709T1531.log
-rw-r--r-- 1 mapr mapr     0 Jul  9 16:08 hcpdpt-20200709T1608.log
-rw-r--r-- 1 mapr mapr  1281 Jul  9 16:08 heisenberg-admincli-svc.csr
-rw-r--r-- 1 mapr mapr  1257 Jul  9 16:08 heisenberg-cldb-svc.csr
-rw-r--r-- 1 mapr mapr  1281 Jul  9 16:08 heisenberg-collectd-svc.csr
-rw-r--r-- 1 mapr mapr  1310 Jul  9 16:08 heisenberg-elasticsearch-svc.csr
-rw-r--r-- 1 mapr mapr  1273 Jul  9 16:08 heisenberg-grafana-svc.csr
-rw-r--r-- 1 mapr mapr  1281 Jul  9 16:08 heisenberg-hivemeta-svc.csr
-rw-r--r-- 1 mapr mapr  1269 Jul  9 16:08 heisenberg-httpfs-svc.csr
-rw-r--r-- 1 mapr mapr  1269 Jul  9 16:08 heisenberg-kibana-svc.csr
-rw-r--r-- 1 mapr mapr  1265 Jul  9 16:08 heisenberg-mcs-svc.csr
-rw-r--r-- 1 mapr mapr  1281 Jul  9 16:08 heisenberg-opentsdb-svc.csr
-rw-r--r-- 1 mapr mapr  1249 Jul  9 16:08 heisenberg-zk-svc.csr
-rwxr-xr-x 1 mapr mapr 18877 Jul  9 15:32 k8_patch_cluster_heisenberg.sh
sh-4.4$
```

6. Use the `.csr` files and your company-specific processes to generate new, signed public certificates for each Data Fabric service using SCP or another tool to export each `.csr` file to your CA-signing server. This process generates `.crt` files (certificates) for each `.csr` file.

7. Use SCP or another tool to move the resulting `.crt` files to the working directory on the Data Fabric cluster.

8. Import the new certificates and replace the secrets in the `Data Fabric` namespace for each service by executing the following command:

```
edftool import-certs
```

9. Enter `yes` at the following prompt to continue the process:

```
Would you like to proceed? y/n:
```

10. Restart the CLDB and Zookeeper pods by executing the `edf update cluster` command in the `admincli-0` pod in `/usr/bin`. For example:

```
kubectl exec -it admincli-0 -n <pod-namespace> -- /bin/bash
edf update cluster
```

11. For the other services, delete the pod for the service to cause a replacement pod to start, initialize, and use the certificates by executing the following command:

```
kubectl delete pod <service-pod-name>
```

For example:

```
kubectl delete pod mcs-0
```

Related reference

Command Reference: edf update cluster on page 718
The `edf update cluster` command updates components in HPE Ezmeral Data Fabric on Kubernetes clusters.

Command Reference: edf shutdown cluster on page 718
The `edf shutdown cluster` command shuts down core components in Kubernetes HPE Ezmeral Data Fabric clusters and prevents them from resuming operations.

Command Reference: edf startup {pause | resume} on page 719
The `edf startup pause` command flags core HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS, such that they will to enter into a nonfunctional state when they restart. The pods resume their startup sequence only after the `edf startup resume` command is executed.

External KMIP Keystore Support

Both bare-metal and container-based HPE Ezmeral Data Fabric implementations include external KMIP Key Store (see External KMIP Keystore Overview; link opens in a new browser tab/window), but there are some key differences:

- Bare-metal configurations configure KMIP using either `configure.sh` with the new HSM options (for a fresh installation) or the `mrhsm` utility (to modify or upgrade existing installations). The `configure.sh` script calls the `mrhsm` utility behind the scenes and will not generate CLDB and DARE keys when HSM is used. The encrypted KMIP configuration is created and then copied to all the CLDB and Zookeeper nodes in the cluster.
- Kubernetes pods are ephemeral, and anything written to the `${MAPR_HOME}/conf/tokens` directory does not persist after the pods are destroyed. The encrypted KMIP configuration must therefore be stored as a Kubernetes secret and mounted as a volume on the CLDB and Zookeeper nodes. The encrypted KMIP configuration is stored in CRs with appropriate labels so that Kubernetes will know where to mount the volume.

KMIP Deployment Workflow

To deploy the External KMIP Key Store with HPE Ezmeral Data Fabric:

1. Configure `kubectl` to point to your Kubernetes environment.
2. Bootstrap your Kubernetes environment by executing the following commands:

```
$ cd bootstrap
$ ./bootstrap.sh install
```

3. Set up the KMIP-enabled HSM by following the instructions in the appropriate KMIP Integration Guide (link opens in a new browser tab/window). You must obtain the HSM IP addresses and port numbers. You must also download the CA certificate, client certificate, and private client key. The currently supported HSMs are:
   - Utimaco ESKM
   - Vormetric DSM
   - Gemalto SafeNet Keysecure
4. Use `kubectl` to install and configure container permissions on the Mac or Linux machine. You need permissions to access the container images in `gcr.io/mapr-252711`. Click here for information about the `gcloud` command (link opens an external website in a new browser tab/window).
5. Run the HSM setup script to configure KMIP for the cluster:

```
$ cd tools
$ ./setup-hsm.sh <cluster-name>
```

6. Deploy the generated CR from the `tools/` directory to push the KMIP secret to the `hpe-secure` namespace via the interface (not manually):

```
$ kubectl apply -f hsm_config/<cluster-name>/
hsmconfig-<cluster-name>.yaml
```

7. Deploy the `dataplatform` CR:

```
$ kubectl apply -f <path-to-dataplatform-CR>
```

8. The remaining steps follow the standard deployment workflow.

Example Setup

The following sample session illustrates a two-node Gemalto SafeNet KeySecure KMIP configuration for a cluster named `demo.cluster.com`:

```
tools % ./setup-hsm.sh demo.cluster.com
Configuring HSM for cluster demo.cluster.com
latest: Pulling from mapr-252711/hsmsetup-6.2.0
Digest:
sha256:22451ee67f8d15c083410d288298a90ec9cf138f0456ece60f559752f0521fc9
Status: Image is up to date for gcr.io/mapr-252711/hsmsetup-6.2.0:latest
gcr.io/mapr-252711/hsmsetup-6.2.0:latest
Configuring HSM for demo.cluster.com. When you are done, type "exit" to
return to the rest of the HSM setup.
[root@88df33691bff mapr]# /opt/mapr/server/mrhsm info -slots
Slot 0
    Slot info:
        Description:          MapRHSM slot ID 0x0
        Manufacturer ID:      HPE MapR-HSM
        Token present:        yes
    Token info:
        Manufacturer ID:      HPE MapR-HSM
        Model:                MapRHSM
        Serial number:
        Initialized:          no
        User PIN initialized: no
        Label:
[root@88df33691bff mapr]# /opt/mapr/server/mrhsm init -label "Gemalto
SafeNet KeySecure"
Enter SO PIN (4-255 characters): ****
Please reenter SO PIN: ****
[root@88df33691bff mapr]# /opt/mapr/server/mrhsm set -ip
"10.10.30.129,10.10.30.182" -cacert /opt/mapr/hsmsetup/
CA.pem -clientcert /opt/mapr/hsmsetup/client.pem -clientkey /opt/mapr/
hsmsetup/key.pem
Enter SO PIN: ****
[root@88df33691bff mapr]# /opt/mapr/server/mrhsm enable -dare
Enter SO PIN: ****
Obtained cluster name demo.cluster.com from mapr-clusters.conf
Enabling MapR HSM on cluster demo.cluster.com
Successfully generated Core KEK, UUID
A39276162C3BFCFD972AF9ED354CE53A8351932EA6772B5790939BC339E8C139
SHA-256 checksum for Core KEK is
5122D496285E1A768D201727521C028E938E9606D54C26529ECF07AF8307B789
Successfully generated Common KEK, UUID
1C7A23C33D6797EFC35040A4C01646E4350C2C986FD24678C47C8EAAAA0C8FA7
SHA-256 checksum for Common KEK is
5E579DE1029486FE1A09190C77374AE1C034ADD0A22F2F6C5439854A68D8980D
No CLDB key found on host
SHA-256 checksum for CLDB key is
3AC9E80BFEA952FAEC181D837D5B83E5C38A712A913742D35A4B9E0AEA17177F
Successfully set encrypted CLDB key in KMIP configuration
SHA-256 checksum for DARE key is
672B1545C131B7B1C76E5776A18E2BBEECFE318534AAE1892975AD79D077C28B
Successfully set encrypted DARE key in KMIP configuration
##############################################################################
##
Copy the entire contents of the KMIP token directory /opt/mapr/conf/tokens
to
all CLDB and Zookeeper nodes. All files in /opt/mapr/conf/tokens must be
owned
by the mapr user and mapr group.
##############################################################################
```

```
##
[root@88df33691bff mapr]# exit
exit
Removing  container
Continue to create HSM secret? (y/n) [y]:
Generating Kubernetes custom resource for KMIP configuration ...

The KMIP configuration generated for this cluster are available at: /Users/
testuser/builds/private-kubernetes/tools/./hsm_config/demo.cluster.com/
hsmconfig-demo.cluster.com.yaml
Please copy them to a machine where you can run the following command:
  kubectl apply -f /Users/testuser/builds/private-kubernetes/tools/./
hsm_config/demo.cluster.com/hsmconfig-demo.cluster.com.yaml
tools % kubectl apply -f /Users/testuser/builds/private-kubernetes/tools/./
hsm_config/demo.cluster.com/hsmconfig-demo.cluster.com.yaml
secret/hsmconfig-demo.cluster.com created
```

Example Notes

In this example:

- The `hsm_config` directory is created in the same directory as the `setup-hsm.sh` script, which is in the `tools/` directory.
- The customized container used to run the `mrhsm` utility to create the encrypted KMIP configuration contains an `/opt/mapr/hsmsetup` directory that is mapped to the `hsm_config/<cluster-name>` directory. Certificates and private keys required for the KMIP configuration can be copied to this directory to make it accessible to the container.
- If you exit the container prior to creating an enabled KMIP configuration, then the configuration will be saved in `hsm_config/<cluster-name>/tokens` where you can access it in future sessions. If you need to set new CA or client certificates or private keys, or re-key the core KEK, then you can also use the `setup-hsm.sh` script.
- The Kubernetes secret for the KMIP configuration is in a compressed `.tar` archive that contains the contents of the `${MAPR_HOME]/conf/tokens` directory.
- For new installations, the `init` container does not generate the CLDB and DARE keys if it finds the HSM secret in the `hpe-secure` namespace. If the HSM secret exists in `hpe-secure`, then the `init` container pulls it from the `hpe-secure` namespace and puts it in the `dataplatform` namespace. The CLDB and Zookeeper pods retrieve this secret and extract the `.tar` archive to the `${MAPR_HOME}/conf/tokens` directory. This allows the CLDB, MFS, and ZooKeeper services to retrieve the CLDB and DARE keys from the HSM upon startup.
- The YAML custom resource used by `kubectl` to push the Kubernetes secret to the `hpe-secure` namespace is stored in `hsm_config/<cluster-name>/hsmconfig-<cluster-name>.yaml`.

This example shows the sample contents of the `hsm_config` directory for two clusters named `new.cluster.com` and `demo.cluster.com`:

```
tools % find hsm_config -print
hsm_config
hsm_config/new.cluster.com
hsm_config/new.cluster.com/tokens
hsm_config/new.cluster.com/tokens/06e32ec9-9a40-3951-1fd2-23ae6332be79
hsm_config/new.cluster.com/tokens/06e32ec9-9a40-3951-1fd2-23ae6332be79/
token.lock
hsm_config/new.cluster.com/tokens/06e32ec9-9a40-3951-1fd2-23ae6332be79/
token.object
```

```
hsm_config/new.cluster.com/tokens/06e32ec9-9a40-3951-1fd2-23ae6332be79/
generation
hsm_config/new.cluster.com/tokens/mrhsm.conf
hsm_config/demo.cluster.com
hsm_config/demo.cluster.com/hsmconfig-demo.cluster.com.tgz
hsm_config/demo.cluster.com/key.pem
hsm_config/demo.cluster.com/README
hsm_config/demo.cluster.com/client.pem
hsm_config/demo.cluster.com/CA.pem
hsm_config/demo.cluster.com/req.pem
hsm_config/demo.cluster.com/hsmconfig-demo.cluster.com.yaml
hsm_config/demo.cluster.com/tokens
hsm_config/demo.cluster.com/tokens/5f2356db-1201-859e-b316-66a512b25fbb
hsm_config/demo.cluster.com/tokens/5f2356db-1201-859e-b316-66a512b25fbb/
token.lock
hsm_config/demo.cluster.com/tokens/5f2356db-1201-859e-b316-66a512b25fbb/
token.object
hsm_config/demo.cluster.com/tokens/5f2356db-1201-859e-b316-66a512b25fbb/
generation
hsm_config/demo.cluster.com/tokens/mrhsm.conf
```

Rekeying and Modifying the KMIP Configuration

You can use the `setup-hsm.sh` script and `mrhsm rekey` command to re-key the Core KEK for an existing configuration. After exiting the `setup-hsm.sh` script, modify the custom resource definition to change the namespace from `hpe-secure` to `dataplatform`:

```
$ more hsmconfig-demo.cluster.com.yaml
apiVersion: v1
kind: Secret
metadata:
  name: hsmconfig-demo.cluster.com
  namespace: demo-cluster.com
type: Opaque
data:
  MAPR_KMIP:     "H4sIAIJHfV8AA+2cXainW13HJ0uyJ6TsQrKQJkrqwjOu9xehi/
WqRzuj6BHMGxnn7JPmOWfwOJKCVEYUdCWUl5ZIlBQYYRB001XSRTcVCUFgFxlYBhlhYBf1+f5n+8
JonrOPzpzy7AUzs/d////Ps9bv5fuy1rPn9q13nj3x7ldcuZfDGJNjvHr6N6XTv8aFO/
+ej6vWZ59d8NHlq8YG49
…
```

NOTE: In the above example, change the `namespace` from `demo-cluster.com` to the actual cluster namespace.

Next, execute the following command to update the secret in the `dataplatform` namespace:

```
$ kubectl apply -f hsmconfig-demo.cluster.com.yaml
```

if you need to use the new key immediately, then restart the CLDB and Zookeeper pods. However, the original Core KEK will still be available and you can wait until the next CLDB and Zookeeper restart.

If you change the HSM configuration (such as changing the HSM server IP addresses or replacing an expired client certificate), then the original HSM servers may no longer be accessible or the client certificates may no longer be valid. Restart the CLDB and Zookeeper pods to update the HSM secret in the `dataplatform` namespace.

Configuring KMIP for an Existing Cluster

To configure KMIP for an existing cluster that already has the CLDB and DARE master key:

1. Configure the KMIP-enabled HSM per the instructions in the appropriate KMIP Integration Guide (link opens in a new browser tab/window; Utimaco ESKM, Vormetric DSM, and Gemalto SafeNet KeySecure are currently supported).

2. Obtain the HSM IP addresses and KMIP port number.

3. Download the CA certificate, client certificate, and private client key.

4. Use `kubectl` to install and configure container permissions on the Mac or Linux machine. You need permissions to access the container images in `gcr.io/mapr-252711`. Click here for information about the `gcloud` command (link opens an external website in a new browser tab/window).

5. Run the HSM setup script to configure KMIP for the cluster:

```
$ cd tools
$ ./setup-hsm.sh <cluster-name>
```

6. As described in Example Notes, copy the client credentials such as the CA certificate, client certificate, and client private key to the shared `hsm_config/<cluster-name>` folder to allow access by the `mrhsm` utility when running `mrhsm set`.

7. Download the CLDB key `${MAPR_HOME}/conf/cldb.key` and DARE master key (if applicable) `${MAPR_HOME}/conf/dare.master.key` from one of the CLDB pods in the cluster, and then place then in a location that is accessible to the container that is started within `setup-hsm.sh`. Be sure to also back up the CLDB and DARE master keys in a safe place.

   For example, if you are in the `tools/` directory where the `setup-hsm.sh` script resides:

```
$ kubectl cp -n demo.cluster.com cldb-0:/opt/mapr/conf/cldb.key
hsm_config/chyelindarenohsm/cldb.key
$ kubectl cp -n demo.cluster.com cldb-0:/opt/mapr/conf/dare.master.key
hsm_config/demo.cluster.com/dare.master.key
```

8. Copy the CLDB key and DARE master key (if applicable) that you downloaded in Step 7 to `/opt/mapr/conf` in the container. For example, within the container:

```
[root@95b57c45e5d6 conf]# ls ../hsmsetup
CA.pem            cldb.key    dare.master.key   tokens
Client.pem   key.pem
[root@95b57c45e5d6 conf]# cp ../hsmsetup/cldb.key .
[root@95b57c45e5d6 conf]# cp ../hsmsetup/dare.master.key .
[root@95b57c45e5d6 conf]# pwd
/opt/mapr/conf
[root@95b57c45e5d6 conf]# ls
cldb.key  daemon.conf  dare.master.key    mapr-clusters.conf
maprhsm.conf  tokens
[root@95b57c45e5d6 conf]#
```

9. Execute the `/opt/mapr/server/mrhsm` utility in the container within the `setup-hsm.sh` script.. If the `mrhsm` utility finds existing CLDB and DARE master keys in `/opt/mapr/conf`, then the `mrhsm enable` command will import the existing CLDB and DARE master keys into the HSM instead of generating new keys in the HSM. Be sure to:

   - Execute `mrhsm enable -dare` if the DARE master key exists
   - Omit the `-dare` option if DARE is not enabled.

10. Modify the generated CR in the `tools/hsm_config/<cluster-name>` directory to change the namespace from `hpe-secure` to the cluster name. For example, if the cluster name is `demo.cluster.com`, then you would modify the CR as follows to push the KMIP secret to the `demo.cluster.com` namespace:

```
metadata:
  name: hsmconfig-demo.cluster.com
  namespace: demo.cluster.com
```

11. Deploy the generated CR from the `tools/` directory to push the KMIP secret to the Data Fabric cluster namespace. For example, if the cluster name is `demo.cluster.com`:

```
$ cd hsm_config/demo.cluster.com
$ kubectl apply -f hsmconfig-demo.cluster.com.yaml
```

12. Verify that the secret is in the cluster namespace. For example, for the `demo.cluster.com` cluster:

```
% kubectl get secrets -n demo.cluster.com | grep hsmconfig
hsmconfig-demo.cluster.com     Opaque                 1      18m
```

NOTE: The KMIP secret will be retrieved from the cluster secret namespace and configured into the `${MAPR_HOME}/conf/tokens` directory during the startup procedure so that the KMIP feature will be enabled the next time the CLDB and ZK pods restart.

13. Restart the CLDB and Zookeeper pods by executing the `edf update cluster` command in the `admincli-0` pod in `/usr/bin`. For example:

```
kubectl exec -it admincli-0 -n <pod-namespace> /bin/bash
edf update cluster
```

14. Verify that each of the CLDB and ZK pods are now HSM-enabled by executing the `mrhsm info -kmip` command on each pod and verifying that the `Enabled` field is set to `Yes`. For example, for pod `cldb-0`:

```
% kubectl exec cldb-0 -n demo.cluster.com -- /opt/mapr/server/mrhsm
info -kmip
Displaying information for KMIP token with serial ff989992b11f9ed3
KMIP Configuration Version 1
-----------------------------
CLDB:
...
Enabled             : Yes
```

Related reference

Command Reference: edf update cluster on page 718
The `edf update cluster` command updates components in HPE Ezmeral Data Fabric on Kubernetes clusters.

Command Reference: edf shutdown cluster on page 718
The `edf shutdown cluster` command shuts down core components in Kubernetes HPE Ezmeral Data Fabric clusters and prevents them from resuming operations.

Command Reference: edf startup {pause | resume} on page 719

The `edf startup pause` command flags core HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS, such that they will to enter into a nonfunctional state when they restart. The pods resume their startup sequence only after the `edf startup resume` command is executed.

Creating a New Data Fabric Cluster

Use this procedure when creating a cluster that implements HPE Ezmeral Data Fabric on Kubernetes.

Important:

- If you want Compute (non-Data Fabric) Kubernetes clusters to use HPE Ezmeral Data Fabric on Kubernetes storage, then you must create aData Fabric cluster using the instructions in this article before creating a Compute Kubernetes cluster, as described in Creating a New Kubernetes Cluster. Istio Service Mesh is not supported on HPE Ezmeral Data Fabric on Kubernetes clusters.
- Before you create a HPE Ezmeral Data Fabric on Kubernetes cluster, ensure that you add Data Fabric nodes as described in Kubernetes Data Fabric Node Installation Overview.

  When you create or expand an HPE Ezmeral Data Fabric on Kubernetes cluster, you cannot reuse existing Kubernetes hosts that were added to HPE Ezmeral Runtime Enterprise without specifying the `Datafabric` tag. However, you can decommission the hosts from the HPE Ezmeral Runtime Enterprise as described in Decommissioning/Deleting a Kubernetes Host on page 555, then add the `Datafabric` tag to the hosts, and then select the hosts when creating or expanding the Data Fabric cluster.
- Ensure that the Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595 are met.
- You cannot shrink a Data Fabric cluster.
- If you are using an air-gap configuration (see Kubernetes Air-Gap Requirements and Air Gap Tab), then you must configure these settings before creating any Kubernetes clusters. Otherwise, you will need to delete the clusters and then create them again.

This process consists of the following steps:

- Before Creating the Cluster
- Step 1: Host Configurations
- Step 2: Cluster Configuration
- Step 3: Authentication
- Step 4: Application Configurations
- Step 5: Summary
- Step 6: HPE Ezmeral Data Fabric as Tenant Storage
- Step 7: Fine-Tuning the Cluster

Before Creating the Cluster

Before creating the HPE Ezmeral Data Fabric on Kubernetes cluster, you must first install the nodes that will be part of the cluster, as described in Kubernetes Data Fabric Node Installation Overview on page 531.

When adding the nodes, consider the following.

For Feature Optimized Configuration:

- **Master node:** Do not set the **Datafabric** tag.

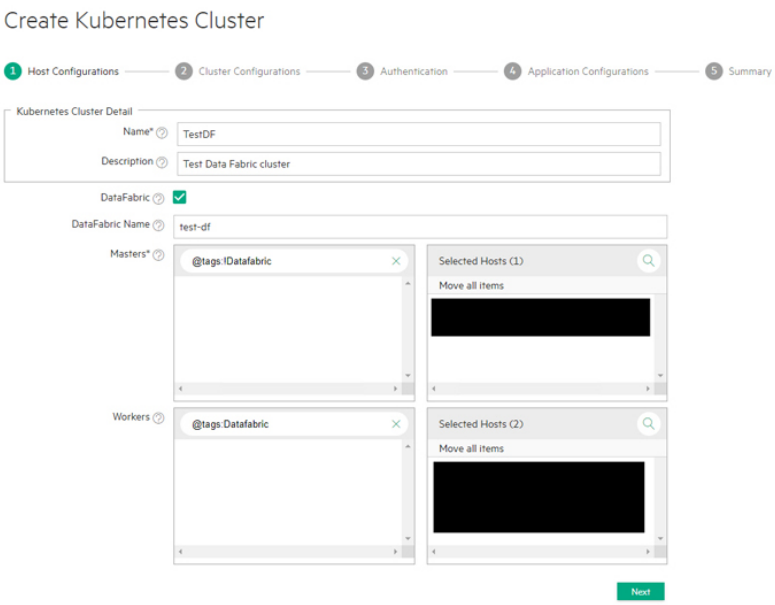
- **Worker node:** Be sure to set the **Datafabric** tag to either `yes`, `YES`, `true`, or `TRUE`.

For Footprint Optimized Configuration:

- Hosts that have **Datafabric** tag set to `yes`, `YES`, `true`, or `TRUE` act as Master Nodes.
- Based on the **Datafabric** settings, on the **Step 1: Host Configurations** screen, Master and worker nodes will be listed in the **Available Hosts** column.

Step 1: Host Configurations

Clicking the **Create Kubernetes Cluster** button in the **Kubernetes Clusters** screen opens the **Step 1: Host Configurations** screen.



To begin creating a new Data Fabric cluster:

1. Enter a name for the new cluster in the **Name** field.
2. Enter a brief description of the new Data Fabric cluster in the **Description** field.
3. Ensure that the **DataFabric** check box is checked.

   CAUTION: Failure to select the **DataFabric** check box will result in creating a new Kubernetes compute-only cluster, as described in Creating a New Kubernetes Cluster on page 463.

4. Enter a name for the Data Fabric cluster in the **DataFabric Name** field. This name is also used as the name of the Custom Resource and namespace that represent the Data Fabric. The name must be:
   - RFC 1123-compliant.
   - At least four characters long and no more than 63 characters long.
   - Can include alphanumeric characters (lower-case letters only) and hyphens.
   - Cannot begin or end with a hyphen.
5. Select the master nodes.

   The Master nodes cannot use the `Datafabric` tag.

You must select an odd number of Master nodes in order to have a quorum (e.g. 3, 5, 7, etc.). Hewlett Packard Enterprise recommends selecting three or more Master nodes to provide High Availability protection for the Data Fabric cluster.

You can search for a node by name, tag, etc. by entering your desired search term in the field and then clicking the **Search** icon (magnifying glass). Nodes that do not have the `Datafabric` tag set to `yes` are automatically selected and displayed. You can also search for nodes by clicking the **Search** icon (magnifying glass) above any of the four cells in the **Hosts** table and then typing any portion of the hostname. The list of nodes automatically refreshes as you type.

To select a master node:

a. In the **Masters** section of the **Hosts** table, hover the mouse over a node in the **Available** column.

A right arrow appears.

b. Click the arrow.

The selected node moves from the **Available Hosts** column to the **Selected Hosts** column.

To deselect a node, you may hover the mouse over a selected node and then click the left arrow to move it back to the **Available Hosts** column.

By default, a taint is placed on the Master nodes that prevents them from being able to run pods. You must untaint these nodes if you want them to be available to run pods, as described here (link opens an external web site in a new browser tab/window).

6. Select the Worker nodes.

- Worker nodes that will be used for data storage must have the `Datafabric=yes` tag. This tag may also be set to `YES`, `true`, or `TRUE`.
- Worker nodes that do not have the `Datafabric` tag will be used for compute functions.
- For information about the minimum and recommended number of Worker nodes, see Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.

You can search for a node by name, tag, and so forth by entering your desired search term in the field and then clicking the **Search** icon (magnifying glass). Nodes that have the `Datafabric` tag set to `yes` are automatically selected and displayed. You can also search for nodes by clicking the **Search** icon (magnifying glass) above any of the four cells in the **Hosts** table and then typing any portion of the hostname. The list of nodes automatically refreshes as you type.

To select a worker node:

a. In the **Workers** section of the **Hosts** table, hover the mouse over a node in the **Available** column.

A right arrow appears.

b. Click the arrow.

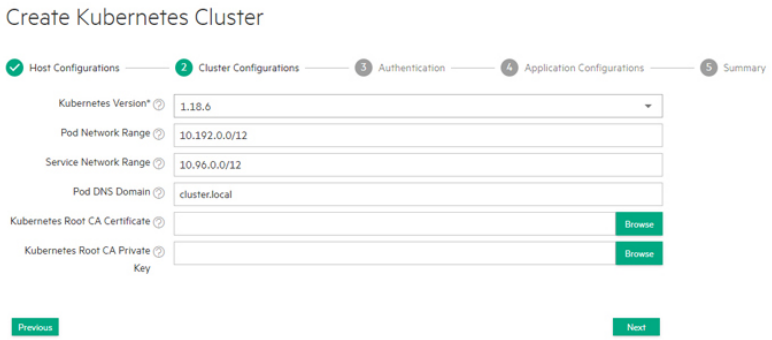
The selected node moves from the **Available Hosts** column to the **Selected Hosts** column.

To deselect a node, hover the mouse over a selected node and then click the left arrow to move it back to the **Available Hosts** column.

7. Click **Next**.

Step 2: Cluster Configuration

The **Step 2: Cluster Configuration** screen appears.



1. Use the **Kubernetes Version** pull-down menu to select the version of Kubernetes to install on the new cluster. You may upgrade this version later, as described in Upgrading Kubernetes. By default, the three most recent versions of Kubernetes recommended by the CNCF (Cloud Native Computing Foundation) are provided. This allows you to use the most recent Kubernetes version available here (link opens an external web site in a new browser tab/window). Specific versions of upstream can be onboarded via a manifest stored in a local repository (see Air Gap Tab).

2. Enter the network range and mask) to use for the pods in this cluster in the **Port Network Range** field. The Calico and Flannel Kubernetes CNI plug ins are pre-installed and configured, and defaults are provided for the Pod CIDR that is within a private range. You need only update these parameters if they conflict with other ranges that are already in use. Check here for additional information (link opens an external web site in a new browser tab/window).

3. Enter the network range and mask to use for the endpoint services in this cluster in the **Service Network Range** field. The Calico and Flannel Kubernetes CNI plugins are pre-installed and configured, defaults are provided for the Pod CIDR that is within a private range. You need only update these parameters if they conflict with other ranges that are already in use. Check the **Choosing IP Address** section here for additional information (link opens an external web site in a new browser tab/window).

4. Enter the DNS domain to use for the service endpoints in this cluster in the **Pod DNS Domain** field.

5. Enter the path to the Kubernetes root CA certificate in the **Kubernetes Root CA Certificate** field. This is the certificate authority that Kubernetes will use to generate the certificates needed for various Kubernetes components, such as `etcd` and `auth proxy`/`front-proxy`. Clicking the **Browse** button opens a standard **Open** dialog that allows you to navigate to and select the desired file.

6. Enter the path to the Kubernetes root CA private key in the **Kubernetes Root CA Private Key** field. This is the private key portion of the root CA certificate. Clicking the **Browse** button opens a standard **Open** dialog that allows you to navigate to and select the desired file.

7. If you are satisfied with your changes, then click **Next** to proceed. Alternatively, you can click **Previous** to return to the **Step 1: Host Configurations** screen.

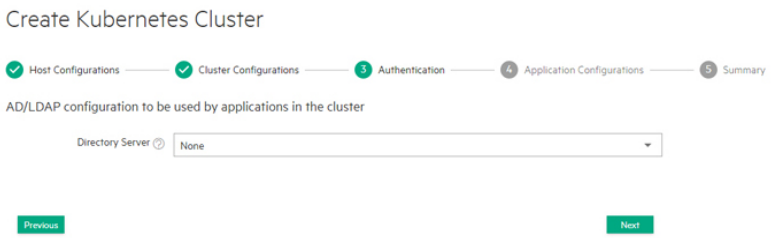
Step 3: Authentication

The **Step 3: Authentication** screen appears. You may either:

- Use the global HPE Ezmeral Runtime Enterprise user authentication.
- Specify user authentication options on a per-Kubernetes-cluster basis.

This is where you enter the LDAP/AD user authentication configuration that will be used by the applications running in this cluster. Any information entered in this screen is posted as a secret in the cluster. For Data Fabric clusters, the Data Fabric bootstrapper finds the Kubernetes secret and specifies

the user-authentication parameters in the form of an `sssd.conf` file. The prompt responses to the bootstrapper indicate an "Existing LDAP option".

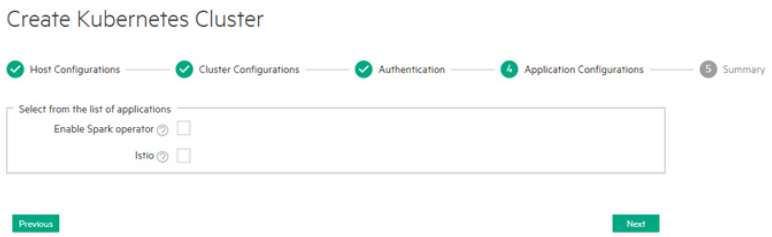


1. You may either:
   - Click **Next** to use the platform-wide authentication settings.
   - Click the **Copy from Platform Authentication** button to copy the platform-level LDAP/AD authentication to this Data Fabric cluster for further editing, as described in Configuring User Authentication Options.
   - Manually enter authentication settings that will only apply to this Data Fabric cluster, as described in Configuring User Authentication Options.
2. Click **Next** to proceed.

Step 4: Application Configurations

The **Step 4: Application Configurations** screen appears with a list of available HPE Ezmeral Runtime Enterprise applications that are not specifically related to a Data Fabric cluster. The list of available add-on applications may vary from that shown below.

Do not select the Istio application when creating or editing a Data Fabric cluster. Istio Service Mesh is not supported on HPE Ezmeral Data Fabric on Kubernetes clusters.



1. Verify that all of the hosts in the cluster meet the host requirements and the cumulative requirements for all the applications that will be selected, and then select the check boxes for the applications.

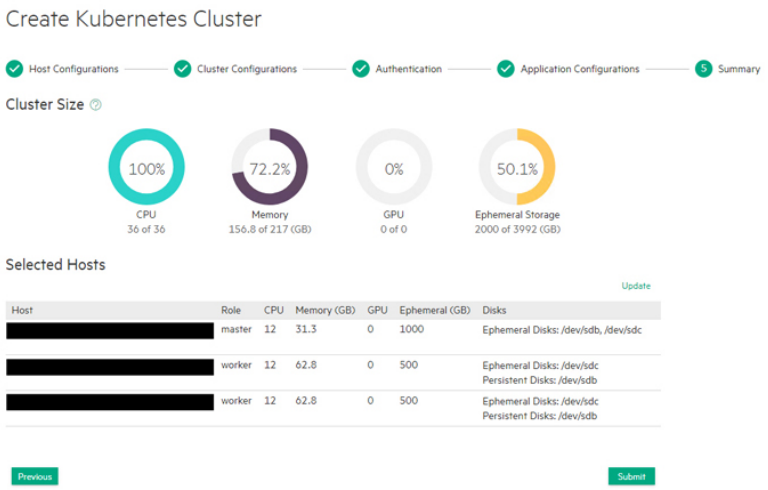
   Not all applications are approriate for all clusters. For example, Do not select the Istio application when creating or editing a Data Fabric cluster. Istio Service Mesh is not supported on HPE Ezmeral Data Fabric on Kubernetes clusters.

   For information about host requirements, see Kubernetes Host Requirements.

   For information about add-on applications, see Add-ons Overview. Requirements are cumulative; for example, if you add two applications, then all the hosts in the cluster must meet the combined requirements of both applications.
2. Review your application selections, and then click **Next** to proceed. Alternatively, you can click **Previous** to return to the **Step 3: Authentication** screen.

Step 5: Summary

The **Step 5: Summary** screen appears.



Review the summary of resources to be assigned to this cluster, and then either click **Submit** to finish creating the new Data Fabric cluster or click **Previous** to return to the **Step 4: Application Configurations** screen. If you need to configure the Open Policy Agent, then see OPA Gatekeeper Policy Configuration on page 469.

HPE Ezmeral Runtime Enterprise validates the make-up of the intended Data Fabric cluster and, if validation passes, proceeds to create the cluster. The HPE Ezmeral Runtime Enterprise bootstrap add-on bootstraps the Data Fabric cluster. If this bootstrapping succeeds, then HPE Ezmeral Runtime Enterprise automatically prepares and applies the Data Fabric CR and then waits for the Data Fabric cluster pods and pod-sets to come up.

- The Data Fabric cluster is ready when its status appears as **Ready** in the **Kubernetes Clusters** screen. See The Kubernetes Cluster Screen.
- If the Data Fabric cluster is not ready within the `picasso_cldb_wakeup_timeout` period (default is 1500 seconds), then HPE Ezmeral Runtime Enterprise will stop cluster creation with an error. You can configure the timeout period as described in User-Configurable Data Fabric Cluster Parameters on page 710.

NOTE: Be sure to register the HPE Ezmeral Data Fabric on Kubernetes cluster as Tenant Storage before creating Compute Kubernetes clusters. Skipping the registration step will require additional manual steps if you already have existing Compute Kubernetes clusters.

Step 6: HPE Ezmeral Data Fabric as Tenant Storage

To select from tenant storage options, and to register, see HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579.

Step 7: Fine-Tuning the Cluster

If desired, you can manually fine-tune the Data Fabric cluster by modifying the cluster Custom Resource file (CR), as described in Manually Creating/Editing a Data Fabric cluster on page 694.

Expanding a Data Fabric Cluster

This procedure describes how to expand an HPE Ezmeral Data Fabric on Kubernetes cluster deployed on HPE Ezmeral Runtime Enterprise.

CAUTION: You cannot shrink HPE Ezmeral Data Fabric on Kubernetes clusters.

NOTE:

You can only expand HPE Ezmeral Data Fabric on Kubernetes clusters that were created with HPE Ezmeral Runtime Enterprise version 5.2 or later.

When you create or expand an HPE Ezmeral Data Fabric on Kubernetes cluster, you cannot reuse existing Kubernetes hosts that were added to HPE Ezmeral Runtime Enterprise without specifying the `Datafabric` tag. However, you can decommission the hosts from the HPE Ezmeral Runtime Enterprise as described in Decommissioning/Deleting a Kubernetes Host on page 555, then add the `Datafabric` tag to the hosts, and then select the hosts when creating or expanding the Data Fabric cluster.

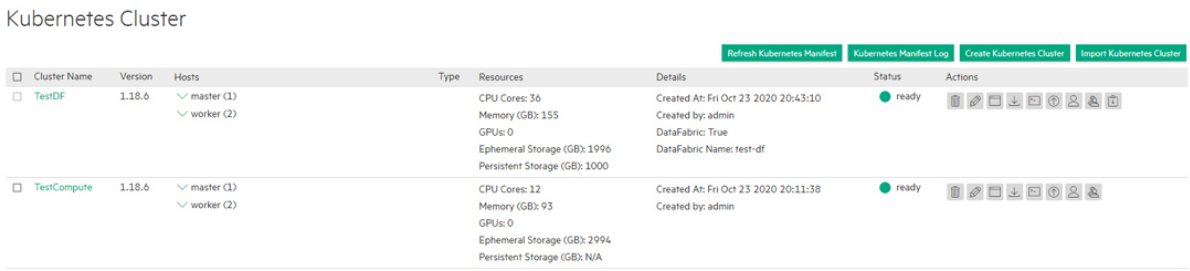
NOTE:

You cannot modify the cluster HA status during expansion.

- Changing the value of the `DisableHA` key is ignored.
- You cannot add pods during cluster expansion if doing so would change the cluster HA status.

To expand a Data Fabric cluster:

1. Add the Data Fabric nodes that will be used to expand the Data Fabric cluster, as described in Kubernetes Data Fabric Node Installation Overview on page 531.
2. Access the **Kubernetes Clusters** screen, as described in The Kubernetes Clusters Screen on page 457.



3. Click the **Edit** icon (pencil) for the Data Fabric cluster.

   The **Step 1: Hosts Configuration** screen appears. The **Datafabric** check box is grayed out and cannot be edited.

4. In the **Masters** row of the **Hosts** table, hover the mouse over a node in the **Available** column.

   A right arrow appears.

5. Click the arrow.

   The selected node moves from the **Available Hosts** column to the **Selected Hosts** column. To deselect a node, you may hover the mouse over a selected node and then click the left arrow to move it back to the **Available Hosts** column. You must select an odd number of Master nodes in order to have a quorum (e.g. 3, 5, 7, etc.). Hewlett Packard Enterprise recommends selecting three or more Master nodes to provide High Availability protection for the Data Fabric cluster. Master nodes cannot use the `Datafabric` tag.

   By default, a taint is placed on the Master nodes that prevents them from being able to run pods. You must untaint these nodes if you want them, to be available to run pods, as described here (link opens an external web site in a new browser tab/window).

6. Repeat Steps 4 and 5 for the Worker nodes. You may add or remove as many Worker nodes as needed to this cluster.

   NOTE:

   You may select Master and Worker nodes with or without the `Datafabric=yes` tag.

   - Nodes tagged `Datafabric=yes` are used for Data Fabric storage.
   - Nodes that are not tagged `Datafabric=yes` are included in the Data Fabric cluster, but are used for compute functions only.

   NOTE: You can search for nodes by clicking the **Search** icon (magnifying glass) above any of the four cells in the **Hosts** table and then typing any portion of the hostname. The list of nodes automatically refreshes as you type.

7. Proceed with Steps 2 and onward of Editing an Existing Kubernetes Cluster on page 480.

   NOTE:

   You cannot change add-ons when expanding a Data Fabric cluster.

HPE Ezmeral Runtime Enterprise validates the make-up of the intended cluster expansion in a manner similar to that when the Data Fabric cluster was created. If validation succeeds, then the existing Data Fabric CR is retrieved and updated to reflect the expansion. A Data Fabric cluster expansion supports an increased `failurecount` for CLDB and/or Zookeeper pod sets provided that the value of the `disableha` key in the Data Fabric CR does not change.

HPE Ezmeral Runtime Enterprise automatically determines whether to add CLDB and Zookeeper nodes and MFS groups. New MFS pods, if any, will be added in new MFS groups that are appended to the list of existing MFS groups already present in the Data Fabric CR.

The web interface expansion process completes quickly, and the **Kubernetes Clusters** screen may show the expanded Data Fabric cluster in the **Ready** state, but the newly-added pods may still be coming up.

If desired, you can manually fine-tune the Data Fabric cluster by modifying the cluster Custom Resource file (CR), as described in Manually Creating/Editing a Data Fabric cluster on page 694.

Shutting Down a Data Fabric Cluster

This procedure performs an orderly shut down of Data Fabric clusters that implement HPE Ezmeral Data Fabric on Kubernetes. This procedure does not apply to bare-metal HPE Ezmeral Data Fabric clusters. This procedure does not shut down the entire HPE Ezmeral Runtime Enterprise.

Prerequisites

- Run the edf check/report commands to verify that the cluster is fully functional. All issues must be resolved before shutting down the cluster.
- Ensure that all tenant services that read or write to this Data Cluster and all tenant applications, such as Spark, are stopped.
- Ensure that no Data Fabric operations, including file replication or mirroring operations, are in progress.

You must have access to the admin CLI pod (default name: `admincli-0`).

About this task

When you use the `edf shutdown cluster` command, pods are shut down and are rebooted, but the pods are put into a wait state immediately after the reboot, which prevents the Data Fabric cluster from becoming operational. When you are ready to resume operations, you can use the `edf startup resume` command to start the Data Fabric cluster.

NOTE:

- This procedure applies to the Data Fabric cluster only. This procedure does not shut down the entire HPE Ezmeral Runtime Enterprise.
- This procedure does not apply to bare-metal HPE Ezmeral Data Fabric clusters.

Procedure

1. Access the admin CLI pod.
   For example:

   ```
   kubectl exec -it admincli-0 -n <namespace> -- /bin/bash
   ```

2. Execute the `edf shutdown cluster` command.
   For example:

   ```
   edf shutdown cluster
   ```

   Data Fabric cluster pods, such as MFS and CLDB, are shut down and rebooted, and then they are put into a wait state.

3. After you complete the upgrade or maintenance task, resume operations on the pods by entering the `edf startup resume` command.

Related reference

Command Reference: edf shutdown cluster on page 718
The `edf shutdown cluster` command shuts down core components in Kubernetes HPE Ezmeral Data Fabric clusters and prevents them from resuming operations.

Command Reference: edf startup {pause | resume} on page 719
The `edf startup pause` command flags core HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS, such that they will to enter into a nonfunctional state when they restart. The pods resume their startup sequence only after the `edf startup resume` command is executed.

Restarting the Data Fabric Cluster

This procedure resumes the startup process for Data Fabric clusters that implement HPE Ezmeral Data Fabric on Kubernetes. This procedure does not apply to bare-metal HPE Ezmeral Data Fabric clusters. This procedure does not restart the entire HPE Ezmeral Runtime Enterprise.

Prerequisites

- The Data Fabric cluster has been shut down with the `edf shutdown cluster` command.
- You must have access to the admin CLI pod (default name: `admincli-0`)

About this task

Use this procedure to resume startup operations on a Kubernetes Data Fabric cluster that has been shut down with the `edf shutdown cluster` command or has had its startup process paused by the `edf startup pause` command.

Procedure

1. On the Kubernetes master node, access the admin CLI pod.
   For example:

   ```
   kubectl exec -it admincli-0 -n <namespace> -- /bin/bash
   ```

2. Execute the `edf startup resume` command.
   For example:

   ```
   edf startup resume
   ```

   The startup process for Data Fabric cluster pods, such as CLDB and MFS, resumes.

3. Verify that the pods are ready.

   You must verify that all Data Fabric services are functional before restarting any Data Fabric operations, such as mirroring, or any tenant applications, such as Spark.

   You can check the status by executing the `edf report ready` command. Consider the following:

   - This command can take a couple of minutes to execute. You might also notice a delay between the display of the second and the third lines of the output.
   - When you execute this command for the first time, you might see the message that the *pods are not ready*. You must wait until the Data Fabric is online, to see the message that the *pods are ready*. It usually takes a few minutes for the Data Fabric to be online, however, it can take up to 30 minutes.

   The following example shows the output when the pods are not ready:

   ```
   edf report ready
   2021/06/14 23:22:34 [edf reports]: [INFO] Checking if pods are stabilized
   for upgrade. This may take a minute or two.
   2021/06/14 23:22:35 [edf reports]: [INFO] Valid MapR user ticket found,
   skipping ticket generation
   2021/06/14 23:24:31 [edf reports]: [ERROR] Pods are not ready for upgrade
   2021/06/14 23:24:31 [edf reports]: [ERROR] Check out /tmp/
   report-20210614232234 for details
   ```

   The following example shows the output when the pods are ready:

   ```
   edf report ready
   2021/06/14 23:28:01 [edf reports]: [INFO] Checking if pods are stabilized
   ```

```
for upgrade. This may take a minute or two.
2021/06/14 23:28:02 [edf reports]: [INFO] Valid MapR user ticket found,
skipping ticket generation
2021/06/14 23:29:52 [edf reports]: [INFO] Pods are ready
```

4. You can now restart tenant applications and perform other Data Fabric operations.

Related reference

Command Reference: edf shutdown cluster on page 718
The `edf shutdown cluster` command shuts down core components in Kubernetes HPE Ezmeral Data Fabric clusters and prevents them from resuming operations.

Command Reference: edf startup {pause | resume} on page 719
The `edf startup pause` command flags core HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS, such that they will to enter into a nonfunctional state when they restart. The pods resume their startup sequence only after the `edf startup resume` command is executed.

Command Reference: edf report ready on page 720
The `edf report ready` command reports the readiness of control plane Kubernetes pods in HPE Ezmeral Data Fabric clusters.

Upgrading and Patching Data Fabric Clusters on Kubernetes

NOTE: In this article, the term tenant refers to Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

Many aspects of an HPE Ezmeral Data Fabric cluster on Kubernetes can be reconfigured while the cluster is running. For example:

- You can change pod settings such as CPU, memory, and storage.
- You can change the number of pods by changing the `count` value in the CR section that describes that pod. For example, to increase the number of MFS pods, increase the value of `count` in the `mfs` section of the CR.
- You can upgrade a pod container image, typically to a new container image version, by changing the `image` value in the pod CR.

See Update Parameters for a list of the parameters that can be updated on a per-component basis.

Upgrading and Patching Procedure

Upgrades and patching changes are implemented as rolling updates, depending on the object workload type. For detailed information about the process for each workload type, see Online Update Behaviors on page 626.

CAUTION: Using the `edf update` feature to increase the number of ZooKeeper pods can cause the CLDB pods to become temporarily unavailable. Only upgrade the ZooKeeper pods when cluster downtime can be tolerated.

NOTE: If needed, see Scaling Up ZK, MFS, and CLDB on page 622.

To upgrade or patch any component:

1. If needed, perform a bootstrap upgrade to ensure that the cluster has the latest operator components. See Running a Bootstrap Upgrade.

2. Edit the CR for the component that you want to update. See Update Parameters

   NOTE: you can download the CR file using following commands:

   ```
   kubectl get dataplatform [name] -o yaml
   ```

   ```
   [name]-cr.yaml
   ```

   .

3. Apply the changes using the `kubectl apply` command.

4. Either:

   - **CLDB or ZooKeeper only**: Proceed to the next step.
   - **All others:** This completes the upgrade/patch process.

5. Log into the `admincli` pod, and then execute the following command:

   ```
   edf update cluster
   ```

   For example:

   ```
   kubectl exec -it admincli-0 -n mycluster -- /bin/bash
   edf update cluster
   ```

6. Verify that the pods are ready by executing the `edf report ready` command. The command can take a couple of minutes to execute. You might notice a delay between the display of the second and the third lines of the output.

   ```
   edf report ready
   2021/06/14 23:28:01 [edf reports]: [INFO] Checking if pods are stabilized
   for upgrade. This may take a minute or two.
   2021/06/14 23:28:02 [edf reports]: [INFO] Valid MapR user ticket found,
   skipping ticket generation
   2021/06/14 23:29:52 [edf reports]: [INFO] Pods are ready
   ```

Scaling Up ZK, MFS, and CLDB

NOTE: These objects cannot be scaled down.

To scale up ZK, MFS, and CLDB objects:

1. In the Data Fabric CR, change the ZK, MFS, or CLDB `failurecount` parameter, and then execute the `kubectl apply` command to apply the changes. See Zookeeper Core Object Settings, MFS Core Object Settings, and CLDB Core Object Settings.

2. Wait for the new pods to start up and be ready. You can verify readiness (`1/1 READY`) by executing the following command:

   ```
   kubectl get pods -n <cluster-name>
   ```

3. Create a new `admincli` pod by deleting the current pod:

   ```
   kubectl delete pod admincli-0 -n <cluster-name>
   ```

4. Wait for the new `admincli` pod to be `1/1 READY`, then exec into that pod and execute the `edf update cluster` command. This step refreshes the existing pods and makes them aware of the new pods. For example:

```
kubectl exec -it admincli-0 -n <cluster-name> -- /bin/bash
edf update cluster
```

Verifying the Upgrade Changes

Check the status of upgraded pods and parameters after applying an upgrade, patch, or configuration change:

- Execute the `edf report ready` command to ensure that the Data Fabric control plane pods are ready.

```
edf report ready
```

- Execute the `get pods` command to check the status of individual pods:

```
kubectl get pods -n mycluster -w
```

- Check parameter values by executing the `describe pod` command. For example, if you updated the image tag for pod `mcs-0`:

```
kubectl describe pod mcs-0 -n mycluster | grep -i image:
```

Updatable Parameters

This sections lists all of the parameters that can be updated on a per-component basis:

- All components
  - `baseimagetag`
  - `imageregistry`
- `admincli`
  - `count`
  - `image`
  - `limitcpu`
  - `limitdisk`
  - `limitmemory`
  - `logLevel`
  - `requestcpu`
  - `requestdisk`
  - `requestmemory`
- `cldb`

    - failurecount
    - image
    - limitcpu
    - limitdisk
    - limitmemory
    - logLevel
    - requestcpu
    - requestdisk
    - requestmemory
- mfs
    - image
    - groups: count
    - limitcpu
    - limitdisk
    - limitmemory
    - logLevel
    - requestcpu
    - requestdisk
    - requestmemory
- webserver
    - count
    - image
    - limitcpu
    - limitdisk
    - limitmemory
    - logLevel
    - requestcpu
    - requestdisk
    - requestmemory
- zookeeper

    - failurecount
    - image
    - limitcpu
    - limitdisk
    - limitmemory
    - loglevel
    - requestcpu
    - requestdisk
    - requestmemory
- hivemetastore
    - count
    - image
    - limitcpu
    - limitdisk
    - limitmemory
    - loglevel
    - requestcpu
    - requestdisk
    - requestmemory
- objectstore
    - count
    - image
    - hostports: limitcpu
    - hostports: limitdisk
    - hostports: limitmemory
    - hostports: loglevel
    - hostports: requestcpu
    - hostports: requestdisk
    - hostports: requestmemory
- collectd

    - `image`
    - `limitcpu`
    - `limitdisk`
    - `limitmemory`
    - `logLevel`
    - `requestcpu`
    - `requestdisk`
    - `requestmemory`
  - `grafana`
    - `count`
    - `image`
    - `limitcpu`
    - `limitdisk`
    - `limitmemory`
    - `logLevel`
    - `requestcpu`
    - `requestdisk`
    - `requestmemory`
  - `opentsdb`
    - `count`
    - `image`
    - `limitcpu`
    - `limitdisk`
    - `limitmemory`
    - `logLevel`
    - `requestcpu`
    - `requestdisk`
    - `requestmemory`

Online Update Behaviors

Upgrades and patching changes are implemented as rolling updates, depending on the object workload type.

- **Deployment - *Grafana*, *Kibana*, *Collectd*:** The DataPlatform operator sees the change and launches a new pod on the cluster (it can be on the same node or another). This new pod contains the changes specified in the updated CR. After the new pod is ready, a "pre-stop" script gracefully shuts down the processes, and then the existing pod is terminated. This process repeats until all pods of a deployment are updated.
- **StatefulSet** -The DataPlatform operator brings down a pod of a specific StatefulSet. A “pre-stop” script ensures processes are gracefully shut down before the pod is terminated. A new pod that has the updated configuration changes is then brought up on the same physical node. The operator waits until this new pod is ready, and then repeats until all pods are updated.

  Because the core data pods, such as CLDB and ZK perform critical functions, the operator does not update the pods after after CR changes are applied. You must execute a command (see Upgrading and Patching Procedure on page 621) to complete the process.

  Examples of StatefulSet pods include the following:

  - CLDB
  - ZooKeeper
  - MCS
  - MFS
  - admincli
  - Object Store
  - NFS Server
  - Elasticsearch
  - OpenTSDB
  - Hive Metastore
  - Gateway pods, such as MapR Gateway, HTTPS Gateway, Data Access Gateway, and Kafka REST Gateway
- **DaemonSet - *Fluentd*:** All running pods are brought down simultaneously, and new pods are then started up in parallel. Parallel updates are not common to all DaemonSets, but are appropriate for Fluentd. A “pre-stop” script ensures processes are gracefully shut down before the pod is terminated.

Related reference

Command Reference: edf update cluster on page 718
The `edf update cluster` command updates components in HPE Ezmeral Data Fabric on Kubernetes clusters.

Command Reference: edf report ready on page 720
The `edf report ready` command reports the readiness of control plane Kubernetes pods in HPE Ezmeral Data Fabric clusters.

Managing HPE Ezmeral Data Fabric on Kubernetes

NOTE: In this article, the term tenant refers to HPE Ezmeral Data Fabric tenants (formerly '"MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

This article describes managing and accessing the Data Fabric cluster and tenants.

Managing Using the CLIs

You can interact with the Data Fabric cluster via the Command Line Interface (CLI) pods (such as `admincli-0`) created in the cluster namespace. You can directly access individual pods (such as CLDB), but best practice is to only do this when needed to debug an issue. In the Kubernetes environment:

- You can access pods via either the `kubectl exec` command or via SSH, as described in SSH.
- Pods are ephemeral. Any state created in a pod might disappear.
- There are two main types of administration pods:
  - The admin CLI pod in the Data Fabric cluster namespace.
  - Tenant CLI pods in the individual tenant namespaces.

Admin CLI Pod

This pod is suitable for runningmaprcli commands commands and data-loading scripts (link opens in a new browser tab/window).HPE Ezmeral Data Fabric Cluster Administrators should access the admin CLI (`admincli-0`) pod in the dataplatform namespace.

For example, you can access the admin CLI pod by using `kubectl` in the Kubernetes Web Terminal:

1. Get the value of the namespace:

   ```
   kubectl get pods -A | grep -e admincli-0 -e NAMESPACE
   ```

   The value of the namespace is returned.

2. The default name of the admin CLI pod is `adminicli-0`. Access the admin CLI pod using a `kubectl exec` command:

   ```
   kubectl exec -it admincli-0 -n <namespace> -- /bin/bash
   ```

Tenant CLI Pod

Kubernetes Tenant Member users can generate tickets or start Spark jobs via the tenant CLI Terminal pod provided in most tenant namespaces. Kubernetes Tenant Administrator users can use the tenant CR to disable this pod.

Accessing the Data Fabric Cluster

There are several ways to access the Data Fabric cluster, filesystem, and other installed components:

HPE Ezmeral Data Fabric Control System

You can access the HPE Ezmeral Data Fabric Control System (MCS) in your internal environment by clicking the Data Fabric Managed Control System link for the Data Fabric cluster in the **Kubernetes Clusters** screen.

NOTE: HPE Ezmeral Data Fabric Control System provides less information in a Kubernetes environment than in a bare-metal HPE Ezmeral Data Fabric environment. The HPE Ezmeral Data Fabric Control System also allows you to manage all aspects of a cluster and provides node-specific data-management features in the bare-metal environment.

SSH

You can use SSH to log in to a container and gather information. By default, all containers come up with SSH running.

- **Internal SSH Access:** SSH is available to Port 22 in every Data Fabric cluster container. Within a cluster, you can SSH from one container to another without specifying a port.
- **External SSH Access:** You must provide the `sshport` and `hostname` to access a container from outside the cluster. If the `sshport` is already defined, you can find that port in the CR for the container. You can understand with following example.

  Following example is for grafana service:

  1. To find the port number, use the following command:

     ```
     kubectl get services -n mycluster | grep -i grafana
     ```

     Example of the result:

     ```
     grafana-svc NodePort 10.111.102.36 <none> 3000:31755/TCP
     ```

     The number after the colon is the port number. In this case, 31755 is the port number for `grafana` service.

  2. To access the service:
     - Access the login page, using the following URL format:

       ```
       https://<ip address of cluster node>:31755
       ```

     - On the login page, enter the username and password.

       For information about how to get the username and password for the mapr user, see Data Fabric Cluster Administrator Username and Password on page 600.

  NOTE: The preceding example is for grafana service. The same procedure can be applied to other services, such as Kibana. To find the port number for any service, enter that service in <sevice name> in the following command:

  ```
  kubectl get services -n mycluster | grep -i <service name>
  ```

  To find the list of container services available, execute the following command:

  ```
  kubectl get services -n mycluster
  ```

To determine the hostname for a container, execute the `kubectl get pod` command. For example:

```
kubectl get pod -n mycluster cldb-0 -o wide
NAME     READY     STATUS     RESTARTS     AGE       IP              NODE
cldb-0   1/1       Running    0            3h32m     10.192.2.10     dev.01.lab
```

To log in using SSH, specify the external port, your user name, and the host name. For example:

```
ssh -p 5000 userj@dev.01.lab
```

Comparing EXEC vs. SSH Access

The `kubectl exec` command is the easiest way to access a container, however this access occurs as the user the container runs as (typically `mapr`). This access is useful for Administrators but may include permissions unsuited to non-admin users. You may want to restrict container access to SSH, which only grants users the privileges granted to their current user accounts.

API

The following APIs grant access to the installed components (link opens in a new browser tab/window):

- HDFS
- S3 Gateway

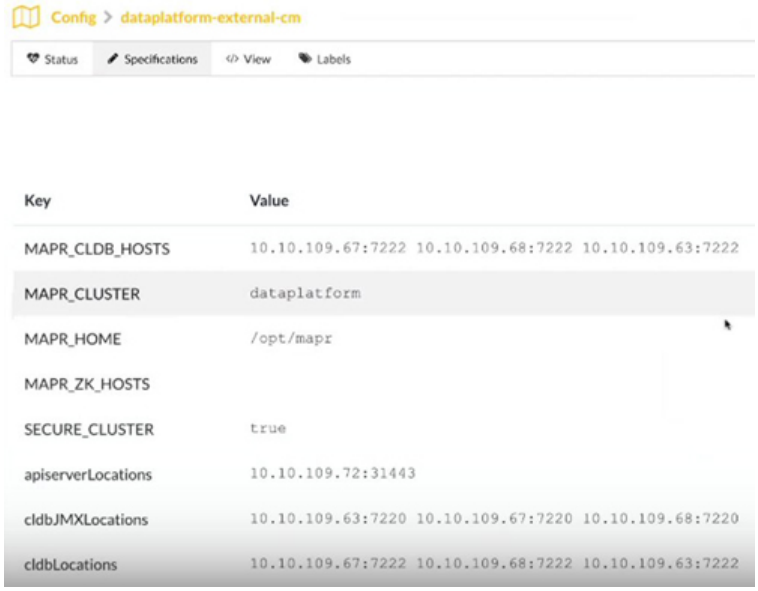
POSIX Client

You can access the Data Fabric cluster using the POSIX Client via the CSI driver. The CSI driver reports the Kubernetes worker node where the POSIX client is scheduled to run as the POSIX client host.
The StorageClass should specify either the IP address of the MCS pod or the `webserver` service. For example:

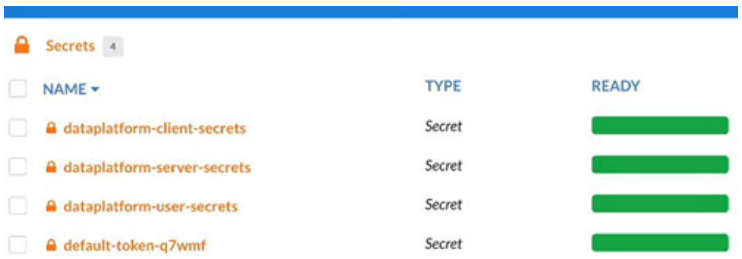
```
mcs-svc.<clustername>.svc.cluster.local
```

External Access to Services

The CLDB object and most services are accessible outside of the cluster, and some include open host ports. You can connect to the corresponding pods without having to run as a pod inside the cluster.
In the namespace `hpe-externalclusterinfo`, the `<data-fabric-cluster-name>-external-cm` configmap provides information about how to access these services from outside the cluster:



The `Data Fabric-hivesite-cm` Hivesite configmap, shows the Hivesite information that is available external to the cluster. `hpe-externalclusterinfo` also provides the secrets needed to connect to the cluster from an external compute tenant that doesn't exist inside the cluster:



Logging and Coredump Directory Structure

The following logs are available:

- The physical Kubernetes node hosting the pod includes component logs.
- Data Fabric logs (link opens in a new browser tab or window).

All Data Fabric pods share a parent logging directory path on the node. This path can be configured in the Data Fabric CR. For example:

```
apiVersion: hcp.hpe.com/v1
kind: DataPlatform
metadata:
  name: dataplatform
spec:
  baseimagetag: "202007092140C"
  imageregistry: gcr.io/mapr-252711
  imagepullsecret: hpe-imagepull-secrets
  environmenttype: hcp
  disableha: false
  loglocation: /var/log/mapr
  corelocation: /var/mapr/cores
  podinfolocation: /var/mapr/podinfo
  security:
    disablesecurity: false
    systemusersecret: system-user-secrets
```

The platform creates logs in this directory that correspond to each pod under this directory and follow a predefined directory structure. The pods themselves are ephemeral, but all logging directories persist on the physical nodes and can be retrieved later.

CAUTION:

The `LOGLOCATION` , `CORELOCATION` , and `PODLOCATION` directories cannot be nested, because this could cause a mount issue. For example, the corelocation directory cannot be nested under either of the other two directories.

Log Format

Logs follow this general format:

```
/UserSpecifiedParentDir/ClusterName/ClusterCreationTime/PodTypeName/PodName
```

For example, a CLDB pod log might look like this:

```
/var/log/mapr/mycluster/20200802174844/cldb/cldb-0
```

The log components are:

- `/UserSpecifiedParentDir` - Data Fabric CR property. Default is `/var/log/mapr/`. Hewlett Packard Enterprise recommends keeping the partition that contains `<UserSpecifiedParentDir>` separate from the partition that contains `/var`, to prevent filling the `/var` partition and risking OS stability/crashes.
- `/ClusterName` - Cluster or namespace name.
- `/ClusterCreationTime` - Time a specific cluster instance was created. This identifier is used because a cluster name can have multiple lifecycles and different cluster instances can share the cluster name.
- `/PodTypeName` - Pod type, such as `cldb` or `mfs`.
- `/PodName` - Pod name.

The `/opt/mapr/cluster_logs` directory is volume-mounted to the `UserSpecifiedParentDir` on the node. This directory is the starting point for all logs on the corresponding physical node. When created, each pod creates its own logging directory following the above rule based on the `UserSpecifiedParentDir`. This per-pod approach ensures that the same directory won't be recreated if it already exists. Stateful pods that do not change nodes between failures (such as CLDB and `mfs`) will keep using the same directory after a pod restart.

Most logs for each pod name contain the same content as `/opt/mapr/logs` because they are replaced with a symlink that points to the logging directory created by each pod. Additional logs (ZooKeeper transactions, `collectd`, `grafana`, etc.) are also included here. A symlink is created whenever a pod starts or restarts. A sticky bit ensures that this symlink behaves like a directory from an application perspective.

Coredump Files

The coredump file uses the same logic as logging. A separate directory called `opt/cluster_cores` is created and mounted to the user-specified core-dump directory in the Data Fabric CR. All core dumps corresponding to each pod follow the same hierarchy as logging. Here again, symlinks replace the original core directory, and a catalog file is added with an imageID where the specific image generates cores.

Spyglass Monitoring with Grafana

You can access Grafana by clicking the Grafana Endpoint link for the Data Fabric cluster in the **Kubernetes Clusters** screen. See The Kubernetes Clusters Screen.

NOTE: Grafana Endpoint is not available for Footprint-Optimized configuration.

The Grafana dashboard allows you to monitor the following components:

- CPU
- Memory
- Network I/O
- Swap
- System Disk IOPS
- System Disk Throughput

NOTE: These metrics do not include Data Fabric-specific metrics, which are node-specific and not pod-specific. Metrics are filtered on the `CollectD` pod's FQDN.

To visualize these metrics in the Grafana dashboard:

1. To find the node on which the grafana pod is running, execute the following command:

```
kubectl get pods -o wide -n <Cluster Name> | grep grafana
grafana-7c8fcbb86f-58mj4 1/1 Running 0 40h 10.192.4.29
mip-bd-vm567.mip.storage.hpecorp.net <none> <none>
```

2. To get the port that Grafana is listening on, execute the following command:

```
kubectl get services -n <Cluster Name> | grep grafana
grafana-svc NodePort 10.109.211.237 <none> 3000:30486/TCP
```

   NOTE: This will be typically in the 30000+ range.

3. Combine the node IP and port number from Step 1. and Step 2, and build the Grafana dashboard URL:

```
https://<node-ip>:<port>
```

4. Launch a browser and navigate to the Grafana dashboard URL:

```
https://<node-ip>:<port>
```

5. Log in to the Grafana interface using the system username (default is mapr) and password.

   NOTE: You can get the password using the following command:

```
kubectl get secret system -n <cluster-name> -o yaml | grep
MAPR_PASSWORD | head -n 1 | awk '{ print $2 }
'
```

   where `df` is the name of the cluster.

6. Select **Home** > **Node Dashboard** to view the metrics.

The page displays the node resources used by components across pods in the Kubernetes environment.

Kibana Monitoring

You can access Kibana by clicking the Kibana Endpoint link for the Data Fabric cluster in the **Kubernetes Clusters** screen. See The Kubernetes Clusters Screen.

NOTE: Kibana Endpoint is not available for Footprint-Optimized configuration.

The default Kibana username is: `admin`

The default password can be obtained from system secret in the Data Fabric namespace. For example, if the name of the Data Fabric cluster is `df`, the command to get the password is the following:

```
kubectl -n df get secret system -o jsonpath="{$.data.MAPR_PASSWORD}" |
base64 -d
```

Managing Storage Pools and File System Instances

The HPE Ezmeral Data Fabric on Kubernetes supports storage pools and multiple instances of the file system. These features are implemented through the `storagepoolsize` and `storagepoolsperinstance` parameters for the `diskinfo` object in the Data Fabric CR.

```
  loglevel: INFO
mfs:
  image: mfs-6.2.0:202007082125C
  groups:
    - name: group1
      count: 2
      diskinfo:
        diskcount: 10
        disktype: SSD
        storagepoolsize: 0
        storagepoolsperinstance: 1
      sshport: 5001
      hostports:
        - hostport: 5660
        - hostport: 5692
        - hostport: 5724
        - hostport: 5756
        - hostport: 8660
      requestcpu: "4000m"
```

- `storagepoolsize` - You can use storage pools to group disks and can control the number of disks in a storage pool by adjusting the `storagepoolsize` value. Each `mfs` group can have a different storage pool size. A storage pool can have up to 32 drives.
- `storagepoolsperinstance` - integer - Number of storage pools that an instance of the file system will manage. The platform launches multiple instances of the file system based on the specified number of storage pools. The default value is `0`, which sets the number of storage pools based on internal algorithms. A value greater than 32 generates an error.

Most installations benefit from having both of these parameters set to 0; however, some advanced situations may call for different settings. See `diskinfo` in MFS Core Object.

Related concepts

Data Fabric Cluster Administrator Username and Password on page 600
This topic defines the HPE Ezmeral Data Fabric cluster administrator and provides information about the default username (`mapr`) and password for the Data Fabric cluster administrator in HPE Ezmeral Runtime Enterprise deployments.

Using the CSI

Deploying HPE Ezmeral Data Fabric for Kubernetes automatically configures the HPE Container Storage Interface (CSI) plug-in as the default Storage Class. Your application pod can then set up PV/PVC to connect to the CSI plug-in. See the CSI Overview in the HPE Ezmeral Data Fabric documentation (link opens in new browser window or tab).

For information about the CSI driver versions supported on HPE Ezmeral Data Fabric on Kubernetes for this release of HPE Ezmeral Runtime Enterprise, see Support Matrixes on page 54.

If desired, you may alternatively use a non-HPE local persistent-volume CSI provided by a third party. If you do this, HPE Ezmeral Runtime Enterprise complies with how OSS Kubernetes works with that CSI.

Types of HPE CSI Plug-In Drivers

HPE Ezmeral Runtime Enterprise allows you to choose from two types of HPE CSI plugins:

- FUSE binary driver. This is the default plug-in.
- Loopback NFS driver.

Both plug-ins are functionally the same and offer similar performance. The Loopback NFS driver is newer.

Both plug-ins can be used on bare metal implementations of HPE Ezmeral Data Fabric and on implementations of HPE Ezmeral Data Fabric on HPE Ezmeral Runtime Enterprise.

Both plug-ins support Raw Block Volumes (link opens in new browser window or tab).

HPE Ezmeral Runtime Enterprise automatically creates the Kubernetes storage class for each CSI when the Kubernetes cluster is created. The default storage class is the FUSE version, but you can choose to use either CSI version when creating the PV/PVC.

More Information About CSI

For information about CSI, see the HPE Ezmeral Data Fabric documentation (links open in a new browser window or tab):

- For information about the differences between the plugins, see Comparing the FUSE POSIX and Loopback NFS Plugins.
- For compatibility information, see CSI Version Compatibility.
- For information about the content and fixes for a specific CSI version, see the CSI Release Notes.

Upgrading the CSI Plug-In

Use this procedure to upgrade the CSI plug-in on Kubernetes clusters that are not **HPE Ezmeral Data Fabric on Kubernetes** clusters in deployments of HPE Ezmeral Runtime Enterprise 5.3.5 or later. Upgrading the CSI plug-in requires restarting or recreating the affected pods.

About this task

For deployments of HPE Ezmeral Runtime Enterprise 5.3.5 or later, this procedure describes how to upgrade the CSI plug-in on Kubernetes clusters that are **not HPE Ezmeral Data Fabric on Kubernetes** clusters. The upgrade requires restarting or recreating pods in the cluster that use a persistent volume claim (PVC).

If this deployment of HPE Ezmeral Runtime Enterprise includes a **HPE Ezmeral Data Fabric on Kubernetes** cluster, the CSI-plugin for that cluster is managed as part of the HPE Ezmeral Data Fabric on Kubernetes deployment, and this procedure does not apply.

If pods on this cluster use a persistent volume claim (PVC) provisioned through HPE CSI driver 1.0.x or 1.1.x, before you upgrade from Kubernetes 1.18.x, upgrade the HPE CSI driver to version `1.2.7-1.0.7`.

For HPE Ezmeral Runtime Enterprise5.5.0 and 5.5.1, see workaround EZCP-3738 in Issues and Workarounds on page 15.

Procedure

1. Instruct users not to use pods in this cluster until after the upgrade.
2. Using SSH, log in to the Kubernetes Master node.
3. Execute the following command to create a directory:

```
mkdir /opt/bluedata/common-install/scripts/tools/hpe-csi-upgrade
```

4. Copy the new directory created on Master node to `/opt/hpe/kubernetes/tools/hpe-csi-upgrade` on the Controller node, using the following command:

```
scp -r <install_user>@<controller-IP>:/opt/hpe/
kubernetes/tools/hpe-csi-upgrade /opt/bluedata/common-install/scripts/
tools/hpe-csi-upgrade
```

5. Update the permissions of the directory using following commands:

```
chmod -R 755 /opt/bluedata/common-install/scripts/tools/hpe-csi-upgrade
```

```
chown -R <install_user>:<install_group> /opt/bluedata/common-install/
scripts/tools/hpe-csi-upgrade
```

6. Stop or Delete the pods that use a persistent volume claim (PVC) provisioned the HPE CSI driver, using the following steps:

   a. Execute the following command to list the pods that needs to be stopped, or deleted :

   ```
   cd /opt/bluedata/common-install/scripts/tools/hpe-csi-upgrade
   ```

   ```
   ./find_pods.sh
   ```

   Samples of Output:

   Output 1:

   ```
   [user1@host1 ~]$ ./find_pods.sh
   Namespace: ns1, Pod(s): pod-1-1 pod-1-2
   Namespace: ns2, Pod(s): pod-2-1 pod-2-2
   ```

   Or

   Output 2:

   ```
   [user1@host1 ~]$ ./find_pods.sh
   None of pods mount the PVC/PV provisioned by DF CSI driver
   ```

   b. Do one of the following:

   - If the pod you identified in the previous step was launched by a pod object, backup the pod, and then delete the pod.

     Backup the pod, using the following command:

     ```
     kubectl -n <namespace> get pods <podname> -o yaml >
     backup_<unique_name>.yaml
     ```

     Delete the pod, using the following command:

     ```
     kubectl -n <namespace> delete pods <podname>
     ```

- If the pod was launched by DaemonSet, set `non-existing=true` in nodeSelector, using the following command:

```
kubectl -n <namespace> patch daemonset <name-of-daemon-set> -p
'{"spec": {"template": {"spec": {"nodeSelector": {"non-existing":
"true"}}}}}'
```

- If the pod was launched by StatefulSet, Deployment, or ReplicaSet object, set `'replicas=0'`, using the following command:

```
kubectl -n <namespace> scale <object-type>/
<object-name> --replicas=0
```

7. Change directories to the `hpe-csi-upgrade` directory:

```
cd /opt/bluedata/common-install/scripts/tools/hpe-csi-upgrade
```

8. Execute the `hpe-csi-upgrade` script, specifying the new CSI versions as follows:

```
./hpe-csi-upgrade.sh -u <CSI-FUSE-version>-<loopback-NFS-version>
```

For example:

```
./hpe-csi-upgrade.sh -u 1.2.7-1.0.7
```

9. Restart or recreate all pods that you stopped or deleted at Step 6:, using the steps that follow:
   - If it is Pod object, recreate the pod, using the following command:

```
kubectl apply -f backup_<unique_name>.yaml
```

   - If it is DaemonSet object, delete the nodeSelector, using the following command:

```
kubectl -n <namespace> patch daemonset <name-of-daemon-set> --type
json -p='[{"op": "remove", "path": "/spec/template/spec/nodeSelector/
non-existing"}]'
```

   - If it is StatefulSet, Deployment, or ReplicaSet object, set 'replicas=X'. Where X is non-zero, using the following command:

```
kubectl -n <namespace> scale <object-type>/<object-name> --replicas=X
```

HPE Ezmeral Data Fabric Control System (MCS)

The HPE Ezmeral Data Fabric includes the HPE Ezmeral Data Fabric Control System (MCS) cluster-management tool that you can use to administer HPE Ezmeral Data Fabric clusters. The Control System provides command line and REST APIs, job monitoring metrics, and help troubleshooting cluster issues.

HPE Ezmeral Runtime Enterprise automatically installs and configures AD/LDAP authentication for HPE Ezmeral Data Fabric as part of cluster creation, and also installs and enables the MAST Gateway Service. If desired, you can set up HttpFS as described below. Please also see the following sections of this article:

- Accessing the HPE Ezmeral Data Fabric Control System (MCS) on page 639 for access details, including the username and password.
- HPE Ezmeral Data Fabric Commands on page 640 for the list of commands that you can run on the Controller host, including the commands needed to identify the containers/virtual nodes running the CLDB and ZooKeeper services.

Implementation Differences

HPE Ezmeral Data Fabric Control System provides less information in a Kubernetes environment than in a bare-metal HPE Ezmeral Data Fabric environment. In a bare-metal HPE Ezmeral Data Fabric implementation, the Control System enables you to manage all aspects of a Data Fabric cluster and provides node-specific data-management features in the bare-metal environment.

HPE Ezmeral Data Fabric Control System in a Kubernetes environment:

- Primarily provides Volumes and Services information.
- Does not display the **Overview**, **Nodes**, **Data**, **Data>Streams**, or **Data>Tables** menu options.
- Volumes information is equivalent to the **Data>Volumes** information provided in a bare-metal environment.
- The Control System only displays Services under the headings **Core**, **Others**, and **Monitoring**. You cannot start, stop, or restart services.
- The **User Permissions** screen only allows you to remove users.

See 6.2 Administration for additional information about using the control system (link opens in a new browser tab or window).

Setting up HttpFS for HPE Ezmeral Data Fabric

HPE Ezmeral Data Fabric supports the optional HttpFS package that allows data access via cURL or any other HTTP client. For additional information, please see the HPE Ezmeral Data Fabric article Installation Instructions. Please also see Additional information (link opens an external website in a new browser tab/window).

HttpFS includes the following key features:

- By default, HttpFS runs in Secure mode and requires basic authentication. You may also configured it to use Kerberos for authentication, as described below.
- HttpFS impersonates users. For example, if User_A authenticates, then any files will be written/read as User_A. All volume and file ACEs are honored.
- HttpFS provides full access to files in MaprFS paths on HPE Ezmeral Data Fabric. It is not integrated with DataTaps.
- When browsing volumes, HttpPFS is similar to MapR Hadoop in that it provides access to data within any mounted volume without exposing volume objects.
- Volume objects are configuration-level structures that are viewed/modified through either the Control System or Data Fabric CLI commands. See Accessing the HPE Ezmeral Data Fabric Control System (MCS) on page 639 and HPE Ezmeral Data Fabric Commands on page 640.

Setting up HttpFS

To set up HttpFS on HPE Ezmeral Data Fabric:

1. Log in to the CLDB node by executing the following command:

```
bdmapr --root /bin/bash
```

2. Verify that `yum` works.
3. Update the proxy setting by executing the following command:

```
echo "proxy=http://web-proxy.corp.enterprise.com:8080" >> /etc/yum.conf
```

4. Update the MapR repository configuration by executing the following command:

```
sed -i /etc/yum.repos.d/mapr.repo 's/gpgcheck=1/gpgcheck=0/g'
sed -i /etc/yum.repos.d/mapr.repo 's/repo_gpgcheck=1/repo_gpgcheck=0/g'
```

5. Install HttpFS by executing the following commands:

```
yum install -y mapr-httpfs
/opt/mapr/server/configure.sh -R
```

6. When HttpFS starts, test it in a web browser by executing the following command:

```
https://<controller_ip>:14000/
```

7. Read a file using the web browser by executing the following command:

```
https://<controller_ip>:14000/webhdfs/v1/<maprfs_path_to_file>?op=OPEN
```

You can now use Postman to create directories or write files. For example, to create a new request:

- **Type:** PUT
- **URL:** `https://<controller_ip>:14000/webhdfs/v1/tmp/testdirectory?op=MKDIRS`
- **Authorization configuration:**
  - **Type:** Basic Auth
  - **Username:** Any valid user (could be `admin`)
  - **Password:** Password for that user

Enable Insecure mode. You will be prompted to enable Insecure mode the first time you send any request and get back the self-signed certificates.

Accessing the HPE Ezmeral Data Fabric Control System (MCS)

HPE Ezmeral Runtime Enterprise automatically routes any request made to port 8443 to the MCS. Thus, once you have configured AD/LDAP authentication, enabled the MAST Gateway service, and set up HttpFS, you can access the MCS at:

```
https://<gateway_host_ip_address>:8443
```

If platform HA is enabled, then the `<controller_ip_address>` must be either the Primary Controller, Shadow Controller, or Cluster IP address. Do not use the Gateway host IP address.

The default username is `admin`. The administrator password is stored on the Primary Controller host at `/opt/bluedata/mapr/conf/mapr-admin-pass`.

The HPE Ezmeral Data Fabric Container Location Database (CLDB) runs on the Primary, Shadow, and Arbiter hosts.

The HPE Ezmeral Data Fabric service runs in a container on the Controller host.

HPE Ezmeral Data Fabric Commands

Run the following commands on the Controller host to get the information needed:

Task	CLI Command
List all nodes with services running on them, along with the topology.	`bdmapr maprcli node list -columns h,svc,racktopo,id`
List all volumes that are present on a node.	`bdmapr maprcli volume list -filter -nodes FQDN_OF_THE_HOST -columns volumename,minreplicas,numreplicas,mountdir,quota,advisoryquota`
List license info.	`bdmapr maprcli license list`
Zookeeper status.	`bdmapr --root /opt/mapr/zookeeper/zookeeper-3.4.11/bin/zookeeper qstatus`
CLDB Master node.	`bdmapr maprcli node cldbmaster`
List CLDB/Zookeeper nodes	`bdmapr maprcli node listcldbzks`
Log on to MapR Container Shell (MCS).	`bdmapr --root bash`

Disk Management in HPE Ezmeral Data Fabric on Kubernetes

The topics in this section describe disk management tasks for disks that are part of storage pools on HPE Ezmeral Data Fabric on Kubernetes on HPE Ezmeral Runtime Enterprise.

The HPE Ezmeral Data Fabric filesystem groups disks into storage pools. A storage pool usually consists of up of two or three disks, but can include more disks. Write operations within a storage pool are striped across disks to improve write performance. Typically, when you manage disks in HPE Ezmeral Data Fabric on Kubernetes, you are interacting with storage pools.

Adding a Disk

This procedure describes adding a disk to a Data Fabric that implements HPE Ezmeral Data Fabric on Kubernetes on HPE Ezmeral Runtime Enterprise.

Prerequisites

- You know the pod to which you are adding the disk.
- You know the location (physical or virtual) to which you are adding the disk.
- **Required access rights:**
  - Platform Administrator or Kubernetes Cluster Administrator access rights are required to download the admin kubeconfig file, which is needed to access Kubernetes cluster pods (see Downloading Admin Kubeconfig on page 486).
  - You must be logged on as the root user on the node that contains the disk and on which the Kubernetes cluster is running.

Procedure

1. Add the disk to the storage system.

   For more information about this step, refer to the documentation for your server and storage system.

   For example, if you are adding storage to a physical server, add the physical disk drives to the disk enclosure.

2. Determine which node the pod is running on.

   In the following example, the disk is to be added to pod `mfs-1`.

   ```
   kubectl describe pod mfs-1 -n mydfcluster | grep Node:
   Node:        mydfnode1-default-pool/192.0.2.75
   ```

3. Delete the pod to which you want to add the disk (the pod restarts automatically).

   For example:

   ```
   kubectl delete pod mfs-1 -n mydfcluster
   ```

4. Access the pod to which you want to add the disk.

   For example:

   ```
   kubectl exec -it mfs-1 -n mydfcluster -- /bin/bash
   ```

5. Get the current list of disks from the node annotations.

   For example:

   ```
   kubectl describe node mydfnode1-default-pool | grep ssdlist
           hpe.com/ssdlist: /dev/sdb,/dev/sdc
   ```

6. Add the new disk to the `ssdlist` annotation.

   In the following example, the new disk is `/dev/sdd`:

   ```
   kubectl annotate --overwrite nodes mydfnode-default-pool hpe.com/
   ssdlist='/dev/sdb,/dev/sdc,/dev/sdd'
   ```

7. Verify that the annotations include the disk you added in the previous step.

   For example:

   ```
   kubectl describe node mydfnode-default-pool | grep ssdlist
           hpe.com/ssdlist: /dev/sdb,/dev/sdc,/dev/sdd
   ```

8. Log in to mfs and verify that the added disk is included in the directory that contains the logical links to Data Fabric disks (`/var/mapr/edf-disks/`).

   For example:

   ```
   ls -l /var/mapr/edf-disks/
   ...
   lrwxrwxrwx 1 root root 8 Nov  4 15:28 drive_ssd_0 -> /dev/sdb
   lrwxrwxrwx 1 root root 8 Nov  4 15:28 drive_ssd_1 -> /dev/sdc
   lrwxrwxrwx 1 root root 8 Nov  4 16:59 drive_ssd_3 -> /dev/sdd
   ...
   ```

   In `maprcli` commands, you specify the disk using the internal name that the Data Fabric file system uses to refer to the disk.

   In the preceding example, the internal name for the `dev/ssd` disk is `drive_ssd_3`.

9. Add the new disk to the Data Fabric file system.

   This step reformats the disk. Any data on the disk will be lost.

   In HPE Ezmeral Data Fabric on Kubernetes deployments, the `host` parameter of `maprcli` commands refers to the pod.

   In the following example, the disk `drive_ssd_3` is being added:

   ```
   maprcli disk add -disks /var/mapr/edf-disks/drive_ssd_3 -host
   mfs-1.mfs-svc.mydfcluster.svc.cluster.local
   ```

10. Verify that the new disk is included in the Data Fabric configuration file:

    To display the configuration file, enter the following command:

    ```
    cat /opt/mapr/conf/disktab
    ```

11. Verify that the new disk exists in the Data Fabric file system.

    For example, verify that the system displays a result for the following command:

    ```
    maprcli disk listall | grep mfs-1 | grep sdd
    ```

12. Verify that there is a new storage pool that includes the new disk.

    To display the list of storage pools, enter the following command:

    ```
    /opt/mapr/server/mrconfig sp list -v
    ```

Removing a Disk

This procedure describes removing a disk from a Data Fabric that implements HPE Ezmeral Data Fabric on Kubernetes on HPE Ezmeral Runtime Enterprise.

Prerequisites

- You know which disk to remove.
- You know which pod is associated with the disk you want to remove.
- **Required access rights:**

- Platform Administrator or Kubernetes Cluster Administrator access rights are required to download the admin kubeconfig file, which is needed to access Kubernetes cluster pods (see Downloading Admin Kubeconfig on page 486).
- You must be logged on as the root user on the node that contains the disk and on which the Kubernetes cluster is running.

IMPORTANT: Removing disks from a storage pool removes all the remaining disks from that storage pool. See Removing Disks from the File System (link opens in a new browser tab/window) for more details.

Procedure

1. Access the pod from which you want to remove the disk.

   In the following example, the pod is `mfs-1`.

   ```
   kubectl exec -it mfs-1 -n mydfcluster -- /bin/bash
   ```

2. Determine to which storage pool the disk belongs.

   When you remove a disk, the other disks in the storage pool are also removed.

   To display the list of storage pools, enter the following command:

   ```
   /opt/mapr/server/mrconfig sp list -v
   ```

3. Determine the internal name used by the Data Fabric file system to refer to the disk.

   In `maprcli` commands, you specify the disk using the internal name that the Data Fabric file system uses to refer to the disk.

   To determine the internal name of a disk, list the directory that contains the logical links to Data Fabric disks (`/var/mapr/edf-disks/`).

   In the following example, the internal name of the `dev/ssd` disk is `drive_ssd_3`:

   ```
   ls -l /var/mapr/edf-disks/
   ...
   lrwxrwxrwx 1 root root 8 Nov  4 15:28 drive_ssd_0 -> /dev/sdb
   lrwxrwxrwx 1 root root 8 Nov  4 15:28 drive_ssd_1 -> /dev/sdc
   lrwxrwxrwx 1 root root 8 Nov  4 16:59 drive_ssd_3 -> /dev/sdd
   ...
   ```

4. Remove the disk from the Data Fabric file system.

   In HPE Ezmeral Data Fabric on Kubernetes deployments, the `host` parameter of `maprcli` commands refers to the pod.

   In the following example, `drive_ssd_3` is being removed:

   ```
   maprcli disk
   remove -host mfs-1.mfs-svc.mydfcluster.svc.cluster.local -disks /var/
   mapr/edf-disks/drive_ssd_3 -force false
   ```

5. Determine which node the pod is running on.

   For example:

   ```
   kubectl describe pod mfs-1 -n mydfcluster | grep Node:
   Node:        mydfnode1-default-pool/192.0.2.75
   ```

6. Get the current list of disks from the node annotations.

   For example:

```
kubectl describe node mydfnode-default-pool | grep ssdlist
        hpe.com/ssdlist: /dev/sdb,/dev/sdc,/dev/sdd
```

7. Remove the disk from the `ssdlist` annotation:

   You remove a disk from the `ssdlist` by overwriting the existing list with new list that includes all the disks except for the disk you want to remove.

   In the following example, the disk `/dev/sdd` has been removed.

```
kubectl annotate --overwrite nodes mydfnode-default-pool hpe.com/
ssdlist='/dev/sdb,/dev/sdc'
```

8. Verify that the disk has been removed from the node annotations.

   For example:

```
kubectl describe node mydfnode-default-pool | grep ssdlist
        hpe.com/ssdlist: /dev/sdb,/dev/sdc
```

9. Verify that the logical link for the removed disk is no longer in the directory that contains the logical links to Data Fabric disks (`/var/mapr/edf-disks/`).

   For example:

```
ls -l /var/mapr/edf-disks/
...
lrwxrwxrwx 1 root root 8 Nov  4 15:28 drive_ssd_0 -> /dev/sdb
lrwxrwxrwx 1 root root 8 Nov  4 15:28 drive_ssd_1 -> /dev/sdc
...
```

10. Edit the Data Fabric configuration file (`/opt/mapr/conf/disktab`) to delete the reference to the disk you are removing:

    For example:

```
vi /opt/mapr/conf/disktab
```

11. Identify and delete logs that are associated with the removed disk.

    To list the logs, use the following command:

```
ls /opt/mapr/logs
```

12. Verify that the removed disk is not in the Data Fabric file system:

    For example, if there is no `ssd` disk in the `mfs-1` pod, the following command does not return a result:

```
maprcli disk listall | grep mfs-1 | grep sdd
```

13. Verify that the storage pool that contained the removed disk no longer exists.

    To display the list of storage pools, enter the following command:

```
/opt/mapr/server/mrconfig sp list -v
```

14. Remove the disk from the host.

    For more information about this step, refer to the documentation for your server and storage system.

    For example, if you are removing a disk from a physical server, remove the physical disk hardware from disk enclosure.

Listing Disk Information

This procedure describes how to list information about disks and storage pools in HPE Ezmeral Data Fabric on Kubernetes on HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights: Platform Administrator or Kubernetes Cluster Administrator access rights are required to download the admin kubeconfig file, which is needed to access Kubernetes cluster pods (see Downloading Admin Kubeconfig on page 486).

About this task

For disks that are a part of HPE Ezmeral Data Fabric on Kubernetes, the `maprcli disk list` command does not display the labels attached to a storage pool. This behavior differs from the output of the command when used on bare-metal implementations of HPE Ezmeral Data Fabric.

For storage pools in HPE Ezmeral Data Fabric on Kubernetes, you use the `mrconfig sp list -v` command.

Procedure

1. Access the pod that contains the disks.

   In the following example, the pod is `mfs-1`.

   ```
   kubectl exec -it mfs-1 -n mydfcluster -- /bin/bash
   ```

2. To display the list of storage pools, enter the following command:

   ```
   /opt/mapr/server/mrconfig sp list -v
   ```

   For example:

   ```
   /opt/mapr/server/mrconfig sp list -v
   ListSPs resp: status 0:2
   No. of SPs (2), totalsize 2990781 MB, totalfree 2978132 MB

   SP 0: name SP1, Online, size 1495390 MB,
   free 1483814 MB, path /var/mapr/edf-disks/drive_nvme_0, log
   200 MB, port 5660, guid 7ec6fc921e4312bb00617cf69603fbb9,
   clusterUuid -8211577265220812227--4311821546211161841, disks /var/
   mapr/edf-disks/drive_nvme_0 /var/mapr/edf-disks/drive_nvme_1 /var/mapr/
   edf-disks/drive_nvme_2, dare 0, label ssd:5
   SP 1: name SP2, Online, size 1495390 MB,
   free 1494317 MB, path /var/mapr/edf-disks/drive_nvme_3, log
   200 MB, port 5660, guid f6e3590203f5120400617cf69702d7a7,
   clusterUuid -8211577265220812227--4311821546211161841, disks /var/
   mapr/edf-disks/drive_nvme_3 /var/mapr/edf-disks/drive_nvme_4 /var/mapr/
   edf-disks/drive_nvme_5, dare 0, label hdd:6
   ```

Using `fsck` to Check for File System Inconsistencies

This procedure describes how use the `fsck` utility to check for and repair file system inconsistencies in a disk storage pool on HPE Ezmeral Data Fabric on Kubernetes on HPE Ezmeral Runtime Enterprise.

Prerequisites

Required access rights:

- Platform Administrator or Kubernetes Cluster Administrator access rights are required to download the admin kubeconfig file, which is needed to access Kubernetes cluster pods (see Downloading Admin Kubeconfig on page 486).
- You must be logged on as the root user on the nodes that contain the disk and on which the Kubernetes cluster is running.

About this task

Most disk failures can be identified and possibly remedied by running the `fsck` utility, which scans the storage pool to which the disk belongs and reports errors. The `fsck` utility can be used on an offline storage pool after a node failure, after a disk failure, a filesystem process crash, or to verify the consistency of data for suspected disk errors.

During this procedure, you place the pod in maintenance mode and take the storage pool offline. You restore operations at the end of the procedure.

Procedure

1. Use `kubectl exec` command to access the CLDB or MFS pod that contains the storage pool that you want to check.

   For example:

   ```
   kubectl exec -it cldb-0 -n mycluster1 -- /bin/bash
   ```

   If needed, you use the `kubectl get pods -n <cluster-name>` command to get the list of pods, and then determine the CLDB or MFS pod in which you want to run the `fsck` tool.

2. Place the pod in maintenance mode by entering the following command:

   ```
   sudo touch /opt/mapr/kubernetes/maintenance
   ```

3. Use the `mrconfig sp list` command to list the storage pools that are in the pod:

   In the following example, there is one storage pool, `SP1`, with path: `/dev/drive0`

   ```
   mrconfig sp list
   ListSPs resp: status 0:1
   No. of SPs (1), totalsize 224491 MB, totalfree 221235 MB

   SP 0: name SP1, Online, size 224491 MB, free 221235 MB, path /dev/drive0
   ```

4. Mark the storage pool as offline.

   For example:

   ```
   mrconfig sp offline /dev/drive0
   ```

5. Verify the storage pool is offline by examining the output of the `mrconfig sp list` command.

   For example:

```
mrconfig sp list
ListSPs resp: status 0:1
No. of SPs (1), totalsize 0 MB, totalfree 0 MB

SP 0: name SP1, Offline, size 2575449 MB, free 0 MB, path /dev/drive0
```

6. Run the `fsck` utility on the storage pool, examine the output, and identify and resolve any errors.

   For information about `fsck` and resolving errors, see the following in the HPE Ezmeral Data Fabric documentation (links open in a new browser tab or window):

   - fsck
   - Examining the Cause of Failure

   For example:

```
/opt/mapr/server/fsck -n SP1

Using logfile /opt/mapr/logs/fsck.log.2021-05-20.19:49:22.28795
tcmalloc: large alloc 26829914112 bytes == 0x55a10d184000 @
0x55a10945a710 0x55a1095c537c 0x55a10938ee7a
fs/common/daremgr.cc:194: Failed to open the file /opt/mapr/conf/
dare.master.key No such file or directory, err 2
tcmalloc: large alloc 26829922304 bytes == 0x55a74dd3c000 @
0x55a10945a710 0x55a1095c50fc 0x55a109336572

FSCK start (initialize storage pool and replay log) ...
Allocator init: 2515g (329711616 blocks) in 5031 groups
1: SG: f 99%: 0 [n 4198 6%, r 0] --> 7 [n 65536 100%, r 0]

FSCK phase 1 (initialize cache and verify log) ...

FSCK phase 2 and 3 (verify all containers and inodes) ...
  done with all containers 242 of 242 ...

FSCK phase 4 (verify namespace and orphanage) ...

FSCK phase 5 (verify allocation bitmap) ...

FSCK completed without errors.
```

7. Bring the storage pool online.

   For example:

```
 mrconfig sp online /dev/drive0
```

8. List the storage pools and verify the storage pool is online.

   For example:

```
mrconfig sp list
ListSPs resp: status 0:1
No. of SPs (1), totalsize 2506499 MB, totalfree 2505357 MB

SP 0: name SP1, Online, size 2506499 MB, free 2505357 MB, path /dev/drive0
```

9. Bring the pod out of maintenance mode by entering the following command:

```
sudo rm -f /opt/mapr/kubernetes/maintenance
```

10. (Optional) Verify that the Data Fabric cluster pods are operational.
    For example, you can execute the `edf report ready` command.

Replacing a Failed Disk

This procedure describes using the `mrconfig` to replace a failed disk that is part a storage pool on HPE Ezmeral Data Fabric on Kubernetes on HPE Ezmeral Runtime Enterprise.

Prerequisites

Prerequisites:

- **Required access rights:**
  - Platform Administrator or Kubernetes Cluster Administrator access rights are required to download the admin kubeconfig file, which is needed to access Kubernetes cluster pods (see Downloading Admin Kubeconfig on page 486).
  - You must be logged on as the root user on the nodes that contain the disk and on which the Kubernetes cluster is running.
- You have identified the disk that has failed and needs replacement.

About this task

During this procedure, you place the pod in maintenance mode and take the storage pool offline. After you replace the failed disk, you will use the `mrconfig` utility to recreate the storage pool, and then you will bring the storage pool and pod back online.

NOTE:

You must use the `mrconfig` utility to perform this task. Using the equivalent `maprcli` commands is not supported.

Procedure

1. Use `kubectl exec` command to access the CLDB or MFS pod that contains the storage pool that contains the failed disk.

   For example:

   ```
   kubectl exec -it cldb-0 -n myclusternode1 -- /bin/bash
   ```

   If needed, you use the `kubectl get pods -n <cluster-name>` command to get the list of pods, and then determine the CLDB or MFS pod in which you want to run the `fsck` tool.

2. Place the pod in maintenance mode by entering the following command:

   ```
   sudo touch /opt/mapr/kubernetes/maintenance
   ```

3. Use the `mrconfig sp list` command to list the storage pools that are in the pod:

   In the following example, there is one storage pool, `SP1`, with path: `/dev/drive0`

   ```
   mrconfig sp list
   ListSPs resp: status 0:1
   No. of SPs (1), totalsize 224491 MB, totalfree 221235 MB

   SP 0: name SP1, Online, size 224491 MB, free 221235 MB, path /dev/drive0
   ```

4. Make note of the other disk drives in the storage pool.

   Later in this procedure you will remove and then add the other disks in the storage pool that containes the failed disk. You can display the disks in the storage pool by entering the `mrconfig dg list <path>` command, where `<path>` is the path of the storage pool. In the output of the command, the drive paths of the disks in the group are listed at the end of the lines that start with `SubDG`.

5. Mark the storage pool as offline.

   For example:

   ```
   mrconfig sp offline /dev/drive0
   ```

6. Verify the storage pool is offline by examining the output of the `mrconfig sp list` command.

   For example:

   ```
   mrconfig sp list
   ListSPs resp: status 0:1
   No. of SPs (1), totalsize 0 MB, totalfree 0 MB

   SP 0: name SP1, Offline, size 2575449 MB, free 0 MB, path /dev/drive0
   ```

7. Remove the failed disk from the configuration.

   CAUTION:

   Removing a disk destroys the data on the disk, so ensure that all data on a disk is backed up and replicated before removing a disk.

   For example:

   ```
   mrconfig disk remove /dev/drive0
   ```

8. Replace the disk hardware. Follow the instructions for the system and disk you are replacing to remove the disk from the system and install the replacement disk.

9. Initialize the replaced disk by using the `mrconfig disk init` command.

   For example:

   ```
   mrconfig disk init -F /dev/drive0
   Disk guid: 7cc56e064fd1e1fe:60a6bfaa0693a2
   ```

10. Load the replaced disk by using the `mrconfig disk load` command.

    For example:

    ```
    /opt/mapr/server/mrconfig disk load /dev/drive0
    guid FEE1D14F-066E-C57C-A293-06AABFA66000
    dgguid 00000000-0000-0000-0000-000000000000
    ```

11. One disk at a time, use the `mrconfig` utility to remove, initialize, and load the other disks that were part of the storage pool that contained the replaced disk.
    After you finish this step, the replaced disk and the remaining disks in the storage pool have been initiated and loaded.

12. Use the `mrconfig dg create raid0`to create a disk group of type `raid0` that includes the disks in the storage pool.

    For example:

    ```
    /opt/mapr/server/mrconfig dg create raid0 /dev/drive0 /dev/drive1 /dev/
    drive2
    CreateDG disks(3) stripeDepth(0) layout(3)
    ```

13. Create a concatenated disk group with `mrconfig dg create concat` by specifying the primary drive.

    For example:

    ```
    mrconfig dg create concat /dev/drive0
    CreateDG disks(1) stripeDepth(0) layout(2)
    ```

    At this point, you can use the `mrconfig dg list` to see the layout of the disk group, and which disk is the primary disk. The primary disk can be used in other commands to refer to the disk group as a whole.

14. Make the storage pool from the newly-created disk group.

    For example:

    ```
    /opt/mapr/server/mrconfig sp make -F /dev/drive0
    ```

15. Make the storage pool from the newly-created disk group.

    For example:

    ```
    /opt/mapr/server/mrconfig sp make -F /dev/drive0
    ```

16. Bring the storage pool online.

    For example:

    ```
    mrconfig sp online /dev/drive0
    ```

17. List the storage pools and verify the storage pool is online.

    For example:

    ```
    mrconfig sp list
    ListSPs resp: status 0:1
    No. of SPs (1), totalsize 2510595 MB, totalfree 2509693 MB

    SP 0: name SP2, Online, size 2510595 MB, free 2509693 MB, path /dev/
    drive0
    ```

    The storage pool is identified by its path. The name of the storage pool is generated automatically, and is not necessarily retained when you recreate a storage pool for a given path.

18. Bring the pod out of maintenance mode:

    ```
    sudo rm -f /opt/mapr/kubernetes/maintenance
    ```

19. (Optional) Verify that the Data Fabric cluster pods are operational.
    For example, you can execute the `edf report ready` command.

HPE Ezmeral Data Fabric Database Administration

Administration of the HPE Ezmeral Data Fabric Database on HPE Ezmeral Data Fabric on Kubernetes is done using the `maprcli` command line interface (not the MCS). HPE Ezmeral Data Fabric Database administration is associated with tables, columns and column families, and table regions.

Administration of the HPE Ezmeral Data Fabric Database on HPE Ezmeral Data Fabric on Kubernetes is done using the `maprcli` command line interface (not the MCS). Regardless of whether the HPE Ezmeral Data Fabric Database table is used for binary files or JSON documents, the same types of commands are used with slightly different parameter options.

HPE Ezmeral Data Fabric Database administration is associated with tables, columns and column families, and table regions.

For more information about administering HPE Ezmeral Data Fabric Database tables and table replication, see Administering Tables in the HPE Ezmeral Data Fabric administration documentation.

Table Replication

Table replication allows you to configure an exact replica of the table on either a local or remote cluster. The source cluster sends updates to the replica as changes are made to the table. In HPE Ezmeral Data Fabric on Kubernetes, replication uses `maprgateway` pods to communicate securely between the source cluster and the gateway on the destination cluster. Both `primary-secondary` and `multi-master` replication are supported.

Table replication allows you to select a table on a source cluster and configure an exact replica of the table on either a local or remote cluster.

Replication uses the `maprgateway` pod or pods to communicate securely between the source cluster and the gateway on the destination cluster.

There are different kinds of replication and multiple ways to configure replication. HPE Ezmeral Data Fabric on Kubernetes supports `primary-secondary` and `multi-master` replication topologies:

- In a `primary-secondary` topology, you replicate one way from source tables to replicas. The replicas can be in a remote cluster or in the cluster where the source tables are located.
- In a `multi-master` replication topology, there are two primary-secondary relationships, with each table playing both the primary and secondary roles. Client applications update both tables and each table replicates updates to the other.

For more information about table replication, see the following in the HPE Ezmeral Data Fabric documentation:

- Understanding Replication
- Managing Table Replication

Configuring Table Replication

Configuring table replication between two Data Fabric clusters involves using `maprcli` commands to configure cross-cluster trust, register the destination gateway, and set up replication. This task contains an example of configuring simple `primary-secondary` table replication between two clusters.

About this task

This task configures simple `primary-secondary` table replication between two clusters.

Procedure

1. Configure cross-cluster trust between the two clusters.
2. Register the destination gateway on the source node by executing the `cluster gateway set` command.

   In the following example:

   - The name of the remote cluster is: `mydfcluster2`
   - The gateway service name is: `mip-ap77-n3-vm02.mip.your.company.net:10007`

   ```
   maprcli cluster gateway set -dstcluster mydfcluster2 -gateways
   "mip-ap77-n3-vm02.mip.your.company.net:10007"
   ```

   You can obtain the gateway service name for a cluster by executing the following command on the cluster:

   ```
   kubectl exec -i admincli-0 -n dataplatform -- /bin/bash -c "maprcli
   cluster gateway list"
   ```

3. Set up primary-secondary replication for an existing source table by executing the following command:

   ```
   maprcli table replica autosetup
   ```

4. Check the replication status by executing the following command:

   ```
   maprcli table replica list
   ```

Configuring Table Replication

```
echo mapr | maprlogin password
echo -e "create '/tmp/t1', 'cf1'\nput '/tmp/t1', 'r1', 'cf1:c1', 'v1'\nlist
'/tmp/ t1'\nscan '/tmp/t1'\nexit" | /usr/bin/hbase shell
maprcli table replica autosetup -path /tmp/t1 -replica /mapr/
mydfcluster2/t2 sleep 30
maprcli table replica list -path /tmp/t1
echo -e "put '/tmp/t1', 'r2', 'cf1:c1', 'v2'\nexit" | /usr/bin/hbase shell
echo -e "scan '/tmp/t1'\nscan '/mapr/mydfcluster2/t2'\nexit" | /usr/bin/
hbase shell
```

Related tasks

Creating Multiple Gateways for Table and Stream Replication on page 657
You can create multiple gateways for table and stream replication by increasing the number of `maprgateway` pods.

More information

Configuring Cross-Cluster Trust on page 652

Configuring Cross-Cluster Trust

NOTE: In this article, the term tenant refers to HPE Ezmeral Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

The `edftool` allows you to configure cross-cluster trust between either:

- One HPE Ezmeral Data Fabric cluster on bare metal and one HPE Ezmeral Data Fabric on Kubernetes cluster.

- Two HPE Ezmeral Data Fabric on Kubernetes clusters.

Trust allows mirroring between the two clusters and also allows tenants in one cluster to access data or tenants in the other cluster. All clusters listed in the `mapr-clusters.conf` file must have unique names in order to configure trust.

Compatibility

Cross-cluster operations are supported between HPE Ezmeral Runtime Enterprise clusters running the `dataplatform` operator with `mapr-core-6.2.0` and other clusters running:

- `dataplatform` operator with `mapr-core-6.2`
- `dataplatform` operator with `mapr-core-6.1`
- Bare-metal HPE Ezmeral Data Fabric clusters running release 6.1.0 or release 6.2.0

In this context, the term *bare-metal* means that HPE Ezmeral Data Fabric is deployed on either a Linux platform or a virtual machine.

About the edftool

The `edftool` simplifies complex security-related HPE Ezmeral Data Fabric tasks, including:

- Setting up trust between two clusters.
- Exporting the public certificates for each service.
- Exporting the private keys for each service.
- Generating certificate-signing requests for each service.
- Importing new certificates.

The `edftool` tool resides in the `admincli-0` pod, but the tool can also be run remotely from a Linux system with admin-level `kubectl` access to the cluster namespace. A client system running the `edftool` tool must have Keytool JDK utility, which is present if Java is installed. The tool uses SSH to log into both clusters and does the following:

- Generates login and service tickets on both clusters.
- Persists the cluster information for both clusters into the `ssl_trustore` and `mapr-clusters.conf` files.

  Each Data Fabric cluster has a configuration file, `mapr-clusters.conf`, that specifies the other Data Fabric clusters that this cluster can connect to. The file identifies the other clusters by specifying the cluster CLDB nodes.

  For more information about the `mapr-clusters.conf` configuration file, see **mapr-clusters.conf** in the HPE Ezmeral Data Fabric documentation.
- For each instance of HPE Ezmeral Data Fabric on Kubernetes, `edftool` generates a `kubectl` patch. The `kubectl` patch enables secrets to persist the trust information after a pod restarts.

Accessing the `edftool` help

1. Log into the `admincli-0` pod by executing the following command:

   ```
   kubectl exec -it -n <pod-namespace> admincli-0 -- /bin/bash
   ```

2. Execute the following command:

```
edftool
```

The tool displays the command help:

```
$ edftool
Tool to help with some of the more complex tasks in the Data Fabric
Usage:
edftool [command]
Available Commands:
cluster-trust Setup trust between two clusters
export-certs  Export the public certs of each serviceexport-keys
Export the private keys of each service
gen-csrs      Generate certificate signing requests for each service
help          Help about any command
import-certs  Import new certs (newly signed?)

Flags:
-h, --help    help for edftool

Use "edftool [command] --help" for more information about a command.
```

3. You can display detailed information about each command by executing the following command:

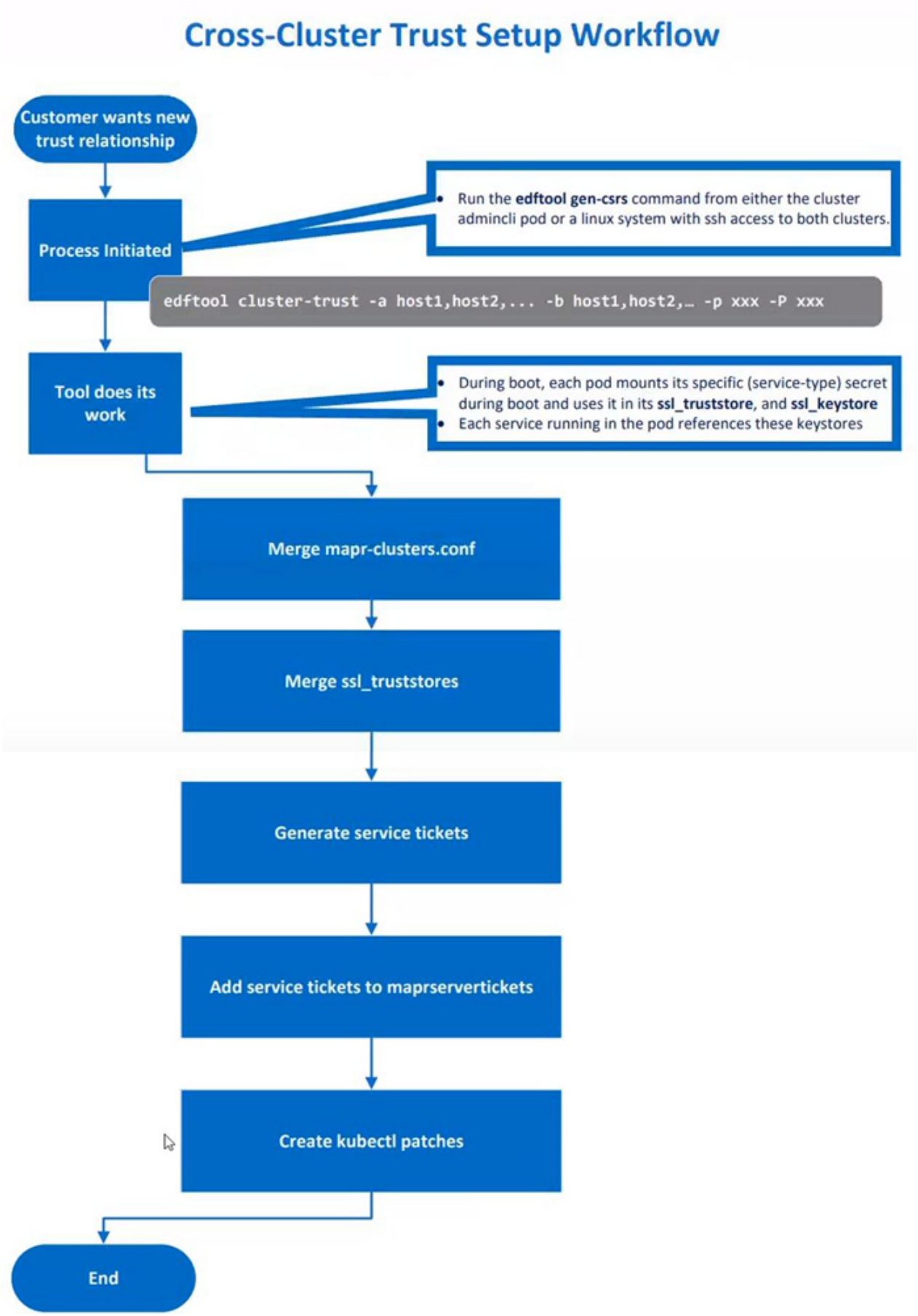
```
edftool <command> --help
```

4. For example:

```
edftool cluster-trust --help
```

Setting Up Cross-Cluster Trust

This illustration depicts the process of setting up cross-cluster trust:



To set up cross-cluster trust, do the following:

1. Execute the following command on either the Kubernetes cluster or the Data Fabric client where the `edftool` is installed:

   ```
   kubectl exec -it -n <pod-namespace> admincli-0 -- /bin/bash
   ```

2. Change to the `/tmp` directory to facilitate logging for the `edftool`:

   ```
   cd /tmp
   ```

3. Execute the `edftool cluster-trust` command with the required parameters.

   The following example sets up cross-cluster trust between an HPE Ezmeral Data Fabric on Kubernetes and a bare-metal HPE Ezmeral Data Fabric cluster:

   ```
   edftool cluster-trust -a 192.168.11.41,192.168.11.42,192.168.11.43 -p
   mapr
   -b 10.123.7.1, 10.123.7.2, 10.123.7.2 -P mapr -S 5000
   ```

   In the example:

   - The first three IP addresses are used for the nodes that the CLDB pods are running on in the Kubernetes cluster, followed by the Kubernetes cluster password. The Kubernetes cluster contains the table to be replicated and is thus the “local” cluster.
   - The next three IP addresses are the IP addresses for the CLDB nodes in the bare-metal cluster, followed by the bare-metal cluster password. The table will be replicated to the bare-metal cluster, which is the “remote” cluster.
   - `-S 5000` is the SSH port override for the local Kubernetes Data Fabric cluster.

     Port 5000 is the default port for containerized Data Fabric clusters. If both clusters were containerized Data Fabric clusters, then another parameter would be required for the remote cluster: `-s 5000.`

4. When prompted by the `edftool`, run the script specified by the prompt. The script applies the patch that enables the secrets to persist the trust information after a pod restart. Patch script files are named `k8_patch_cluster_<cluster-name>.sh`, where `<cluster-name>` is the name of the cluster to which the patch should be applied.

   For example:

   ```
   Please run the script './k8_patch_cluster_mydfcluster.sh' on a client with
   kubectl access to it and rights to modify secrets and configmaps in the
   dataplatform namespace.
   ```

   If you are establishing trust between a Kubernetes Data Fabric cluster and a bare-metal HPE Ezmeral Data Fabric cluster, then a patch is created for the Kubernetes Data Fabric cluster only.

   If you are establishing trust between two Kubernetes Data Fabric clusters, then two patches are created. You must do the following:

   a. Run one of the scripts in the `admincli` pod on the local Kubernetes cluster.

   b. Copy the other script to a node that has client access to the remote Kubernetes cluster.

   c. Run the script on the remote Kubernetes cluster.

5. Check the screen output for errors when the operation completes.

6. Log in to the remote cluster:

```
maprlogin password -cluster <cluster-name>
```

7. Execute the following command to view files and directories on the remote cluster, thereby ensuring correct trust configuration:

```
hadoop fs -ls /mapr/<cluster-name>
```

You should now be able log in to the remote cluster from the local cluster, set up a volume on one cluster and a mirror volume on the other cluster, and start replication. See Creating Remote Mirrors (link opens in a new browser tab/window).

Changes that require reconfiguration

You must reconfigure cross-cluster trust by running the edftool in the following circumstances:

- The IP address of a CLDB pod on the Kubernetes cluster changes.
- Additional CLDB pods are created.

To identify the full set of IP addresses for the CLDB nodes on the Kubernetes cluster, see the `cldbLocations` values for the `<cluster-name>-external-cm` in the `hpe-externalclusterinfo` namespace. The `dataplatform` operator automatically generates the `external-cm` ConfigMap to indicate the current values for various cluster parameters.

Creating Multiple Gateways for Table and Stream Replication

You can create multiple gateways for table and stream replication by increasing the number of `maprgateway` pods.

About this task

By default, HPE Ezmeral Runtime Enterprise 5.3 and later includes the configuration of a single `maprgateway` pod. For increased performance, high availability (HA), or both, you can create multiple gateways for table and stream replication either before or after the Data Fabric clusters are deployed.

Procedure

1. Increment the count parameter for the `maprgateway` pod.

```
maprgateway:
   count: 2
      image: maprgateway-6.2.0:202101192115C
```

2. Reapply the custom resource.

```
kubectl apply -f private-kubernetes/examples/picasso14/dataplatform/
full.yaml
```

3. Register the `maprgateway` instances by executing the `maprcli cluster gateway set` command.

   For example:

   ```
   maprcli cluster gateway set -dstcluster mydfcluster -gateways
   "maprgateway- svc.mydfcluster.svc.cluster.local"
   ```

   Setting the gateway to the gateway service only needs to be done once, because the service automatically handles subsequent changes to the number of gateways.

   If you are configuring gateways before deploying the cluster, you must wait until after the cluster is deployed before you can register the gateway instances.

   In HPE Ezmeral Data Fabric on Kubernetes, the gateway service name has the following format, where `<cluster-name>` is the name of the cluster:

   ```
   maprgateway-svc.<cluster-name>.svc.cluster.local
   ```

   You can obtain the gateway service name for a cluster by executing the following command on the cluster:

   ```
   kubectl exec -i admincli-0 -n dataplatform -- /bin/bash -c "maprcli
   cluster gateway list"
   ```

Example `maprgateway` Pod for database replication

This example shows and describes the fields of the `maprgateway` pod portion of an HPE Ezmeral Data Fabric on Kubernetes Custom Resource (CR) template.

HPE Ezmeral Runtime Enterprise includes a `maprgateway` pod that enables database replication for tables and streams.

Example `maprgateway` pod (yaml)

The following example is an excerpt of a Data Fabric Custom Resource (CR) template that HPE Ezmeral Runtime Enterprise reads when generating the CR for creating the Data Fabric cluster. The pod definition is also in the the file: `private-kubernetes/examples/picasso14/dataplatform/full.yaml`

The example shows a `maprgateway` pod configuration. In most cases, you can use the default values.

For example, to increase the number of `maprgateway` pods, you need only modify the value of the `count` field.

```
gateways:
      .
      .
      .
   maprgateway:
      count: 1
      image: maprgateway-6.2.0:202101192115C
      sshport: 5013
      requestcpu: "2000m"
      limitcpu: "8000m"
      requestmemory: 8Gi
      limitmemory: 8Gi
      requestdisk: 23Gi
      limitdisk: 46Gi
      loglevel: INFO
```

The following lists the `maprgateway` fields and their descriptions:

`count`	The number of replication gateway instances.
`image`	The image name and tag
`sshport`	The node port to use for external SSH requests.
`requestcpu`	Reserved pod CPU amount. Example: `2000m`
`limitcpu`	Maximum pod CPU amount. Example: `8000m`
`requestmemeory`	Reserved pod memory. Example: `6Gi`
`limitmemory`	Maximum pot memory. Example: `8Gi`
`requestdisk`	Reserved pod ephemeral storage space. Example: `23Gi`
`limitdisk`	Maximum pod ephemeral storage space. Example: `46Gi`

`loglevel`	Log level for the pod container. Values:
	`FATAL`
	`ERROR`
	`WARN`
	`INFO`
	`DEBUG`
	Default: `INFO`

Debugging and Troubleshooting

NOTE: In this article, the term tenant refers to HPE Ezmeral Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

This article contains the following information to help with debugging your environment if you run into errors or warnings during or after bootstrapping and applying CRs:

See also Troubleshooting Guide for Kubernetes Clusters (link opens an external website in a new browser tab/window).

Verifying Bootstrapping

- Execute the following command to verify that Data Fabric, tenant, and Spark operators are active by listing namespaces and pods:

  ```
  kubectl get ns
  ```

  This should list the `hpe-system` and `spark-operator` active namespaces.

- Execute the following command to verify that the Data Fabric and tenant operator pods are ready:

  ```
  kubectl get pods -n hpe-system
  ```

- If any pod is not up and ready, then execute either of the following commands to check the `State and Events` metrics for debugging:

  ```
  kubectl describe pod <pod-name> -n hpe-system
  kubectl logs <pod-name> -n hpe-system
  ```

- If the deployment created multiple pods, then use the `kubectl describe deployments`, Replica Sets (rs), and pods to look for any errors.

Verifying Data Fabric CR Deployment

Determine if pods are up and running,

1. List the pods in the Data Fabric cluster namespace to determine whether pods are up and running by executing the following command:

   ```
   kubectl get pods -n <Data-Fabric-cluster-namespace>
   ```

2. If any pod is not up and ready, then check the `State and Events` metrics for debugging by executing either of the following commands:

   ```
   kubectl describe pod <pod-name> -n <data-platform-cluster-namespace>
   ```

   ```
   kubectl logs <pod-name> -n <Data-Fabric-cluster-namespace>
   ```

   Wait until all pods show as **Running**, **Ready**, or **Completed**.

3. Confirm that the Data Fabric CR is working:

   a. exec to the `cldb` and `mfs` pods.

   b. Execute the following command:

   ```
   kubectl exec -it -n <cluster-name> cldb-0 /bin/bash
   ```

   c. Execute the following command to log in as the `mapr` user:

   ```
   su - mapr
   ```

d. Generate a `mapr` ticket using the `mapr` user credentials for the Data Fabric cluster namespace by executing the following command:

```
maprlogin password
```

Verifying Tenant CR Deployment

List the pods in your tenant namespace by running the following command:

```
kubectl get pods -n <tenant-name>
```

Verify that all pods listed in CR are Ready and Running with the expected number of instances. If any pod is not up and ready, run either of the following commands to check for the State and Events metrics for debugging:

```
kubectl describe pod <pod-name> -n <tenant-name>
```

```
kubectl logs <pod-name> -n <tenant-name>
```

Getting the IP addresses of MFS pods

The following command returns the IP addresses (internal and external) of the MFS pods:

```
maprcli node list -columns h
```

Applying RBAC Changes

The Tenant Operator supports RBAC authorization for any users or groups listed in the Tenant CR. For applying any changes in RBAC settings using the CR, you must delete the deployment by running the `kubectl delete -f <cr-tenant-xyz.yaml>` command and recreate using the `kubectl apply -f <cr-tenant-xyz.yaml>` command.

Troubleshooting

This section contains troubleshooting tips for the following issues:

FailedMount Warning

You may see the following warning when you run the describe command for pods:

```
"Warning FailedMount 8m20s (x6 over 8m35s) kubelet, worker2
MountVolume.SetUp failed
 for volume "client-secrets" : secret "mapr-client-secrets" not found
 Warning FailedMount 8m20s (x6 over 8m35s) kubelet, worker2
MountVolume.SetUp failed
 for volume "server-secrets" : secret "mapr-server-secrets" not found
 Warning FailedMount 8m20s (x6 over 8m35s) kubelet, worker2
```

```
MountVolume.SetUp failed
 for volume "ssh-secrets" : secret "mapr-ssh-secrets" not found"
```

This is normal and expected. Pods cannot mount the secrets until the `init` job has run. If you see timeouts because of these issues, then it is likely that resource constraints prevented scheduling the `init` job.

You may see the following warning when deploying the CR:

```
Warning FailedMount 51m (x7 over 51m) kubelet, aks-agentpool-34842125-0
 MountVolume.SetUp failed for volume "client-secrets" : secrets
 "mapr-client-secrets" not found
 Warning FailedMount 51m (x7 over 51m) kubelet, aks-agentpool-34842125-0
 MountVolume.SetUp failed for volume "server-secrets" : secrets
 "mapr-server-secrets" not found
```

You can ignore event messages like this because they do not prevent pods from launching.

FailedScheduling Warning

You may see the following warning when you execute the `describe` command for the pod:

```
Events:
Type    Reason            Age                From              Message
----    ------            ---                ----              -------
Warning Failed Scheduling 30m (x22 over 31m) default-scheduler 0/5 nodes
are available: 5
node(s) didn't have free ports for the requested pod ports.
```

This indicates a mismatch between the number of nodes in the cluster versus the number of instances of CLDB and the Data Fabric filesystem. Both require the same host ports and therefore cannot be deployed on the same node. The default installation requires three (3) CLDB nodes and two (2) Data Fabric filesystem nodes. This problem can also occur if all five nodes are present but one or more nodes cannot host a Data Fabric file system or CLDB container because the nodes are too small for the scheduler to schedule one of those pods.

Objectstore Pod Not Ready

The `objectstore` pod may not ready after a long time. For example:

```
Warning FailedMount 5m57s (x65 over 151m) kubelet, atsqa8c145.qa.lab Unable
 to mount volumes for pod
 "objectstore-0_mycluster(23af1481-41e2-11e9-b693-40167e367edb)": timeout
 expired waiting for volumes to attach or mount for pod
 "mycluster"/"objectstore-0". list of unmounted
volumes=[objectstore-csi-volume]. list of unattached volumes=[cluster-cm
status-cm replace-cm logs cores podinfo ldap-cm sssd-secrets ssh-secrets
 client-secrets server-secrets objectstore-csi-volume
 mapr-mycluster-cluster-token-f4krn]
```

If the `objectstore` pod remains stuck in the `init` state for more than 10 minutes, then manually delete and relaunch the pod by executing the following command:

```
kubectl delete pod -n <namespace> objectstore-0 pod "objectstore-0" deleted
```

CrashLoopBackOff or RunContainerError

1. Execute the following command to get the pods in the `kube-system` namespace:

   ```
   kubectl get pod -n kube-system
   ```

2. Check the pod `Status and Events` metrics by executing the following command:

   ```
   kubectl describe pod <pod-name> -n kube-system
   ```

   For example, if the pod named `kube-flannel-ds-amd64-v7qdt` failed, then execute the following command:

   ```
   kubectl describe pod kube-flannel-ds-amd64-v7qdt -n kube-system
   ```

If the pod is not ready because of Events errors or warnings, then recreate the cluster.

CLDB Running but Other Pods Waiting

The CLDB may be in the **Running** state but other pods are failing to initialize, waiting for CLDB, after applying `<cr-cluster-full.yaml>`. This occurs because the ObjectStore `init` container is unavailable and the pod fails to initialize, waiting for CLDB. Check the CLDB logs. If you see the message `Setting up disk failed`:

1. Execute the following command:

   ```
   kubectl get pods -n <cluster-name>
   ```

   You may notice that objectstore-0 has failed to initialize:

   ```
   objectstore-0 0/1 Init:0/1 0 1h
   ```

2. Execute the following command to get pod information:

```
kubectl describe pod objectstore-0 -n <cluster-name>
```

You might see something similar to:

```
Status: Pending
IP:
Controlled By: StatefulSet/objectstore
Init Containers: cldb-available:
Container ID:
Image: busybox
Image ID:
Port:
Host Port:
Command:
sh -c avail='UNAVAILABLE';
while \\\[ $avail -ne 'AVAILABLE' \\\];
do
echo waiting for CLDB;
sleep 10;
avail=\\\`cat /opt/mapr/kubernetes/status-cm/CLDB_STATUS\\\`;
done;
State: Waiting
Reason: PodInitializing
```

3. Get CLDB pod logs by executing the following command:

```
kubectl logs cldb-0 -n <cluster-name>
```

You may see something similar to:

```
2019/03/05 21:26:33 common.sh: \[INFO\] Setting up disk with:
 /opt/mapr/server/disksetup -F /opt/mapr/conf/disks.txt /dev/sdb
 failed. Error 16, Device or resource busy. Disk is used by some other
 module/process.
2019/03/05 21:26:37 common.sh: \[WARNING\]
 /opt/mapr/server/disksetup failed with error code 1... Retrying in 10
 seconds
2019/03/05 21:26:47 common.sh: \[INFO\] Setting up disk with:
 /opt/mapr/server/disksetup -F /opt/mapr/conf/disks.txt /dev/sdb
 failed. Error 16, Device or resource busy. Disk is used by some other
 module/process.
2019/03/05 21:26:52 common.sh: \[WARNING\]
 /opt/mapr/server/disksetup failed with error code 1... Retrying in 10
 seconds
2019/03/05 21:27:02 common.sh: \[INFO\] Setting up disk with:
 /opt/mapr/server/disksetup -F /opt/mapr/conf/disks.txt /dev/sdb
 failed. Error 16, Device or resource busy. Disk is used by some other
 module/process.
```

4. SSH to the cluster nodes.

5. Execute the following command:

```
disk -l
```

6. Identify a disk that is not used or free and change the disk values in the CR `simpleDeploymentDisks` list. For example:

```
kubectl delete -f cr-Data Fabricr-full.yaml
kubectl get ns
kubectl apply -f cr-Data Fabric-full.yaml
```

Object Store (S3 Gateway) Overview

The object store functionality provided for container-based HPE Ezmeral Data Fabric is similar to the S3 Gateway feature included in the bare-metal HPE Ezmeral Data Fabric, which is described in the HPE Ezmeral Data Fabric documentation in S3 Gateway (link opens in a new browser tab/window).

Deployment

To deploy the object store, you must create a Data Fabric cluster as described in Creating a New Data Fabric Cluster on page 611, and then deploy that cluster with the object store applied.

When you create a Data Fabric cluster by using the HPE Ezmeral Runtime Enterprise GUI, a single ObjectStore Zone is created by default.

This example shows a single zone object-store deployment:

```
apiVersion: hcp.hpe.com/v1
kind: DataPlatform
metadata:
  name: dataplatform
spec:
  baseimagetag: "202103030809C"
  imageregistry: gcr.io/mapr-252711
  environmenttype: hcp
  simpledeploymentdisks:
    - /dev/sdc
    - /dev/sdd
  disableha: true
  core:
    zookeeper:
      failurecount: 0
    cldb:
      failurecount: 0
    webserver:
      count: 1
    admincli:
      count: 1
  gateways:
    objectstore:
      imageregistry: gcr.io/mapr-252711
      image: objectstore-2.0.0:202103030809C
      zones:
        - name: zone1
          count: 1
          size: 10Gi
          fspath: ""
          hostports:
            - hostport: 9000
              nodeport: 31900
          requestcpu: "1000m"
          limitcpu: "4000m"
          requestmemory: 2Gi
          limitmemory: 2Gi
          requestdisk: 20Gi
```

```
            limitdisk: 30Gi
            loglevel: INFO
```

The object-store deployment uses the following fields:

- `imageregistry` – Registry where container images are stored.
- `image` - Image name and tag.
- `zones` - Object store zones.
- `name` - Zone name.
- `count` - Number of instances in the zone.
- `fspath` - Mount folder path, formatted as `/mapr/csi-volume/FOLDER_NAME`. If this property is not specified, then the path will be automatically set to `/mapr/csi-volume/objectstore-ZONE_NAME-svc` and will employ the service name for the zone.
- `hostports`- Object store node and service port. This value will overwrite the `port` value from `configmap`. The default port of the object store, 9000, can cause conflicts with Erlang RPC. In such cases, change the port of the object store to any free port.
- `nodeport` - external port on all cluster nodes, which will be used for forwarding requests to Objectstore instances in this zone. If nodeport is not specified, then forwarding from external port will not be configured for this zone.
- `size` - size of data fabric volume for Objectstore
- `loglevel` - Container logging level. This value will overwrite the `loglevel` value from configmap.

Configuration

Configure the object store by preparing a `configmap`. You can edit the `configmap` using an editor, such as:

```
KUBE_EDITOR="nano" kubectl edit configmap objectstore-cm -n dataplatform
```

For example:

```
minio.json:
----
{
     "fsPath": "/mapr/csi-volume//objectstore-0",
     "deploymentMode": "S3",
     "oldAccessKey": "",
     "oldSecretKey": "",
     "port": "9000",
     "logPath": "/opt/mapr/objectstore-client/objectstore-client-2.0.0/logs/
minio.log",
     "logLevel": 4
 }
objectstore.sample.logrotate:
----
/opt/mapr/objectstore-client/objectstore-client-2.0.0/logs/minio.log
{
    rotate 7
    daily
    compress
    missingok
```

```
    sharedscripts
    postrotate
        /bin/kill -HUP 'cat /opt/mapr/pid/objectstore.pid 2> /dev/null'
2> /dev/null || true
    endscript
}
```

The `minio.json` section of the `configmap` maps your configuration to the pod `minio.json` file. See S3 Gateway (link opens in a new browser tab/window). Verify that the `configmap` specifies all object store pods in all zones. Recreate all object store pods after modifying the `configmap`.

NOTE:

The values of `port` and `logLevel` in `configmap` will be overwritten by the values of `hostport` and `loglevel` value from deployment.

Scaling

The number of object-store instances in a zone can be scaled, as described in Upgrading and Patching the Data Fabric Cluster. The required pods are automatically started or terminated as needed after scaling instances up or down or adding a new zone.

HA Support

Objectstore 2.0.0 supports working in HA mode. Kubernetes makes HA available inside zones, and all pods inside one zone are thus mounted to the same folder. A separate service is created for each zone, and the service FQDN allows access to each instance. To check the services:

```
kubectl get svc -n dataplatform
```

Service FQDNs are formatted as follows:

```
objectstore-ZONE_NAME-svc.dataplatform.svc.YOUR_CLUSTER_DNS_PREFIX
```

If you use the MinIO client to make any administrative change to the object store configuration (such as adding new users, groups, policies, or notifications), then you must manually restart all instances (re-create pods) to avoid behavior collisions in different instances.

Limitations

- All object store zones and pods use one `configmap`.
- The `fspath` property overrides the `configmap` value. If the `fspath` property is not set, then the default value for the zone overrides the `configmap` value.
- Zone services provide only HA. They do not provide distributed mode and load balancing.
- The maximum number of object store instances is the same as the number of nodes in the cluster, because each object store requires an open port for listening connections.

HPE Ezmeral Data Fabric Event Store

HPE Ezmeral Data Fabric Event Store provides a reliable, global event streaming system that integrates publish and subscribe messaging to HPE Ezmeral Data Fabric on Kubernetes in HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Data Fabric Event Store provides a reliable, global event streaming system that integrates publish and subscribe messaging to HPE Ezmeral Data Fabric on Kubernetes in HPE Ezmeral Runtime Enterprise. Topics in HPE Ezmeral Data Fabric Event Store are grouped into streams, to which

administrators can apply security, retention, and replication policies. Combined with filesystem and HPE Ezmeral Data Fabric Database in HPE Ezmeral Data Fabric, using these streams enables organizations to create a centralized, secure data lake that unifies files, database tables, and message topics.

Implementation in HPE Ezmeral Runtime Enterprise

HPE Ezmeral Data Fabric Event Store is created in HPE Ezmeral Data Fabric on Kubernetes by default. It requires no additional process to manage, leverages the same architecture as the rest of HPE Ezmeral Data Fabric, and requires minimal additional management.

For information about the HPE Ezmeral Runtime Enterprise Kafka REST client interface to the Event Store, see Kafka REST Support on page 671.

The event store as implemented for HPE Ezmeral Data Fabric on Kubernetes is similar to the event store feature implemented in bare-metal HPE Ezmeral Data Fabric.

For more information about HPE Ezmeral Data Fabric Event Store, see HPE Ezmeral Data Fabric Event Store in the HPE Ezmeral Data Fabric documentation.

Erasure coding

In HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes supports storage tiers that use erasure coding for data. Erasure coding (EC) is a method of protecting data on lower-cost hardware that also reduces storage overhead in the range of 1.2x-1.5x. EC ensures that if data becomes corrupted, it can be reconstructed using information about the data that is present elsewhere.

In HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes provides rule-based automated data tiering functions to offload less frequently used data to specific nodes or low-cost hardware. Typically, erasure coding is used when storing "warm" tier data. Erasure coding is a method of protecting data on lower-cost hardware that also reduces storage overhead in the range of 1.2x-1.5x.

> **TIP:**
>
> For an excellent introduction to erasure coding, see this tech talk.

Erasure coding (EC) is a data protection method in which data is broken into fragments, expanded and encoded with redundant data pieces, and stored across a set of different locations or storage media. EC ensures that if data becomes corrupted, it can be reconstructed using information about the data that is present elsewhere.

A key decision involved in setting up erasure coding is selecting the erasure coding scheme. Considerations include how many nodes you can afford, how long you can tolerate waiting for a failed data node to be rebuilt, and how many failures you expect to occur.

Erasure coding schemes are expressed as numbers separated by the + (plus sign):

- When the scheme does not include local parity, two numbers are used. For example `10+2` indicates a scheme without local parity where `10` is the number of data nodes and `2` is the number of parity nodes. Generally these schemes are expressed as `m+n`.
- When the scheme includes local parity, three numbers are used. For example `10+2+2` indicates a scheme with local parity where `10` is the number of data nodes, followed by `2` local parity nodes, followed by `2` global parity nodes.

For erasure coding schemes without local parity, the recommended **total** number of nodes is `m+2n` (rather than `m+n`) to ensure Data-Fabric self-healing and proper operation after `n` failures. With `m+2n` nodes, `n` failures will self-heal with no operator intervention. For example, the recommended total number of nodes when you select a `3+2` erasure coding scheme is seven: Three data nodes and two times the number of parity nodes.

Although data can continue to be read after experiencing n failures with only `m+n` nodes, performance is significantly reduced because each read requires rebuilding data fragments. Also, manual intervention is required to protect the data from further failures. Data will not be erasure coded if only `m` nodes are available.

In erasure coding schemes with local parity, data nodes are divided into groups, with each group having a local parity node. Recovery from a failed node is faster because fewer nodes must be read when rebuilding the failed node.

For detailed information about erasure coding and a list of recommended coding schemes, see Erasure Coding Scheme for Data Protection and Recovery in the HPE Ezmeral Data Fabric documentation.

Data Tiering

Data tiering is the process by which data is moved among storage tiers as a way for a business to ensure that the appropriate data resides on the appropriate storage technology. Typical data tiering includes hot (replicated), warm (erasure coded), and cold (remote storage) tiers.

Data tiering is the process by which data is moved among storage tiers as a way for a business to ensure that the appropriate data resides on the appropriate storage technology.

A typical use of data tiering for a business is to balance performance, capacity, and cost:

- Data that is active and frequently accessed is referred to as "hot" data and is stored on the highest-performance storage technologies, which have a higher cost.
- Data that is less-frequently accessed can be stored on lower cost, lower performance storage technologies. This second tier of data is referred to as "warm" data.
- Data that is to be kept in long term storage for archiving, and yet still can be brought back to operational status, is referred to as "cold" data.

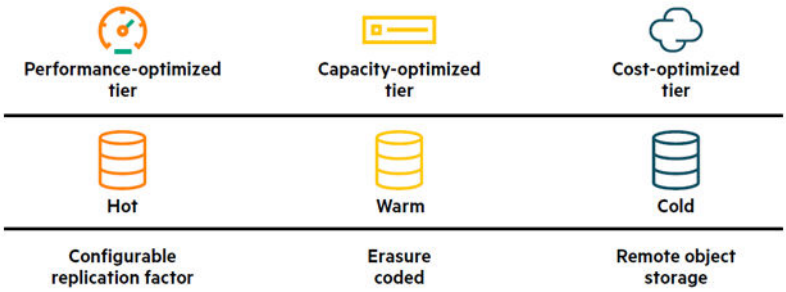


Figure 10: Performance Versus Cost for Data Tiers

For more information about data tiering, see Data Tiering in the HPE Ezmeral Data Fabric documentation.

MAST Gateway

The Data Fabric automated storage tiering (MAST) Gateway acts as the centralized entry point for all the tiering operations. CLDB assigns tiering-enabled volumes to MAST Gateways for processing all tiering operations for the volume.

The Data Fabric automated storage tiering (MAST) Gateway acts as the centralized entry point for all the tiering operations. CLDB assigns tiering-enabled volumes to MAST Gateways for processing all tiering operations for the volume.

If you are upgrading from a version of HPE Ezmeral Runtime Enterprise prior to version 5.3, you must enable the MAST gateway by adding the following to the gateways section of the dataplatform CR:

```
gateways:
 mast: true
```

For more information about the MAST gateway, see the following in the HPE Ezmeral Data Fabric documentation:

- Overview of the MAST Gateway
- Managing the MAST Gateway

Example: Creating a 10+2+2 EC volume using `maprcli`

This example shows the steps to use `maprcli` to create a 10+2+2 erasure coded (EC) volume in an HPE Ezmeral Data Fabric cluster.

Prerequisites

The erasure coding scheme in this example requires at least 14 worker nodes in the Data Fabric cluster. The erasure coding scheme, `10+2+2` requires 10 data nodes, two global parity nodes, and two local parity nodes.

About this task

This example shows the steps to use `maprcli` to create a volume in a Data Fabric cluster that uses 10+2+2 erasure coding (EC), which is an erasure coding scheme with local parity. This example includes creating custom tiering rules and offload schedules.

Procedure

1. Create a schedule.

   Schedules specify recurring points in time at which certain actions are to occur. In the following example, an erasure coding schedule specifies how often data is automatically offloaded to a different storage tier. Data is offloaded according to the tier rules.

   For more information about schedules, see **Creating a Schedule** in the HPE Ezmeral Data Fabric documentation.

   In the following example,

   - The name of the EC schedule is: `my_ec_schedule`
   - The offloading frequency is: `hourly`
   - The `id` is a unique number that is not one of the default schedules (`1`through `4`).

   ```
   maprcli schedule create -schedule '{ "id":5, "name":"my_ec_schedule",
   "inuse":0, "rules":[ { "frequency":"hourly", "retain":"10y" } ] }'
   ```

2. Create an erasure coded tier with default values.

   In the following example, the type parameter value, `ectier` specifies that an erasure coded tier is created.

   ```
   maprcli tier create -name mywarm_tier -type ectier
   ```

3. Create a tier rule.

   Tier rules define the criteria for offloading data.

   In the following example, the rule `my_rule` specifies data that is owned by the `mapr` user and was last modified 10 minutes (600 seconds) ago.

   ```
   maprcli tier rule create -name my_rule -expr "(u:mapr & m:600s)"
   ```

4. Create an erasure coded volume.

   The `ecscheme` parameter specifies the erasure coding scheme. In the following example, the erasure coding scheme is `10+2+2`, which specifies an erasure coding scheme with 10 data nodes, two local parity nodes, and two global parity nodes.

   For more information about erasure coding schemes, see **Erasure Coding Scheme for Data Protection and Recovery** in the HPE Ezmeral Data Fabric documentation.

   In the following example, the volume is enabled for data tiering by setting the `tieringenable` parameter to `true`. The `tieringenable` parameter must be used instead of the `ecenable` parameter because a tier name is also specified.

   ```
   maprcli volume create -name my_volume -path "/my_vol" -tieringenable
   true -tiername mywarm_tier -ecscheme "10+2+2" -ectopology "/
   data" -tieringrule my_rule -offloadschedule 5
   ```

5. (Optional) Force an offload to occur.

   ```
   maprcli volume offload -name my_volume
   ```

Kafka REST Support

HPE Ezmeral Runtime Enterprise version 5.3 enables a new `kafkarest` pod that provides a RESTful interface to HPE Ezmeral Data Fabric Event Store clusters to consume and produce messages and to perform administrative operations. The supported Kafka REST version is 5.1.2. For more information about the event store, see HPE Ezmeral Data Fabric Event Store.

HPE Ezmeral Data Fabric Database
In HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes supports HPE Ezmeral Data Fabric Database. HPE Ezmeral Data Fabric Databaseis an enterprise-grade, high-performance, NoSQL database management system you can use for real-time operational analytics. There are a few implementation differences between the bare-metal and HPE Ezmeral Data Fabric on Kubernetesimplementations of HPE Ezmeral Data Fabric Database.

HPE Ezmeral Data Fabric Database is an enterprise-grade, high performance, NoSQL (“Not Only SQL”) database management system. You can use it to add real-time operational analytics capabilities to big data applications. As a multi-model NoSQL database, it supports both JSON document models and wide column data models.

HPE Ezmeral Data Fabric Database can be used as both a document database and a wide-column database. As a document database, JSON documents are stored in a HPE Ezmeral Data Fabric DatabaseJSON table. As a wide-column database, binary files are in stored in HPE Ezmeral Data Fabric Database binary tables.

Implementation Differences

In most cases, HPE Ezmeral Data Fabric Database functions the same way in HPE Ezmeral Data Fabric on Kubernetes as it does in HPE Ezmeral Data Fabric on bare metal or virtual machines.

The differences are the following:

- HPE Ezmeral Data Fabric on Kubernetes does not support Hadoop.

  The database utilities that perform copy operations have the `mapreduce` or `directcopy` parameter set to `true` by default. To use these utilities in HPE Ezmeral Data Fabric on Kubernetes environments, you must set the `mapreduce` or `directcopy` parameter to `false`.

- When you set up table or stream replication using the CLI (`maprcli`), you can use the `replica autosetup` command only. See Setting Up Table Replication Using the CLI or Setting Up Stream Replication Using the CLI in the HPE Ezmeral Data Fabric documentation.

For more information about the HPE Ezmeral Data Fabric Database, see HPE Ezmeral Data Fabric Database in the HPE Ezmeral Data Fabric documentation.

HBase Binary Tables
In HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes supports HPE Ezmeral Data Fabric Database binary tables and the HPE Ezmeral Data Fabric implementation of HBase Shell.

In HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes supports HPE Ezmeral Data Fabric Database binary tables and the HPE Ezmeral Data Fabric implementation of HBase Shell, as described in the HPE Ezmeral Data Fabric documentation.

For more information about HBase binary tables in HPE Ezmeral Data Fabric, see:

HPE Ezmeral Data Fabric Database as a Wide-Column Database

For more information about HBase shell support, see:

HPE Ezmeral Data Fabric Database HBase Shell (Binary Tables)

Kafka REST Example CR and Field Descriptions

The following example from `private-kubernetes/examples/picasso14/dataplatform/full.yaml` shows the `kafkarest` pod:

```
gateways:
  nfs: true
  mast: true
  objectstore:
    image: objectstore-2.0.0:202101192115C
    .
    .
    .
  kafkarest:
    count: 1
    image: kafkarest-5.1.2:202101192115C
    sshport: 5015
    requestcpu: "2000m"
    limitcpu: "8000m"
    requestmemory: 4Gi
    limitmemory: 4Gi
    requestdisk: 20Gi
    limitdisk: 30Gi
    loglevel: INFO
```

The following table describes the fields in the `kafkarest` example:

Name	Description
`count`	Number of `kafkarest` pod instances.
`image`	Image name and tag.
`sshport`	Node port to use for external SSH requests.
`requestcpu`	Reserved pod CPU amount, in the format (`[1 - 9][0-9]+m`). For example: `2000m`.
`limitcpu`	Maximum pod CPU, in the format (`[1 - 9][0-9]+m`). For example: `8000m`.
`requestmemory`	Reserved pod memory amount, in the format (`[1 - 9]+Gi`). For example: `4Gi`.
`limitmemory`	Maximum pod memory amount, in the format (`[1- 9]+Gi`).For example: `4Gi`.

Name	Description
`requestdisk`	Reserved pod ephemeral storage space. Default is `20Gi`.
`limitdisk`	Maximum pod ephemeral storage space. Default is `30Gi`.
`loglevel`	Log level for the pod container. Value can be ERROR, CODE, INFO (default), or DEBUG. ERROR shows the least detail, while DEBUG shows the most detail.

Kafka REST ConfigMap

You can customize the `kafkarest` config files by editing the `kafkarest-cm` ConfigMap. Adding a file to the `data` section of `kafkarest-cm` copies the file contents to `/opt/mapr/kafka-rest/kafka-rest-5.1.2/config/` and overrides the existing files after pod restart.

The following example customizes the `/opt/mapr/kafka-rest/kafka-rest-5.1.2/config/kafka-rest.properties` file:

1. Add the customized `kafka-rest.properties` content in the `data` section by editing the existing `kafkarest-cm: $ kubectl -n dataplatform edit cm kafkarest-cm`:

```
...
ApiVersion: v1
data:
  kafka-rest.properties: |
    listeners=https://0.0.0.0:8082
    authentication.enable=true
    impersonation.enable=true
    schema.registry.enable=false
    streams.default.stream=/st
    headers.file=/opt/mapr/kafka-rest/kafka-rest-5.1.2/config/
headers.xml
    host.name=kafkarest-svc.dataplatform.svc.cluster.local
kind: ConfigMap metadata:
...
```

2. Restart the `kafkarest` pods:

```
$ kubectl delete pods -n dataplatform kafkarest-0
```

3. Check the `kafkarest` config file:

```
$ kubectl -n dataplatform exec --stdin --tty kafkarest-0 -- /bin/bash
$ cat /opt/mapr/kafka-rest/kafka-rest-5.1.2/config/kafka-rest.properties
listeners=https://0.0.0.0:8082
authentication.enable=true
impersonation.enable=true
schema.registry.enable=false
streams.default.stream=/st
headers.file=/opt/mapr/kafka-rest/kafka-rest-5.1.2/config/headers.xml
host.name=kafkarest-svc.dataplatform.svc.cluster.local
```

Customize Environment Variables and Kafka REST Proxy Heap Size

You can customize environmental variables for the `kafkarest` pod by using the `kafkarest` StatefulSet.

The following example customizes the Kafka REST Proxy heap size:

1. Edit the `kafkarest` StatefulSet:

```
$ kubectl edit statefulset kafkarest -n dataplatform
```

2. Add the variable name and value to the `env:` section:

```
spec:
- template:
  - spec:
    - containers:
     - env:
       - name: KAFKAREST_HEAP_OPTS
         value: -Xmx4096m
```

Kafka REST Pod Deployment Considerations

The following considerations apply when deploying the `kafkarest` pod:

- The REST Proxy does not store any state on disk.
- High Availability (HA) is not supported, but the `kafkarest-svc` Kubernetes service does implement load balancing. Your consumer client needs to handle exceptions returned from a failed consumer instance when deploying multiple `kafkarest` pods. It does this by attempting to create a new consumer instance using the `kafkarest-svc` address in order to switch to an active `kafkarest` pod.
- To avoid long GCs, HPE recommends running multiple `kafkarest` pods instead of using heap sizes larger than 8 GB.

Kafka REST Service Endpoints for Internal and External Clients

Kafka REST Service Endpoints for Internal Clients

Internal clients started on the Kubernetes pods can communicate with the Kafka REST Proxy using `kafkarest-svc` port 8082.

For example:

```
$ curl -<username>:<password> https://kafkarest-svc.<domain-name>:8082/
<uri-path> --cacert <truststore-file-path>/ssl_truststore.pem
```

The default value of `<truststore-file-path>` is `/opt/mapr/conf/`. You can customize this value.

The `ssl_truststore.pem` file is automatically generated during the cluster installation process. For internal clients, `/opt/mapr/conf/ssl_truststore.pem` is already mounted. Administrators can customize the path and file content.

Kafka REST Service Endpoints for External Clients

External clients started outside of the Kubernetes cluster can communicate with the Kafka RESTful API using `worker nodes hostname port 31882`.

For example:

```
$ kubectl get nodes
NAME          STATUS    ROLES     AGE      VERSION
master.lab    Ready     master    6d22h    v1.18.6
worker1.lab   Ready     worker    6d22h    v1.18.6
worker2.lab   Ready     worker    6d22h    v1.18.6
```

```
...

$ curl -<username>:<password> https://worker1.lab.<domain-name>:31882/
<uri-path> --cacert <truststore-file-path>/ssl_truststore.pem
```

The default value of `<truststore-file-path>` is determined by the location of the truststore file. You can customize this value.

The `ssl_truststore.pem` file is automatically generated during the cluster installation process. For external clients, you can retrieve the truststore from a Cluster Administrator.

Policy-Based Security

In HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes supports policy-based security (PBS), and the creation and management of security policies for Data Fabric objects through `maprcli` commands.

In HPE Ezmeral Runtime Enterprise, HPE Ezmeral Data Fabric on Kubernetes supports policy-based security (PBS), and the creation and management of security policies for Data Fabric objects through `maprcli` commands. For some tasks, you can also use the Control System (MCS).

A security policy is a classification that encapsulates security controls on data. Security controls define which users are authorized to access and modify data objects, whether to audit data operations, and whether to protect data in motion with wire-level encryption.

For example, consider a scenario in which one of your data classifications is sensitive employee data. With policy-based security, you can create a security policy named `employeeData`. As part of the security policy, one of the security controls you might define includes access control expressions (ACEs) that specify which users are allowed to access the employee data. You can then apply the security policy to relevant employee data objects. When you need to grant new users access to the employee data, you only need to modify that one security policy instead of modifying the ACEs defined on each of the employee data objects.

Examples of HPE Ezmeral Data Fabric objects that can be assigned ("tagged" with) security policies include the following:

- HPE Ezmeral Data Fabric file system volumes, directories and files
- HPE Ezmeral Data Fabric Database JSON tables, column families, and fields

For more information about policy-based security (PBS) for HPE Ezmeral Data Fabric, see Policy-Based Security in the HPE Ezmeral Data Fabric documentation.

Policy Based Security versus Centralized Policy Management

In HPE Ezmeral Runtime Enterprise, policy-based security and Centralized Policy Management have similar names but separate functions and scopes. The policy-based security feature applies to HPE Ezmeral Data Fabric objects. The Centralized Policy Management feature applies to Kubernetes cluster objects and can manage security policies from a central repository.

The policy-based security feature is separate from the Centralized Policy Management feature of the HPE Ezmeral Runtime Enterprise.

The policy-based security feature applies to HPE Ezmeral Data Fabric objects. A security-policy server in each of the security-policy Data Fabric clusters enforces the policies and manages the security-policy metadata in an internal volume named `mapr.pbs.base`.

The Centralized Policy Management feature, in contrast, is the fine-grained control of objects in your Kubernetes cluster, in which you express policies as YAML files (Kubernetes manifests), and apply them on the Kubernetes cluster. These YAML files can then be stored in a repository such as GitHub and applied to cluster objects automatically.

For more information about Centralized Policy Management, see Centralized Policy Management on page 336.

Setting Up Policy-Based Security
In HPE Ezmeral Runtime Enterprise, policy-based security (PBS) for HPE Ezmeral Data Fabric on Kubernetes is enabled by default. Before you can begin creating security policies, you must use `maprcli` commands to perform some set up tasks.

In HPE Ezmeral Runtime Enterprise, policy-based security (PBS) for HPE Ezmeral Data Fabric on Kubernetes is enabled by default. Before you can begin creating security policies, you must use `maprcli` commands to do the following:

1. Designate a global policy master.

   You must set one cluster as the global policy master before you can create security policies. The cluster set as the global policy master is the only cluster on which you can create or update security policies.

2. Set permissions for creating and managing security policies.

   To create security policies, an administrator must have cluster-level `cp` (create security policy) permission. By default, the `cp` permission is not assigned to all administrators. Administrators with cluster-level `a (admin)` permission can grant `cp` permission to themselves or other administrators.

For more information about these tasks, see Policy-Based Security and Policy-Based Security Quick Reference in the HPE Ezmeral Data Fabric documentation.

Creating, managing, and monitoring security policies for Data Fabric objects
Managing policy-based security on HPE Ezmeral Data Fabric on Kubernetes in HPE Ezmeral Runtime Enterprise is the same as managing policy-based security on bare-metal HPE Ezmeral Data Fabric clusters. Using `maprcli` or the Control Service (MCS), you can perform the same tasks that are described in the HPE Ezmeral Data Fabric documentation.

About this task

To create, manage, and monitor security policies on HPE Ezmeral Data Fabric on Kubernetes objects, log in to the `admincli` pod and access the `maprcli` interface. You use the same `maprcli` commands and you can perform the same tasks that are described in the HPE Ezmeral Data Fabric documentation.

For more information, see Policy-Based Security and Policy-Based Security Quick Reference in the HPE Ezmeral Data Fabric documentation.

Procedure

- You can use the `maprcli` interface for all tasks.

  Access the `maprcli` from the `admincli` pod.

  You use the same `maprcli` commands and you can perform the same tasks that are described in the HPE Ezmeral Data Fabric documentation.

- For some tasks, you can use the Control System (MCS).

  Access the `maprcli` from the `admincli` pod.

  You can perform the same tasks that are described in the HPE Ezmeral Data Fabric documentation.

Manual and Advanced Tasks

The topics in this section describe the manual tasks and information for advanced users of HPE Ezmeral Data Fabric on Kubernetes in HPE Ezmeral Runtime Enterprise.

Manual Deployment Workflow

HPE Ezmeral Runtime Enterprise versions 5.2 and later automate many of the processes described in this article. See:

- Creating a New Data Fabric Cluster on page 611
- Expanding a Data Fabric Cluster on page 616

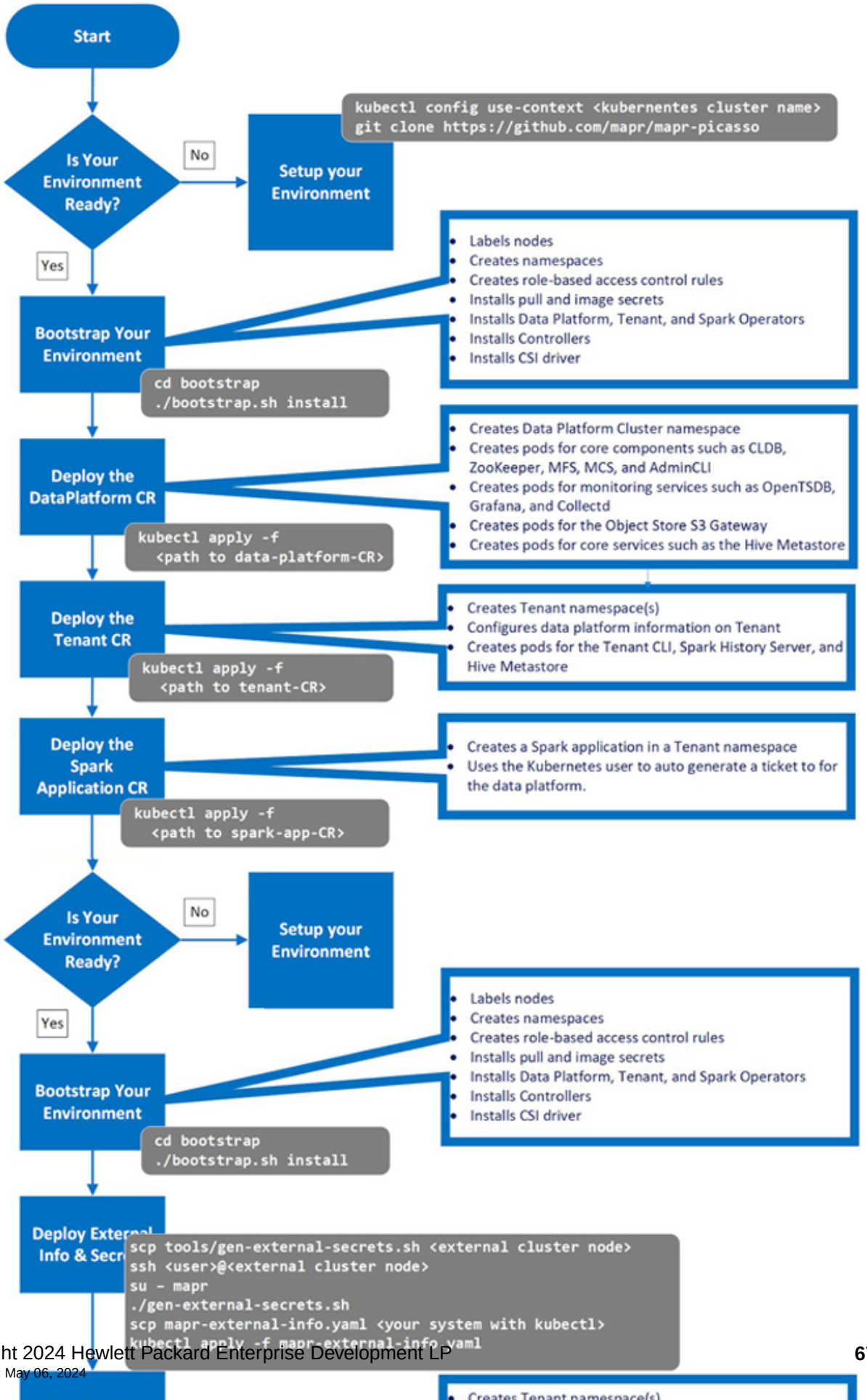
This information is presented for educational, maintenance, and debugging by users with advanced knowledge of HPE Ezmeral Data Fabric.

NOTE: In this article, the term tenant refers to Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

The general manual workflow to deploy HPE Ezmeral Data Fabric on Kubernetes is the following:

1. Configure `kubectl` to point to your Kubernetes environment, as described here (link open an external website in a new browser tab/window).
2. Run the bootstrap utility, as described in Manually Bootstrapping the Environment on page 679.
3. Manage the nodes and disks used by HPE Ezmeral Data Fabric clusters and tenants, as described in Manually Managing Nodes and Disks on page 679.
4. Either:
   - **No existing HPE Ezmeral Data Fabric on Kubernetes cluster:** Install a Data Fabric CR to create a Data Fabric cluster, as described in either Creating a New Data Fabric Cluster on page 611 or Manually Creating/Editing a Data Fabric cluster on page 694.
   - **With an existing HPE Ezmeral Data Fabric on Kubernetes cluster**: See CR Parameters.
5. Do one of the following:
   - If you are creating the tenant in a Kubernetes environment outside the storage cluster environment, then deploy external storage cluster host information and user, server, and client secrets, as described in Setting Up External Storage Cluster Secrets.
   - If the tenant is in the same environment as the storage cluster, procede to the next step.
6. Install one or more Tenant CRs to create new tenants, as described in Tenant CR Parameters.
7. Optionally, install a Spark CR for a Spark job.

The following diagram depicts this process:



Manually Managing Nodes and Disks

HPE Ezmeral Runtime Enterprise versions 5.2 and later automate many of the processes described in this article. See:

- Creating a New Data Fabric Cluster on page 611
- Expanding a Data Fabric Cluster on page 616

This information is presented for educational, maintenance, and debugging by users with advanced knowledge of HPE Ezmeral Data Fabric.

NOTE: In this article, the term tenant refers to Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

The bootstrap utility validates the nodes and disks for deploying Data Fabric clusters and tenants, and applies labels and annotations on the nodes for scheduling pods.

Labeling Nodes

You may optionally modify how detected nodes are used by changing the value of these labels and annotations before deploying a Data Fabric clusters and/or tenants by executing the following command:

```
kubectl describe node <nodename>
```

The bootstrap utility automatically uses all nodes in the Kubernetes cluster that do not have a `mapr.com/usernode` label unless you update the label for a node before using the bootstrap utility, as follows:

- To update the label to not install HPE Ezmeral Data Fabric on a node, execute the command:

  ```
  kubectl label node --overwrite <node_name> "mapr.com/usenode=false"
  ```

- To update the label to install HPE Ezmeral Data Fabric on a node (default option), execute this command:

  ```
  kubectl label node --overwrite <node_name> "mapr.com/usenode=true"
  ```

See Node Labels for additional information about this and other labels.

Labeling Disks

You must manually apply disk labels to every node on which you are installing HPE Ezmeral Data Fabric before running the bootstrap utility. The bootstrap script does provide a fake node labeller that can perform the labeling for you, but this feature may not label the disks as desired.

Manually Bootstrapping the Environment

HPE Ezmeral Runtime Enterprise versions 5.2 and later automate many of the processes described in this article. See:

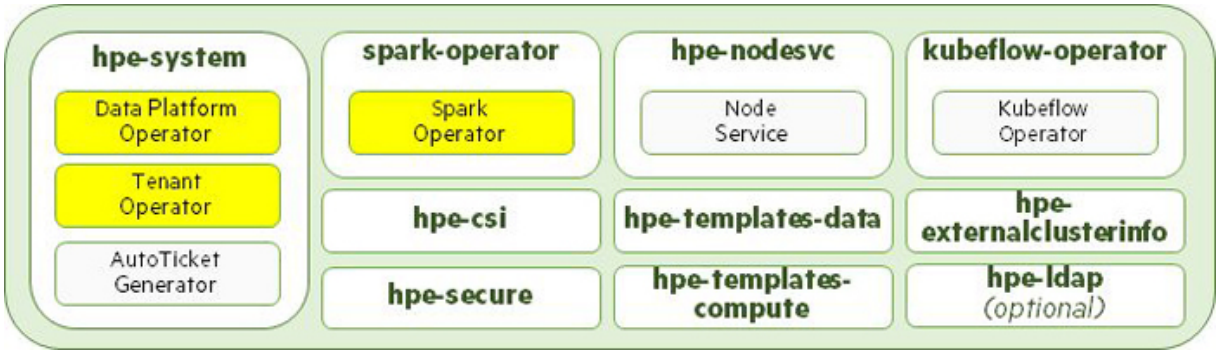
- Creating a New Data Fabric Cluster on page 611
- Expanding a Data Fabric Cluster on page 616

This information is presented for educational, maintenance, and debugging by users with advanced knowledge of HPE Ezmeral Data Fabric.

NOTE: In this article, the term tenant refers to Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

You must run the bootstrap utility with the install directive in your Kubernetes environment to install the operators for creating Data Fabric clusters, tenants, etc. The following diagram depicts:

- Namespaces created by the bootstrap utility (green text).
- Operators installed by the bootstrap utility (yellow boxes).
- Pods created by the bootstrap utility (light green and yellow).



The Kubernetes environment is ready for installing Data Fabric clusters, tenants, and compute engines once bootstrapping process concludes.

This article contains the following sections:

- **Overview:** Provides a high-level overview of the bootstrap installation process. See Overview.
- **Bootstrap Installation Process:** Provides detailed bootstrap installation instructions. See Bootstrap Installation Process.
- **Bootstrap Upgrade Process:** Describes how to use the bootstrap script to upgrade the Kubernetes environment. See Bootstrap Upgrade Process.
- **Bootstrap Uninstall Process:** Describes using the bootstrap script for uninstallation. See Bootstrap Uninstall Process.
- **Bootstrap Log Level:** Describes how to set the bootstrap log detail level. See Bootstrap Log Level.

Overview

This section provides an overview of the bootstrap installation process.

Step 1: Prepare for Bootstrapping

Perform the following on the system on which you plan to run the `bootstrap.sh` utility:

1. Have a Linux or Mac environment available. The bootstrap.sh script does not support Windows.
2. Verify that `kubectl` is installed on your system.
3. Set the `kubectl` context to the Kubernetes cluster you are deploying, as described here (link opens an external website in a new browser tab/window).
4. Download the `.zip` file containing the bootstrap utility and examples.
5. Run the `bootstrap.sh` utility located in the `bootstrap` directory when the cluster is ready for bootstrapping:

```
cd ./boostrap
```

6. If not already installed, install Python (either versions 2.7.5 - 2.7.99 or between v3.7.0 and 3.8.99; link opens an external website in a new browser tab/window). This is required to run `bootstrap.sh` . An error message appears if your Python version varies from the ranges prescribed here.
7. Install pip version 18.0 or later (link opens an external website in a new browser tab/window).
8. Install openSSL version v1.0.1 or later (link opens an external website in a new browser tab/window).
9. Install the latest version of the Python virtual environment by executing the following command:

```
pip install virtualenv
```

Step 2: Verify Cluster Readiness for Bootstrapping

HPE Ezmeral Data Fabric must be installed on an existing Kubernetes cluster. Perform the following procedure to verify that the Kubernetes cluster is configured and ready for bootstrapping:

1. Verify that `kubectl` is installed in your client machine and has CLI access to the Kubernetes cluster by executing the following command:

```
kubectl version
```

2. If the Kubernetes cluster is not a GKE cluster, then:
   - Verify that the Kubernetes cluster is running and that the current-context is set to the IP address of the master node by executing the following command:

```
kubectl config current-context
```

   - If the current-context is not set to the master node IP address, then execute the following command:

```
kubectl config set-context
```

3. Execute the command kubectl get nodes, and then verify all of the following:
   - The cluster has at least five (5) nodes.
   - The correct number of nodes is displayed.
   - One node is a `master`.
   - The node status is `ready`.
   - The cluster IP address is displayed and correct.
4. Verify that at least the following default Kubernetes namespaces are active on the cluster by executing the following command:

```
kubectl get namespace
```

```
  default
  kube-public
  kube-system
```

5. Ensure that `kubectl` is configured with superuser access to the Kubernetes cluster. Your user ID should have Kubernetes Cluster Administrator privileges in order to bootstrap. If needed, execute the following command to provide the permissions:

```
kubectl create clusterrolebinding user-cluster-admin-binding
--clusterrole=cluster-admin --user=<USER>
```

Step 3: Get Help

If needed, you can obtain general or command-specific help by executing the following commands:

- **General:** Execute the following command from the bootstrap directory:

```
$ ./bootstrap.sh
```

This command returns the following:

```


Bootstrap operations for MapR software
Usage: bootstrap.sh COMMAND [OPTIONS]
Commands:
  Install | uninstall | upgrade    Run command - must be supplied

  Options:
    --help List help for the specified command
```

```
Examples:
```

```
bootstrap.sh    install        Run installer
  bootstrap.sh    uninstall      Run uninstaller
  bootstrap.sh install --help    Get installation options
  bootstrap.sh uninstall --help  Get uninstallation options
  bootstrap.sh upgradel --help   Get bootstrap upgrade options
```

- **Command-specific:** Execute the following command to obtain help about that specific command;

```
bootstrap.sh <command> --help
```

Step 4: Run the Bootstrap Utility

You can run the bootstrap utility in one of the following modes:

- Prompt
- Record
- Headless

PROMPT Mode

This is the default, interactive mode if you do not specify a `--mode` parameter when invoking bootstrap.sh. In this mode, the utility prompts you for input, as described in Bootstrap Prompts, below. To run the utility in this mode, execute the following command:

```
./bootstrap.sh [install|upgrade|uninstall] [-m|--mode PROMPT_MODE]
```

To bootstrap the Kubernetes environment in `PROMPT_MODE`:

1. Go to the `bootstrap` directory by executing the following command:

```
cd bootstrap
```

2. Invoke the `bootstrap.sh` utility by executing the following command:

```
./bootstrap [install|upgrade|uninstall]
```

3. Enter appropriate responses to the prompts, which will vary depending on how you are invoking `bootstrap.sh`:
   - Bootstrap Install
   - Bootstrap Upgrade
   - Bootstrap Uninstall

RECORD Mode

This interactive mode prompts you for input for required settings, as described in Bootstrap Prompts, below, performs the bootstrapping function including namespace creation and resource installation, and creates a file containing a record of the settings that you can use later to invoke `bootstrap.sh` in `HEADLESS_MODE`.

To run the utility in this mode, execute the following command:

```
./bootstrap.sh [install|upgrade|uninstall] -m|--mode
RECORD_MODE -r|--response-file <response_file_name>
```

To bootstrap the Kubernetes environment in `RECORD_MODE`:

1. Go to the `bootstrap` directory by executing the following command:

```
cd bootstrap
```

2. Invoke the `bootstrap.sh` utility by executing the following command:

```
./bootstrap [install|upgrade|uninstall] -m|--mode
RECORD_MODE  -r|--responsefile <response_file_name>
```

3. Enter appropriate responses to the prompts, which will vary depending on how you are invoking `bootstrap.sh`:
   - Bootstrap Install
   - Bootstrap Upgrade
   - Bootstrap Uninstall

You may view sample `RECORD_MODE` output file by opening `bootstrap/sampleresponsefile.txt`.

HEADLESS Mode

This non-interactive mode uses a response file that was created either manually (see Bootstrap Install Settings), or automatically when the utility was invoked in RECORD_MODE.

To run the utility in this mode, execute the following command:

```
./bootstrap.sh [install|upgrade|uninstall] -m|--mode
HEADLESS_MODE -r|--response-file <response_file_name>
```

To bootstrap the Kubernetes environment in `HEADLESS_MODE`:

1. Go to the `bootstrap` directory by executing the following command:

   ```
   cd bootstrap
   ```

2. Invoke the `bootstrap.sh` utility by executing the following command:

   ```
   ./bootstrap [install|upgrade|uninstall] -m|--mode
   HEADLESS_MODE  -r|--responsefile <response_file_name>
   ```

Step 5: Post-Installation

After running a bootstrap installation, you may either:

- Manually create an HPE Ezmeral Data Fabric on Kubernetes cluster using the Data Fabric Custom Resource, as described in Manually Creating/Editing a Data Fabric cluster on page 694.
- Manually create Data Fabric tenants using the Tenant Custom Resource. You can choose to configure tenants to access data on either an existing external (on-prem or another supported environment), or an internalData Fabric cluster. See Manually Creating a New HPE Ezmeral Data Fabric Tenant on page 703.

Bootstrap Prompts

The following prompts appear when running `boostrap.sh` in either Prompt or Record mode:

1. Begin by going to the `bootstrap` directory and then run the `bootstrap.sh` utility with the install directive by executing the following commands:

   ```
   cd bootstrap
   ./bootstrap.sh install
   ```

2. To install CSI (see Container Storage Interface), enter `yes` (this is the default option) at the following prompt:

   ```
   >>> Install MapR CSI driver? (yes/no) [yes]:
   ```

3. Either:

   - Install the Tenant operator that manages tenants by entering `yes` (this is the default option) at the following prompt:

     ```
     >>> Install Computer? (yes/no) [yes]:
     ```

   - Enter no if you do not want to install the Tenant operator, and then proceed to the next step. See About Tenants for more information.

4. Either:

- Install the Data Fabric operator that manages internal Data Fabric clusters by entering `yes` (this is the default option) at the following prompt:

```
>>> Install Data Platform? (yes/no) [yes]:
```

- Enter no if you do not want to install the Data Fabric operator, and then proceed to the next step. See About HPE Ezmeral Data Fabric on Kubernetes on page 590 for additional information.

5. The bootstrap utility validates the Kubernetes environment it is configured to connect to. If there are no issues, you will see:

```
Looking good... connected to Kubernetes
```

6. Choose the correct user authentication option by entering one of the following:

```
>>> Choose an option ('EXISTING', 'NONE', 'EXAMPLE') [EXAMPLE]:
```

This is crucial for ensuring proper connectivity between tenants and internal or external storage clusters. The following options are available:

- `EXISTING` - Uses an existing LDAP server in your environment (recommended for production). See Adding Certificate Files During Bootstrap Installation on page 688. This option prompts you for user and group information, which must match existing user and group accounts that have been pre-configured for use by HPE Ezmeral Data Fabric. The user account prompts request username, password, and user ID. The group prompts request the `groupname`, and group ID. The group ID must be for the group in which the user account is configured. You will also be prompted for the following two common LDAP configuration files:
  - `ldap.conf` - Configures all OpenLDAP clients in the HPE Ezmeral Data Fabric environment.
  - `sssd.conf` - Configures all SSSD clients in the HPE Ezmeral Data Fabric environment.
- `None` - Choosing this option is not recommended. If you choose this option, the automatic ticket generator does not start, which can affect the ability to run applications such as Spark. This option does not perform any LDAP configuration. Instead, raw local Linux users must be configured everywhere in the HPE Ezmeral Data Fabric environment.
- `EXAMPLE` - Use an example OpenLDAP container installed in the `hpe-ldap` namespace. Default users and groups are pre-configured in the service, and the Data Fabric Kubernetes environment is configured to use this service. This option is good for testing but not recommended for production usage because it is not secure.

7. If you are using an air-gap Docker registry (see Kubernetes Air-Gap Requirements on page 834, Air Gap Tab on page 799, and Using an Air-Gapped Docker Registry on page 690), then select yes at the following prompt, otherwise select no, which is the default answer.:

```
>>> Use Airgapped Docker Registry? (yes/no) [no]:
```

8. As described in Managing Nodes and Disks, you must use the node labeller by answering yes at the following prompt. Answer no exits the script.

```
>>> Write fake labels to nodes for testing without HCP 5.1? (yes/no)
[yes]:
```

9. Enter yes to confirm that you want to start the bootstrapping process:

   We are now ready to install the basic components for running the HPE Ezmeral Data Fabric on Kubernetes…

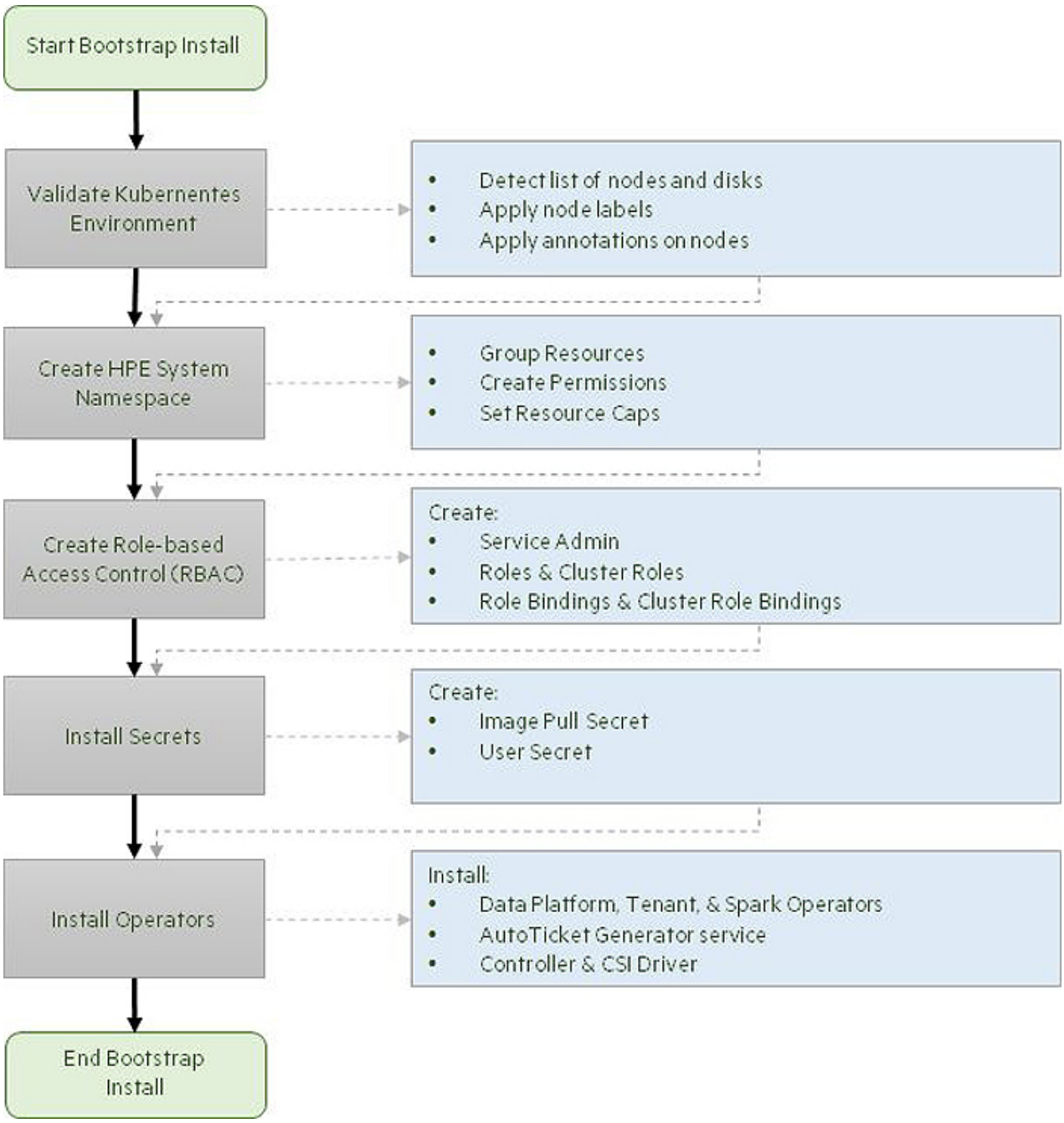
   ```
   >>> Continue with installation? (yes/no) [yes]:
   ```

The utility creates the necessary objects and then indicates the final installation status. For example:

```
This Kubernetes environment has been successfully bootstrapped for Data
Fabric  components can now be created via the newly installed operators
```

Bootstrap Installation Process

The following illustration depicts the bootstrap installation process:



The bootstrap process:

1. Validates the nodes in the Kubernetes environment to determine the list of nodes that are available for use, and applies labels and annotations on the available nodes. This validation determines information such as the number of nodes, their sizes, and the number and types of disks available on the nodes.

2. Creates several required Kubernetes namespaces (see Namespaces) that host various operators, group resources, Role-Based Access Control (RBAC) support files, and the CSI driver.

3. Sets up RBAC files, including:

   - A service account named `hpe-dataplatformoperator`.

   - A cluster role named `hpe-dataplatformoperator` that contains cluster-wide permissions.

- A cluster role binding also named `hpe-dataplatformoperator` that ties cluster roles to users and service accounts. See Using RBAC Authorizations (link opens an external website in a new browser tab/window) and Kubernetes Tenant RBAC.

4. Creates a system user secret that holds the sensitive information that you entered earlier. It also creates a pull secret that is used to pull images and operators from the repository and user secret.
5. Installs the config maps, operators (combinations of CRDs and Controllers), and creates the pods for the CSI driver.
6. Creates PVs and PVCs to allocate storage to the pods.

Bootstrap Install Settings

The following keys can be placed in a manually-generated response file that will be used to run the bootstrap utility in `HEADLESS_MODE` for the install directive:

- `CREATE_COMPUTE` - Specifies whether (`yes`) or not (`no`) to install the compute (tenant) components.
- `CREATE_STORAGE` - Specifies whether (`yes`) or not (`no`) to install the Data Fabric cluster components.
- `INSTALL_CSI` - Specifies whether (`yes`) or not (`no`) to install the HPE Ezmeral Data Fabric CSI Driver.
- `LDAP_OPTION` - Specifies the default authentication type choice for configuring the Data Fabric cluster and tenants. Value can be one of the following:
  - `EXTERNAL` - Configure and use an external LDAP service as the default for Data Fabric clusters and tenants.
  - `NONE` - Install non-LDAP settings to use raw (local) Linux users and configure this as the default for Data Fabric clusters and tenants.
  - `EXAMPLE` - Install the sample `hpe-ldap` OpenLDAP service and configure this as the default for Data Fabric clusters and tenants.
- `MAPR_USER` - Data Fabric cluster system user name, if `LDAP_OPTION` Is set to `EXTERNAL` or `NONE`.
- `MAPR_GROUP` - Data Fabric cluster system users group name, if `LDAP_OPTION` Is set to `EXTERNAL` or `NONE`..
- `MAPR_UID` - Data Fabric cluster system user ID, if `LDAP_OPTION` Is set to `EXTERNAL` or `NONE`.
- `MAPR_GID` - Data Fabric cluster system user's group ID, if `LDAP_OPTION` Is set to `EXTERNAL` or `NONE`.
- `USE_AIRGAP` - Whether (`yes`) or not (`no`) to use the Default docker repository for images or a different repository, such as in a local air-gapped environment.
- `AIRGAP_REGISTRY` - Name of the Docker repository, if `USE_AIRGAP` is set to `yes`. This value should be a URL that is accessible to the bootstrap system.
- `CONTINUE_INSTALL` - Whether (`yes`) or not (`no`) to continue installation after all applicable information has been entered.

Adding Certificate Files During Bootstrap Installation

During bootstrap installation, setting `LDAP_OPTION` to `EXISTING` allows you to specify a common `ldap.conf`, `sssd.conf`, and any user-provided certificates that will be used when initializing pods. The two `.conf` files allow you to customize aspects of your LDAP and SSSD configuration.

Example 1

This example sets `LDAP_OPTION` to `EXISTING` with the default `ldap.conf` and `sssd.conf` files plus a certificate file named `vault.pem`:

```
Please choose a user authentication configuration option from the three
listed:
```

```
EXAMPLE    ) Use an example OpenLDAP container (not for production use)
```

```
EXISTING ) Use an existing LDAP server in your environment
```

```
NONE    ) Use raw Linux users in each container (not recommended)
```

```
>>> Choose an option ('EXAMPLE', 'EXISTING', 'NONE') [EXAMPLE]: EXISTING
```

```
Please answer the following questions:
```

```
>>>What admin user account from your authentication provider would you like
to create
and register as the data admin during podinitialization? [custadmin]:
```

```
>>> What is admin user's uid? [7000]:
```

```
The data fabric uses common ldap.conf, sssd.conf, and any provided certs
when
bringing up pods.
```

```
 >>>Please provide an ldap.conf file to import [ldap.conf]:
```

```
 >>>Please provide an sssd.conf file to import [sssd.conf]:
```

```
Optionally, if your LDAP/SSSD setup is configured to verify TLS certs, enter
individual or bundle CA certificate files to include. Hit Enter (blank file
name)
when done.
```

```
>>> Certificate file to import: vault.pem

cert file added
```

```
>>> Another certificate file to import:
```

The `Certificate file to import` prompts allow you to specify either a path to a file or just the file name if the file is in your local directory. The certificates are added to the `hpe-secure` namespace.

Example 2

The following example `ldap.conf` file specifies `TLS_REQCERT` as a demand, which means that the CA in the `CERTS` directory should match the CA supplied by your LDAP server. This is just one example of a customization that you can implement using the `ldap.conf` file:

```
#
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE tlsshot.com

URI ldaps://tlsldap.myldap.com/

#SIZELIMIT    12

#TIMELIMIT    15

#DEREF    never

TLS_CACERTDIR    /etc/openldap/certs
TLS_REQCERT    demand

# Turning this off breaks GSSAPI used with krb5 when rdns = false
SASL_NOCANON    on
```

Using an Air-Gapped Docker Registry

The bootstrap utility now includes a prompt for using an air-gapped Docker registry rather than the default registry:

```
>>> Use Airgapped Docker Registry? (yes/no) [no]:
```

By default, bootstrap pulls Docker images from `cr.io/mapr-252711`. If you answer `yes` to this prompt, then you can choose a different repository. An air-gapped repository is a local or remote repository that you have prepared in advance that is physically isolated from the Internet or unsecured public networks. If you specify an air-gapped repository, then you must supply a URL or path that is accessible to the bootstrap system. You may also need to update any CRs that contain an entry for the Docker registry. For example:

```
spec:
  baseimagetag: "202008021206C"
  imageregistry: gcr.io/mapr-252711
  imagepullsecret: hpe-imagepull-secrets
  environmenttype: hcp
```

Using the Bootstrap Install `--setup_only` Option

The `bootstrap.sh install` command includes a `--setup_only` option that only installs the Python `virtualenv` components. This option is useful when you want to embed the bootstrapper in a Docker file, because it only installs the virtual environment components required for the bootstrapper to run. To use this option:

1. Go to the bootstrap directory
2. Invoke bootstrap.sh with the following options:

   ```
   ./bootstrap.sh install --setup_only
   ```

Bootstrap Upgrade Process

The bootstrap upgrade process takes only few minutes to upgrade operators and supporting files by only making changes for which a patch or new version is available. This process can be performed while the cluster is online. In general, you should perform a bootstrap upgrade before updating the Data Fabric cluster configuration.

A bootstrap upgrade is not always needed, such as if you recently installed HPE Ezmeral Runtime Enterprise. However, you should consider an upgrade if:

- You never performed a bootstrap upgrade
- You are unsure when the last bootstrap upgrade was performed.
- You know that new operators are available.
- The Data Fabric cluster is in an invalid state and upgrading the operators might help resolve an issue.

To perform the bootstrap upgrade:

1. Go to the `bootstrap` directory:

   ```
   cd bootstrap
   ```

2. Run the upgrade command:

   ```
   ./bootstrap.sh upgrade
   ```

3. The bootstrap script verifies that the current client and server versions are compatible and examines the operators and namespaces.
4. The bootstrap script detects the installed components and upgrades only those components that need upgrading. The script then asks whether (`yes`) or not (`no`; default) you want to use an air-gapped Docker registry:

   ```
   >>> Use Airgapped Docker Registry? (yes/no) [no]:
   ```

5. The bootstrap script prompts you whether (`yes`; default) or not (`no`) to continue the upgrade:

   ```
   >>> Continue with upgrade? (yes/no) [yes];
   ```

If you select yes to continue the upgrade, the bootstrap script begins applying updates, and then displays some of the components that were upgraded. For example:

```
Gathering Data Fabric cluster information...
Checking namespaces...
Checking operators...
data fabric installed: True
compute installed: True
ldap installed: True
spark installed: True
csi installed: True
Data Platform Operator:
    Pod: dataplatformoperator-b586c667d-24zft
    Image: clusteroperator-1.0.0:202007092203
    Create Time: 2020-07-27T16:55:45Z
    Status: Running
 Tenant Operator:
```

```
    Pod: tenantoperator-6c97ffdc5f-fcxz6
    Image: tenantoperator-1.0.0:202007092203
    Create Time: 2020-07-27T16:56:02Z
    Status: Running
 LDAP Pod
    Pod: ldap-0
    Image: ldap-6.2.0:202007092140C
    Create Time: 2020-07-27T16:55:43Z
    Status: Running
 Spark Operator:
    Pod: sparkoperator-686bbb7898-6kxhs
    Image: spark-operator-2.4.4:202006020640
    Create Time: 2020-07-27T16:56:05z
    Status: Running
 This Kubernetes environment has been successfully bootstrapped for the Data
 Fabric. Data Fabric components can now be created via the newly installed
 operators
```

Bootstrap Uninstall Process

Using the bootstrap script to uninstall HPE Ezmeral Data Fabric frees the resources in the environment and removes:

- All of the components (operators, CRDs, sample YAML files, secrets, and the CSI driver).
- The namespaces created in the Kubernetes environment.
- The service accounts, including the roles and role bindings.

Uninstall mode defaults to `PROMPT_MODE`, however you can also run it also in `RECORD_MODE` or `HEADLESS_MODE` if desired.

CAUTION: THE BOOTSTRAP UNINSTALL process DELETES NAMESPACES SUCH AS `MAPR-CONFIGURATION-CLUSTERS`, WHICH MIGHT CONTAIN USER-CREATED OBJECTS. IF YOU NEED TO PRESERVE ANY USER-CREATED OBJECTS, THEN BACK UP THE OBJECTS BEFORE PROCEEDING WITH THE UNINSTALL.

CAUTION: REMOVING A CRD REMOVES THE COMPONENTS DEPLOYED USING A CUSTOM RESOURCE ASSOCIATED WITH THAT CRD FROM THE KUBERNETES ENVIRONMENT. FOR EXAMPLE, REMOVING THE CRD FOR A STORAGE CLUSTER REMOVES THAT STORAGE CLUSTER FROM THE KUBERNETES ENVIRONMENT.

To uninstall using the bootstrap script:

1. Go to the bootstrap directory and then invoke `bootstrap.sh` with the `uninstall` directive by executing the following commands:

   ```
   cd bootstrap
           ./bootstrap.sh uninstall
   ```

2. Specify whether (`yes`) or not (`no`; default) to proceed with the uninstall:

   ```
   This will uninstall ALL HPE operators from your Kubernetes environment.
   This will
   cause all Tenants to be destroyed. They cannot be recovered!
   >>> Do you agree? (yes/no) [no]:
   ```

3. Specify whether (`yes`) or not (`no`; default) to uninstall the CSI driver:

```
>>> Remove the HPE CSI driver? (yes/no) [no]:
```

4. Specify whether (`yes`) or not (`no`; default) to uninstall the Data Platform (including any Data Fabric cluster):

```
>>> Remove Data Platform? (yes/no) [no]:
```

5. Specify whether (`yes`) or not (`no`; default) to uninstall the templates data:

```
>>> Remove the Data Platform Templates? (yes/no) [no]:
```

6. Specify whether (`yes`) or not (`no`; default) to uninstall the external cluster data:

```
>>> Remove the External Cluster Info? (yes/no) [no]:
```

7. Specify whether (`yes`) or not (no; default) to uninstall the `hpe-secure` data:

```
>>> Remove the Secure Namespace? (yes/no) [no]:
```

8. Specify whether (`yes`) or not (`no`; default) to uninstall the compute components (including tenants):

```
>>> Remove Compute? (yes/no) [no]:
```

9. Specify whether (`yes`) or not (`no`; default) to uninstall the compute templates data:

```
>>> Remove the Compute Templates? (yes/no) [no]:
```

The utility deletes the necessary objects and indicates the status of the uninstallation. For example:

This Kubernetes environment has the Data Fabric cluster successfully uninstalled

Bootstrap Uninstall Settings

The following key names can be automatically or manually placed in a response file for running the bootstrap utility in HEADLESS_MODE for the uninstall directive:

- `AGREEMENT` - Whether (`yes`) or not (`no`; this is the default option, which terminates the script) to perform the uninstall operation.
- `REMOVE_COMPUTE` - Whether (`yes`) or not (`no`) to uninstall the compute (tenant) components.
- `REMOVE_COMPUTE_TEMPLATES` - Whether (`yes`) or not (`no`) to uninstall secrets and configmaps in the compute templates namespace.
- `REMOVE_CSI` - Whether (`yes`) or not (`no`) to uninstall the Data Fabric CSI driver.
- `REMOVE_EXTERNAL_INFO` - Whether (`yes`) or not (`no`) to uninstall the secrets and configmaps in the external info namespace.
- `REMOVE_SECURE` - Whether (`yes`) or not (`no`) to uninstall the secrets and configmaps in the `hpe-secure` namespace.

- `REMOVE_STORAGE` - Whether (`yes`) or not (`no`) to uninstall the Data Fabric components.
- `REMOVE_STORAGE_TEMPLATES` - Whether (`yes`) or not (`no`) you want to uninstall secrets and configmaps in the storage templates namespace.

Bootstrap Log Level

If needed, you can use the bootstrap log located in `<bootstrap_dir>/src/common/mapr_conf/logger.yaml` to troubleshoot bootstrapping issues. You can also adjust the level of detail provided by the log, as follows:

1. Open `<bootstrap_dir>/src/common/mapr_conf/logger.yaml` for editing.
2. In the `handlers/logFileHandler` section, find the level setting.
3. Change the log level to one of these values:
   - `Level: !!python/name:logging.DEBUG` (most detailed logs)
   - `Level: !!python/name:logging.INFO`
   - `Level: !!python/name:logging.CODE`
   - `Level: !!python/name:logging.ERROR` (least detailed logs)

Manually Creating/Editing a Data Fabric cluster

NOTE: In this article, the term tenant refers to HPE Ezmeral Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

A Custom Resource file (CR) is the blueprint for creating a Data Fabric cluster. Data Fabric cluster creation therefore begins with either creating a new CR or editing an existing CR for the cluster. The CR specifies settings for the core components (such as CLDB, ZooKeeper, MCS, or Objectstore gateway) and the shared services (such as Hive Metastore) to be installed in the cluster.

If you are editing a Data Fabric cluster that was created in HPE Ezmeral Runtime Enterprise as described in Creating a New Data Fabric Cluster on page 611, then a new CR is automatically created for that cluster. This CR is the blueprint for how the Data Fabric cluster should be created by the DataPlatform operator. However, you can customize the CR to match your unique workflow usage using the information presented in this article.

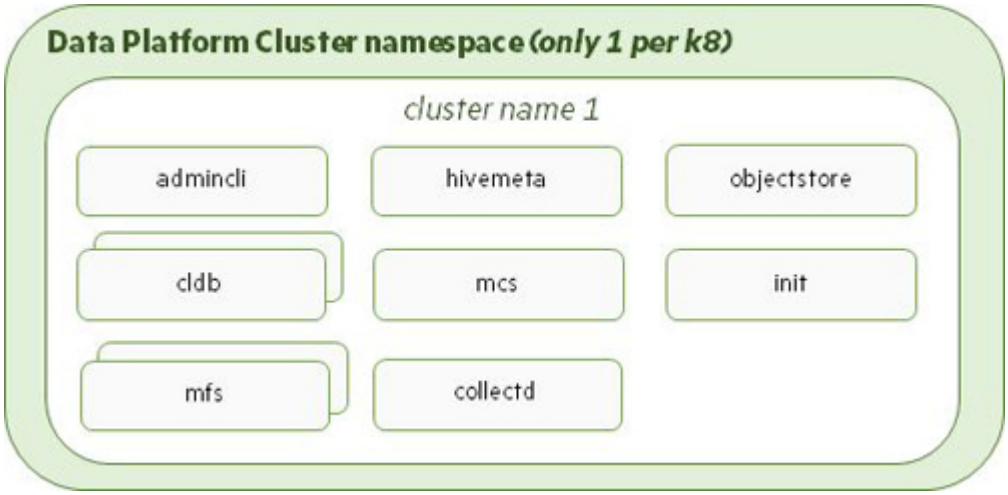
NOTE: If desired, you may either use or modify one of the sample Data Fabric CRs to create the Data Fabric cluster. Download sample files from the HPE Ezmeral dataplatform CR example files repository (link opens an external website in a new browser tab or window).

NOTE: Omitting or failing to specify a value for a required property assigns the default value, if any, to that property. Any property not documented in this article is ignored, even if a value is set for that property.

The DataPlatform operator reads this blueprint and creates the cluster. This operator consists of:

- A Custom Resource Definition (CRD) that contains the syntax and required properties for creating the Custom Resource.
- A Controller pod that uses the Custom Resource to build the Data Fabric cluster.

The DataPlatform operator can deploy one HPE Ezmeral Data Fabric on Kubernetes cluster per HPE Ezmeral Runtime Enterprise deployment. This cluster consists of the core, metrics, logging, object gateway, and shared services. The following illustration shows the default list of Data Fabric cluster components deployed by the DataPlatform operator:



Data Fabric CR Parameters

The Data Fabric CR can contain values for some or all of the following properties:

- `baseimagetag` - string - The tag to use for pulling all images from a common build. Use this tag to avoid specifying an individual tag for each image.
- `imageregistry` - string - Image registry location. This should be the full registry tag. See the sample CRs for the default Google cloud registry.
- imagepullsecret - string - Name of the secret that contains login information for the image repository.
- `environmenttype` - string - Kubernetes environment in which to deploy the storage cluster on. This must be set to `vanilla`.
- `simpledeploymentdisks` - array - List of disks to use in case of a simple deployment where all nodes have the same number of existing disks. For a more complex deployment, specify the disk information in the `diskinfo` object of `cldb` and `mfs`. The default value is:

  ```
  /dev/sdb
  /dev/sdc
  /dev/sdd
  ```

- `disableha` - boolean - Whether (`true`) or not (`false`; default) to disable High Availability (HA) enforcement for all pods. Disabling HA turns off HA pod guarantees, such as running only a single instance of CLDB or ZK instead of the minimum three. Enabling enforcement launches enough pods to ensure an HA cluster even if you request lower individual pod counts.
- `dnsdomain` - string - Kubernetes cluster DNS domain suffix to use. The default value is `cluster.local`.
- `loglocation` - string - Top-level writable directory where cluster logs are stored on the host machine. An extra level of hierarchy is added underneath this directory to separate information from a different cluster and cluster create times. The default value is `/var/log/mapr`. Container logs are stored in a subdirectory on the pod.
- `corelocation` - string - Optionally specifies the host location where core files for cluster pods are stored. The default value of `/var/log/mapr/cores` can be changed to any writable location on the host node.

- `podinfolocation` -string - Top-level directory where persistent pod information is stored. An extra level of hierarchy is added underneath this directory to separate information from a different cluster. The default value is `/var/log/mapr/podinfo`.
- `security` - object- Settings for installing a secure Data Fabric cluster. See Security Object Settings.
- `debuginfo` - object - Settings for debugging the logs. See Debuginfo Object Settings .
- `core` - object- Settings for the core Data Fabric cluster pods. See Core Object Settings.
- `monitoring` - object - Settings for monitoring services. See Monitoring Object Settings.
- `gateways` - object - Settings for gateway service pods. See Gateway Object Settings.
- `coreservices` - object - Settings for other Data Fabric services. See Core Services Settings.
- `externaldomain` - string - External domain of the host or NAT addresses used for external connections. Default is empty.

Security Object Settings

These are the security settings for all of the pods in the Data Fabric cluster. The `security` object in the Data Fabric CR must contain values for the following properties:

- `systemusersecret` - string - Name of the secret that contains system user information for starting the pods in Kubernetes.
- `disablesecurity` - boolean - Whether (`true`) or not (`false`; default) security should be disabled.
- `usedare` - boolean - Whether (`true`) or not (`false`; default) data-at-rest encryption must be enabled on the cluster. This must be set to `false` if `disableSecurity` is set to `false`.

DebugInfo Object Settings

Specify cluster-wide debugging settings that apply to all pods. Changing the debugging level for an individual pod overrides the cluster-wide settings. The `debugging` object in the Data Fabric CR must contain values for the following properties:

- `loglevel` - string - See Bootstrap Log Levels.
- `preservefailedpods` - boolean - Whether (`true`) or not (`false`; default) pods should not be allowed to restart in the event of a failure. Setting this value to true simplifies pod debugging, but causes the cluster to lose the native Kubernetes resilience that arises from pods restarting themselves when there is a problem.
- `wipelogs` - boolean - Whether (`true`) or not (`false`; default) to remove log information at the start of a container run. This setting is ignored if `hostid` is already present.

Core Object Settings

The `core` object in the Data Fabric CR must contain values for the following properties:

- `init` - object - Pod initialization settings for cluster key and certificate generation. The init pod generates initial cluster information including security keys based on the specification in the cluster CR. The cluster will not function if the `init` container does not start. Once started, the `init` container runs as a job and disappears after its work is finished. See Core Init Object Settings.

- `zookeeper` - object - ZooKeeper pod settings. Zookeeper contains critical cluster coordination information used by the MCS, `maprcli`, and CLDB. ZooKeeper pods run as part of a `statefulset`. See ZooKeeper Core Object Settings.
- `cldb` - object - CLDB pod settings. CLDB contains location information for all data stored in HPE Ezmeral Data Fabric. CLDB pods run as part of a statefulset. See CLDB Core Object Settings.
- `mfs` - object - File system pod settings. MFS pods physically store your data and run as part of a statefulset. See MFS Core Object Settings.
- `webserver` - object - Data Fabric Control System settings. The web-server containers run as part of a statefulset and host the admin interface. WebServer Core Object Settings.
- `admincli` - object - Admin client pod settings for administering the core data platform components. Admin CLI pods run as part of a statefulset. See AdminCLI Core Object Settings. Kubernetes Cluster Administrators log in to the Admin CLI to run various cluster maintenance tasks. See Admin CLI Pod.

Core Init Object Settings

The `init` object in the `core` object of the Data Fabric CR must contain values for the following properties:

- `image` - string - Image to use for the `init` pod container. The default value is `init-<mapr-version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `loglevel` - string - See Bootstrap Log Levels.

ZooKeeper Core Object Settings

The `zookeeper` object in the `core` object of the Data Fabric CR must contain values for the following properties:

- `failurecount` - integer - Number of failures to tolerate. If `disableHA` is enabled (set to `true`), then you can specify a value of `0` to create a single ZooKeeper instance. Otherwise, create 3 ZooKeeper instances for a single failure and increment by 2 for each additional failure to tolerate. For example, the default value of `1` creates 3 ZooKeeper instances, a value of 2, creates 5 ZooKeeper instances, and so on.
- `image` - string - Image to use for the pod container. The default value is `zookeeper-<mapr-version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `sshport` - integer - Node port to use to handle external SSH requests. The default value is `5000`.
- `loglevel` - string - See Bootstrap Log Levels.

CLDB Core Object Settings

`Diskinfo Settings` is deprecated from HPE Ezmeral Data Fabric on Kubernetes version 1.5, and user does not need to specify any details about the disks.

All disks present on the host will be categorized as `hdd/ssd/nvme` device and made available in the `cldb/mfs` pods under `/dev/mapr/edf-disks path`.

MFS Core Object Settings

The `mfs` object in the `core` object of the Data Fabric CR must contain values for the following properties:

- `image` - string - Image to use for the pod container. The default value is `mfs-<mapr-version>:<baseimagetag>`.
- `sshport` - integer - Node port to use to handle external SSH requests. The default value is `5001`.
- `hostports` - object - Externally-available pod ports. The default value for a single file system instance is `5660, 5692, 5724, 5756, 8660`. See HostPorts CLDB and MFS Group Object Settings.
- `requestcpu` - string - CPU amount to reserve for the pod, in the format `([1 - 9][0-9]+m)`. For example: `200m`.
- `limitcpu` - string - Maximum CPU for the pod, in the format `([1 - 9][0-9]+m)`. For example: `12000m`.
- `requestmemory` - string - Amount of memory to reserve for the pod, in the format `([1 - 9]+Gi)`. For example: `4Gi`.
- `limitmemory` - string - Maximum memory amount for the pod, in the format `([1 - 9]+Gi)`. For example: `4Gi`.
- `requestdisk` - string - Amount of ephemeral storage space to reserve for the pod. Default is `5Gi`.
- `limitdisk` - string - Maximum amount of ephemeral storage space to reserve for the pod. Default is `20Gi`.
- `loglevel` - string - See Bootstrap Log Levels.

MFS Group Object Settings

`Diskinfo Settings` is deprecated from HPE Ezmeral Data Fabric on Kubernetes version 1.5, and user does not need to specify any details about the disks.

All disks present on the host will be categorized as `hdd/ssd/nvme` device and made available the `cldb/mfs` pods under `/dev/mapr/edf-disks`.

HostPorts CLDB and MFS Group Object Settings

The `hostports` object in the `cldb and mfs:groups` object of the Data Fabric CR must contain values for the following properties:

- `mfs1port` - integer - First file system port. The default value is `5660`. For each additional instance, the port number is incremented by 1. That is, instance 0 will use `5660`, instance 1 will use `5661`, and so on for each additional instance.
- `mfs2port` - integer - Second file system port. The default value is `5692`. For each additional instance, the port number is incremented by 1. That is, instance 0 will use `5692`, instance 1 will use `5693`, and so on for each additional instance.
- `mfs3port` - integer - Third file system port. The default value is `5724`. For each additional instance, the port number is incremented by 1. That is, instance 0 will use `5724`, instance 1 will use `5725`, and so on for each additional instance.
- `mfs4port` - integer - Fourth file system port. The default value is `5756`. For each additional instance, the port number is incremented by 1. That is, instance 0 will use `5756`, instance 1 will use `5757`, and so on for each additional instance.

MFS Group DiskInfo Object Settings

The `diskinfo` object in the `mfs:groups` object of the Data Fabric CR must contain values for the following properties:

- `diskcount` - integer - Number of disks in this group. The default value is `3`. Pods for this service are not created on nodes that do not meet this requirement. This is ignored if `simpledeploymentdisks` information is specified in the Data Fabric CR.
- `disktype` - string - Type of disk in this group (`hdd`, `ssd`, `nvme`). The default value is `ssd`. Pods for this service will not be created on nodes that do not meet this requirement.
- `reducemfsrequirements` - boolean - Whether (`true`) or not (`false`; default) memory and CPU resources required by MFS should be reduced at the expense of DB performance.
- `storagepoolsize` - integer - Number of disks in the storage pool. This is configured during disksetup (link opens in a new browser tab/window). The default value is `0`, which uses a single storage pool. Any other number is passed to the `disksetup` utility as the stripe width. For example, if there are `10` disks and the storage pool size is `2`, then `5` storage pools with `2` disks each are created, and if the storage pools size is `5` the `2` storage pools of `5` disks each are created.
- `storagepoolsperinstance` - integer - Number of storage pools that an instance of the file system will manage. The platform launches multiple instances of the file system based on the specified number of storage pools. The default value is `0`, which sets the number of storage pools based on internal algorithms. A value greater than `32` generates an error.

WebServer Core Object Settings

The `webserver` object in the `core` object of the Data Fabric CR must contain values for the following properties:

- `count` - integer - Number of pod instances. At least one instance of the Data Fabric Control System (MCS) is required.
- `image` - string - Image to use for the pod container. The default value is `webserver-<mapr-version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `sshport` - integer - Node port to use to handle external SSH requests.
- `loglevel` - string - See Bootstrap Log Levels.

AdminCLI Core Object Settings

The `admincli` object in the `core` object of the Data Fabric CR must contain values for the following properties:

- `count` - integer - Number of pod instances. At least one instance of the admin CLI is required.
- `image` - string - Image to use for the pod container. The default value is `admincli-<mapr-version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `sshport` - integer - Node port to use to handle external SSH requests. The default value is `5003`.
- `loglevel` - string - See Bootstrap Log Levels.

Monitoring Object Settings

The `monitoring` object of the Data Fabric CR must contain values for the following properties:

- `monitormetrics` -boolean - Whether (`true`; default) or not (`false`) to enable monitoring of some cluster metrics using the installed monitoring services such as `collectd`, OpenTSDB, or Grafana.
- `collectd` - object -Collectd settings. Collectd runs as a deployment and collects various metrics from running pods. See CollectD Monitoring Object Settings.
- `opentsdb` - object - OpenTSDB settings. OpenTSDB pods run as part of a statefulset and hold the metrics generated by Collectd. See OpenTSDB Monitoring Object Settings.
- `grafana` - object - Grafana settings. Grafana pods run as part of a deployment and provide the interface for the metrics stored in OpenTSD. See Grafana Monitoring Object Settings.

CollectD Monitoring Object Settings

The `collectd` object in the `monitoring` object of the Data Fabric CR must contain values for the following properties:

- `disablecollectd` - boolean - Whether (`true`) or not (`false`; default) to disable collectd.
- `image` - string - The image to use for the pod container. The default value is `collectd-<version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `loglevel` - string - See Bootstrap Log Levels.

OpenTSDB Monitoring Object Settings

The `opentsdb` object in the `monitoring` object of the Data Fabric CR must contain values for the following properties:

- `count` - integer - Number of pod instances.
- `image` - string - Image to use for the pod container. The default value is `opentsdb-<version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `loglevel` - string - See Bootstrap Log Levels.

Grafana Monitoring Object Settings

The `grafana` object in the `monitoring` object of the Data Fabric CR must contain values for the following properties:

- `count` - integer - Number of pod instances.
- `image` - string - Image to use for the pod container. The default value is `grafana-<version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `loglevel` - string - See Bootstrap Log Levels.

last-updated: May 06, 2024

Gateway Object Settings

The `gateways` object of the Data Fabric CR must contain values for the following property:

- `objectstore` - object - Settings for the Data Fabric Object Store with S3-Compatible API. The Objectstore runs as a statefulset and allows S3/Minio API requests to data platform data. See Object Store Gateway Object Settings.

Object Store Gateway Object Settings

The `objectstore` object in the `gateways` object of the Data Fabric CR must contain values for the following properties:

- `count` - integer - Number of pod instances.
- `image` - string - Image to use for the pod container. The default value is `objectstore-<version>:<baseimagetag>`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `sshport` - integer - Node port on the node to use to handle external SSH requests.
- `hostports` - object - Externally-usable pod port.
- `loglevel` - string - See Bootstrap Log Levels.

Core Services Settings

Core services are additional pods that run in the Data Fabric cluster namespace, which means that core service pods are a single high availability cluster per Data Fabric cluster. Hive Metastore runs as a deployment and is currently the only non-gateway or monitoring cluster service that runs in the Data Fabric cluster namespace. This configuration allows Hive Metastore to be shared across tenants. If needed, Hive Metastore can also be run as a tenant service. The Hive Metastore object contains values for the following Hive Metastore properties:

- `count` - integer - Number of pod instances.
- `image` - string - Image to use for the pod container. The default value is `hivemeta-<version>:<baseimagetag>`.
- `useexternaldb` - boolean - Whether (`true`) or not (`false`; default) Hive Metastore should use an external DB instead of the embedded Derby DB.
- `externaldbserver` - string - DB server address to use for Hive Metastore. This value is ignored if `useexternaldb` is set to `false`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `sshport` - integer - Node port to use to handle external SSH requests.
- `loglevel` - string - See Bootstrap Log Levels.

Custom Configuration File Settings

In the `clusterCustomizationFiles` object in the cluster CR

- The custom configuration files specified using ConfigMaps in the CR are deployed in the `hpe-templates-data` namespace. Pods use the settings in the ConfigMaps when launching a service.

last-updated: May 06, 2024

- The custom configuration files specified using Secrets in the CR are deployed in the `hpe-secure` namespace. Pods use the settings in the Secrets when launching a service.

The `clusterCustomizationFiles` object in the cluster CR contains values for the following properties:

- `podSecurityPolicy` - string - Name of the pod security policy that should be used by the Data Fabric cluster. This should be in the `hpe-secure` namespace.
- `networkPolicy` - string - Name of the network policy that should be used by the Data Fabric cluster. This should be in the `hpe-secure` namespace.
- sslSecret - string - Name of the secret containing SSL certificates that should be used by the Data Fabric cluster. This should be in the `hpe-secure` namespace.
- `sshSecret` - string - Name of the secret containing SSH keys that should be used by the Data Fabric cluster. This should be in the `hpe-secure` namespace.
- `zkConfig` - string - Name of the ConfigMap containing ZooKeeper settings that should be used by the Data Fabric cluster. The default value is `zookeeper-cm`.
- `cldbConfig` - string - Name of the ConfigMap containing CLDB settings that should be used by the Data Fabric cluster. The default value is `cldb-cm`.
- `mfsConfig` - string - Name of the ConfigMap containing MapR file system settings that should be used by the Data Fabric cluster. The default value is `mfs-cm`.
- `webserverConfig` - string - Name of the ConfigMap containing Data Fabric Control System (MCS) settings that should be used by the Data Fabric cluster. The default value is `webserver-cm`.
- `collectdConfig` - string - Nme of the ConfigMap containing collectD settings that should be used by the Data Fabric cluster. The default value is `collectd-cm`.
- `opentsdbConfig` - string - Name of the ConfigMap containing OpenTSDB settings that should be used by the Data Fabric cluster. The default value is `opentsdb-cm`.
- `grafanaConfig` - string - Name of the ConfigMap containing Grafana settings that should be used by the Data Fabric cluster. The default value is `grafana-cm`.
- `objectstoreConfig` - string - Name of the ConfigMap containing MapR Object Store settings that should be used by the Data Fabric cluster. The default value is `objectstore-cm`.
- `hiveMetastoreConfig` - string - Name of the ConfigMap containing Hive Metastore settings that should be used by the Data Fabric cluster and tenant. The default value is `hivemetastore-cm`.
- `adminCLIConfig` - string - Name of the ConfigMap containing Admin CLI settings that should be used by the Data Fabric cluster. The default value is `admincli-cm`.
- `tenantCLIConfig` - string - Name of the ConfigMap containing Tenant Terminal settings that should be used by the tenant. The default value is `tenantcli-cm`.

Deploying HPE Ezmeral Data Fabric on Kubernetes

To create the HPE Ezmeral Data Fabric on Kubernetes environment:

1. Create or edit the custom resource (CR) YAML file for the Data Fabric cluster. The settings in the CR are described previously in this topic.

2. Execute the following command to use the CR to create the Data Fabric cluster:

```
kubectl apply -f <path-to-data-platform-cluster-custom-resource-file>
```

   Containers are created on the pods for running the Data Fabric cluster services when the Data Fabric CR is deployed.

3. Verify whether or not the Data Fabric cluster has been created by executing the following command. The cluster namespace is the cluster name that was specified in the CR:

```
kubectl get pods -n <data-fabric-cluster-namespace>
```

Manually Creating a New HPE Ezmeral Data Fabric Tenant

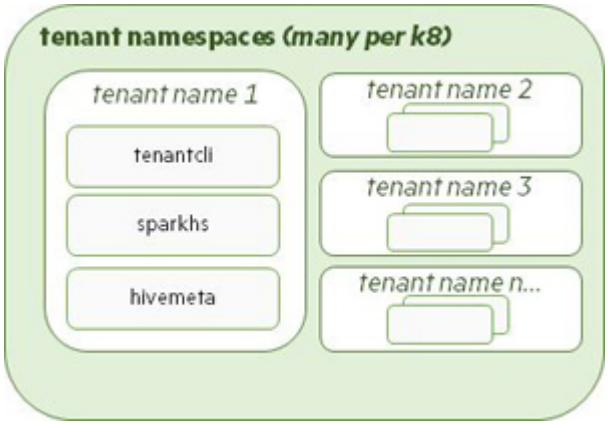
NOTE: In this article, the term tenant refers to HPE Ezmeral Data Fabric tenants (formerly "'MapR tenants") and not to Kubernetes tenants unless explicitly noted otherwise on a case-by-case basis.

Tenants connect to either an internal Data Fabric cluster or an external storage cluster.

- Creating a tenant that connects to an internal Data Fabric cluster begins with submitting a tenant CR in the same Kubernetes environment as the Data Fabric cluster.
- Creating a Tenant that connects to an external storage cluster begins with setting up and deploying the external storage cluster and user, server, and client secrets before submitting a Tenant CR. During the bootstrapping phase, the installer deploys the tenant operator that can be used to build the tenant CRs required to build the tenant namespaces in the Kubernetes environment. In this scenario, the external storage cluster must be visible from the pods running in the cluster where you plan to create the tenant. Verify connectivity by opening a shell to a running pod on the Kubernetes cluster and then pinging nodes on the storage cluster.

Tenant CR Parameters

The Tenant operator contains the tenant Custom Resource Definition (CRD), which validates the Tenant Custom Resource (CR) file that the Controller uses to create the tenant pods. The Tenant operator can deploy one or more instances of a tenant namespace in the Kubernetes environment to run compute applications, such as Spark, as shown in the following illustration:



A custom Tenant CR that specifies cluster connection settings and tenant resources should be created for each HPE Ezmeral Data Fabric tenant. See Defining the Tenant Using the CR.

last-updated: May 06, 2024

NOTE: If desired, you may either use or modify one of the sample Data Fabric CRs to create the Data Fabric cluster. Sample files are located in the `examples/picasso141/tenant` directory. These sample files are named `hctenant-*.yaml`. Sample files for connecting to an internal Data Fabric cluster have `internal` in the filename, while sample files for connecting to an external cluster have `external` in the filename.

NOTE: Omitting or failing to specify a value for a required property assigns the default value, if any, to that property. Any property not documented in this article is ignored, even if a value is set for that property.

Before deploying a tenant CR for an external storage cluster, you must first deploy the external cluster information and secrets that the tenant will use to connect. You may either:

- Run the `gen-external-secrets.sh` utility in the tools directory to gather this host information and generate various secrets.
- Manually create the required information. The following sample information templates in the `examples/picasso141/secrets` directory can help you collect this manual information:
  - **Secure external storage cluster:**`mapr-user-secret-secure-customer.yaml`
  - **Unsecure external storage cluster:**`mapr-user-secret-unsecure-customer.yaml`

You need not generate this information for an internal Data Fabric cluster because the system automatically obtains this information from the cluster namespace.

Defining the Tenant Using the CR

The Tenant CR should contain values for the following properties:

- `clustername` - string - Name of either the internal Data Fabric cluster or the external storage cluster to associate with this tenant.
- `clustertype` - string - This will be either `internal` (if the Data Fabric cluster is in the same environment as the Tenant) or `external` (if the storage cluster is outside the tenant Kubernetes environment).
- `baseimagetag` - string - The tag to use for pulling all the images.
- `imageregistry` - string - Image registry location.
- `imagepullsecret` - string - Name of the secret that contains the login information for the image repository.
- `loglocation` - string - Optional node location for storing tenant pod logs. This can be any writable location, subject to node OS restrictions. Default is `/var/log/mapr/<tenant>/`.
- `corelocation` - string - Optional node location for storing core tenant pod files. This can be writable location on the node. Default is `/var/log/mapr/<tenant>/cores/`.
- `podinfolocation` - string - (Optional top-level directory for storing persistent pod information, separated by cluster. This can be any writable location on the node, subject to node OS restrictions. Default is `/var/log/mapr/<tenant>/podinfo/`.
- `security` - object - See Security Object Settings.
- `debugging` -object - See Debug Settings.
- `tenantservice` - object - See Tenant Services Object Settings.

- `tenantcustomizationfiles` - object - See Tenant Customization File Object Settings.
- `userlist` - array - List of user IDs to add to the tenant Role-Based Access Control (RBAC).
- `grouplist` - array - List of group IDs to add to the Tenant RBAC.

Security Object Settings

These settings specify tenant security parameters.

- Tenants configured to use an internal Data Fabric cluster inherit security settings from the cluster.
- For tenants connecting to an external storage cluster, the storage cluster host information and user, server, and client secrets must be set up and deployed before deploying the tenant CR. See External Storage Cluster Secret Settings.

The `externalClusterInfo` object in the tenant CR must contain values for the following properties if the HPE Ezmeral Data Fabric storage cluster is not in the same environment as the tenant:

- `dnsdomain` - string - Kubernetes cluster DNS domain suffix to use. Default is `cluster.local`.
- `environmenttype` - string - Kubernetes environment on which to deploy the tenant. Value must be `vanilla`.
- `externalusersecret` - string - Name of the secret containing the system user info for starting the pods. This secret is pulled from the hpe-`externalclusterinfo`namespace and can be generated by `gen-external-secrets.sh` in the `tools` directory. Default is `mapr-user-secret`.
- `externalconfigmap` - string - Name of the secret containing the location of the external storage cluster hosts for communicating with the storage cluster. This information is pulled from the `hpe-externalclusterinfo` namespace. Default is `mapr-external-cm`.
- `externalhivesiteconfigmap` - string - Name of the `configmap` containing the properties from the external `hive-site.xml` file. This `configmap` can be generated by `gen-external-secrets.sh` in the `tools` directory if the storage cluster is not in the same environment as the tenant. This is available in the `hpe-externalclusterinfo` namespace. Default is `mapr-hivesite.cm`.
- `externalserversecret` - string - Name of the secret containing the external server secret info for communicating with the external storage cluster. This secret can be generated by `gen-external-secrets.sh` in the `tools` directory and is pulled from the `hpe-externalclusterinfo namespace`. Default is `mapr-server-secrets`.
- `externalclientsecret` - string - Name of the secret containing the client secret information for communicating with the external storage cluster. This secret can be generated by `gen-external-secrets.sh` in the `tools` directory and is pulled from the `hpe-externalclusterinfo` namespace. Default is `mapr-client-secrets`.
- `sshSecret` - string - Name of the secret containing the container SSH keys. Default is `mapr-ssh-secret`.

Debug Settings

The `debugging` object of theTenant CR must contain values for the following properties:

- `loglevel` - string - See Bootstrap Log Levels.

last-updated: May 06, 2024

- `preservefailedpods` - boolean - Whether (`true`) or not (`false`; default) to prevent pods from restarting in the event of a failure. Setting the value to true will allow you to debug pods more easily, but your cluster will lose the native Kubernetes resilience that comes from pods restarting themselves when there is trouble.
- `wipelogs` - boolean - Whether (`true`) or not (`false`; default) to remove log information at the start of a container run.

Tenant Services Object Settings

The `tenantservices` object of the Tenant CR specifies the following settings:

- `tenantcli` - Administration client launched in the tenant namespace.
- `hivemetastore` - Can be used in place of a Hive Metastore launched as a cluster-wide service. Access to this Hive Metastore is limited to users and compute engines in this tenant.
- `spark-hs` - Spark HistoryServer launched in the tenant namespace.

Each of these objects must contain values for the following properties:

- `image` - string - `tenantcli-6.1.0:<TIMESTAMP>. hivemeta-2.3:<TIMESTAMP>. spark-hs-2.4.4:<TIMESTAMP>`.
- `count` - integer - Number of pod instances. Default is `1`.
- `[sizing fields]` - strings - See Pod Sizing Fields.
- `loglevel` - string - See Bootstrap Log Levels.

Tenant Customization File Object Settings

The following custom configuration files specified using ConfigMaps in the CR are deployed in the `hpe-templates-compute` namespace and used by pods when launching a service:

- `hivemetastoreconfig` - string - Name of a configmap template containing Hive Metastore config files in hpe-config-compute. Default is `hivemetastore-cm`.
- `sparkhsconfig` - string - Name of a configmap template containing Spark HistoryServer config files in hpe-config-compute. Default is `sparkhistory-cm`.
- `sparkmasterconfig` - string - Name of a configmap template containing Spark Master config files in hpe-config-compute. Default is `sparkhistory-cm`.
- `sparkuiproxyconfig` - string - Name of a configmap template containing Spark UI Proxy config files in hpe-config-compute. Default is `sparkhistory-cm`.
- `sparkworkerconfig` - string - Name of a configmap template containing Spark Worker config files in hpe-config-compute. Default is `sparkhistory-cm`.

Creating and Deploying External Tenant Information

You must manually configure the external storage cluster host and security information when creating a tenant to connect to that cluster, including:

- External storage cluster CLDB and ZooKeeper host locations to which the tenant must connect.
- HPE Ezmeral Data Fabric user, client, and server secrets that must be created before the Tenant is created.

There are two ways to get and set this information:

- **Using a script:** See Automatic Method.
- **Manually:** See Manual Method.

Automatic Method

You can use the `gen-external-secrets.sh` utility in the `tools` directory to automatically generate a secret for both secure and unsecure storage clusters:

1. Determine whether Hive Metastore is installed on the storage cluster. You can find the node where Hive Metastore is installed by executing the following command:

   ```
   maprcli node list -filter [csvc==hivemeta] -columns name
   ```

2. Use `scp` or another method to copy `tools/gen-external-secrets.sh` to the Hive Metastore node on storage cluster. If Hive Metastore is not installed, the copy the script to any node on the storage cluster.

3. Start the tool by executing either of the following commands on the storage cluster as the admin user (typically `mapr`):

   - Unsecure external storage cluster:

     ```
     su - mapr
     ./gen-external-secrets.sh
     ```

   - Secure external storage cluster:

     ```
     ./gen-external-secrets.sh
     ```

4. When prompted, enter a name for the generated secret file. Default is `mapr-external-info.yaml`. If you are creating tenants that connect to different external storage clusters, then these secrets must have different names because they are all deployed in the same `hpe-externalclusterinfo` namespace. Each tenant CR must point to the correct secret, depending on the secret name.

5. When prompted, enter the username and password the HPE Ezmeral Data Fabric services will use for Data Fabric cluster administration. The default user is `mapr`.

   To obtain the default password, see Data Fabric Cluster Administrator Username and Password on page 600.

6. Specify whether the node is a Kubernetes storage node by entering either `y` (storage cluster is running on a Kubernetes environment) or `n` (storage cluster is running on a non-Kubernetes environment).

7. When prompted, enter the following user secret information:

   - **Server ConfigMaps:** Cluster host location. Default is `mapr-external.cm`.
   - **User secret:** Secret generated for MapR system user credentials. Default is `mapr-user-secrets`.
   - **Server secret (secure clusters only):** Secret generated for the MapR maprserverticket in `/opt/mapr/conf`. Default is `mapr-server-secrets`.
   - **Client secret (secure clusters only):** Secret generated for the `ssl_truststore` in `/opt/mapr/conf`. Default is `mapr-client-secrets`.

- **Hivesite configmap:** Information from the `hive-site.xml file`. Default is `mapr-hivesite.cm`. You may need change the settings in the generated file.

8. Copy the generated file to a machine that has a copy of `kubectl` and is able to communicate with the Kubernetes cluster hosting the external tenant.

9. Deploy the secret the `hpe-externalclusterinfo` namespace by executing the following command:

```
kubectl apply -f <mapr-external-secrets.yaml>
```

Manual Method

You can either:

- Modify the sample `mapr-external-info-secure.yaml` file (for a secure storage cluster) or `mapr-external-info-unsecure.yaml` file (for an unsecure storage cluster) in `examples/secrettemplates` to set values for the following properties.

- Create a custom file.

If you are creating or modifying your own cluster secret file, then the properties described in the following sections must be set in the secret files for the external storage cluster host, user, server, and client secret information:

- External Storage Cluster User Secret Settings

- External Server Secret Settings

- External Client Secret Settings

After creating the files, deploy the secrets in the Kubernetes environment. See Deploying the External Storage Cluster Secrets.

External Storage Cluster User Secret Settings

The cluster secret file must contain valid values for the following external storage cluster user secret properties:

- `name` - Name of the external storage cluster information.

- `namespace` - Namespace where the information is deployed.

- `MAPR_USER` - User that runs the Spark job. This must be Base64 encoded. Default is `mapr`.

- `MAPR_PASSWORD` - Password of the user that runs Spark job. This must be Base64 encoded. To obtain the default password, see Data Fabric Cluster Administrator Username and Password on page 600.

- `MAPR_GROUP` - Group of the user that runs the Spark job. This must be Base64 encoded. Default is `mapr`.

- `MAPR_UID` - User ID that runs the Spark job. This must be Base64 encoded. Default is `5000`.

- `MAPR_GID` - Group ID of the user that runs the Spark job. This must be Base64 encoded. Default value is `5000`.

External Server Secret Settings

The cluster secret file must contain valid values for the following external server secret properties:

- `maprserverticket` - Value of the `maprserverticket` automatically generated and stored in `/opt/mapr/conf` on the secure storage cluster. This must be Base64 encoded.
- `ssl_keystore.p12` - Value of the `ssl_keystore.p12` automatically generated and stored in `/opt/mapr/conf` on the secure storage cluster. This must be Base64 encoded.
- `ssl_keystore.pem` - Value of the `ssl_keystore.pem` automatically generated and stored in `/opt/mapr/conf` on the secure storage cluster. This must be Base64 encoded.

External Client Secret Settings

The cluster secret file must contain valid values for the following external client secret properties:

- `ssl_truststore` - Value of the `ssl_truststore` automatically generated for a secure cluster and stored in `/opt/mapr/conf` on the secure storage cluster. This must be Base64 encoded.
- `ssl_truststore.p12` - Value of the `ssl_keystore.p12` automatically generated and stored in `/opt/mapr/conf` on the secure storage cluster. This must be Base64 encoded.
- `ssl_truststore.pem` - Value of the `ssl_keystore.pem` automatically generated and stored in `/opt/mapr/conf` on the secure storage cluster. This must be Base64 encoded.

External Storage Cluster Host Information Settings

You can modify the `mapr-external-configmap.yaml` file in `examples/secrettemplates` to set values for the location of the service hosts on the external storage cluster or create your own custom file. The file must contain values for the following properties:

- `clustername` - Name of the external storage cluster. This must be Base64 encoded.
- `disableSecurity` - Whether (`true`) or not (`false`; default) security is disabled on the storage cluster.
- `cldbLocations` - Base64 encoded comma-separated list of CLDB hosts on the external storage cluster in the following format:

  ```
  hostname|IP[:port_no][,hostname|IP[:port_no]...]
  ```

- `zkLocations` - Base64 encoded comma-separated list of ZooKeeper hosts on the external storage cluster in the following format:

  ```
  hostname|IP[:port_no][,hostname|IP[:port_no]...]
  ```

- `esLocations` - Base64 encoded comma-separated list of Elasticsearch hosts on the external storage cluster in the following format:

  ```
  hostname|IP[:port_no][,hostname|IP[:port_no]...]
  ```

- tsdbLocations - Base64 encoded comma-separated list of openTSDB hosts on the external storage cluster in the following format:

  ```
  hostname|IP[:port_no][,hostname|IP[:port_no]...]
  ```

- `hivemetaLocations` - Base64 encoded comma-separated list of Hive Metastorage hosts on the external storage cluster in the following format:

  ```
  hostname|IP[:port_no][,hostname|IP[:port_no]...]
  ```

Deploying the External Storage Cluster Secrets

After creating the files, deploy the secrets and configmaps by executing the following command:

```
kubectl apply -f <mapr-external-cluster-info-file.yaml>
```

Deploying the Data Fabric Tenant

NOTE:

You must have either of the following before deploying a tenant:

- Running internal Data Fabric cluster. Wait until the cluster is fully started so that cluster settings can be configured on the tenant.
- Running external storage cluster. You must have already created information about that cluster in the `hpe-externalclusterinfo` namespace.

To create the Tenant namespace in the Kubernetes environment:

1. Either create a new tenant CR or modify an existing sample, as described in Defining the Tenant Using the CR.
2. Create the Tenant using the tenant CR by executing the following command:

   ```
   kubectl apply -f <path-to-tenant-resource-yaml-file>
   ```

3. Run the following command to verify that the tenant has been created by executing the following command:

   ```
   kubectl get pods -n <tenant-namespace>
   ```

You can now use the Spark operator to deploy Spark applications in the tenant namespace.

User-Configurable Data Fabric Cluster Parameters

This article describes two methods for configuring Data Fabric cluster parameters:

- Using a template CR
- Using bd_mgmt_config

Method1 : Template CR

This section refers to the Data Fabric Custom Resource (CR) template that HPE Ezmeral Runtime Enterprise reads when generating the CR for creating the Data Fabric cluster. Modifications to this CR template are effective if made before creating the Data Fabric cluster. Kubernetes Administrator users can access this template at:

```
/opt/bluedata/common-install/bd_mgmt/picasso_dataplatform_cr.cfg
```

This file is a partial CR specification where some fields have been templatized for use by HPE Ezmeral Runtime Enterprise. Advanced users may modify the non-templatized fields. You cannot change CLDB and MFS pod specifications here. Hewlett Packard Enterprise recommends limiting modifications to either enabling/disabling services or changing service resource allocations. You may want to save a copy of the original `/opt/bluedata/common-install/bd_mgmt/picasso_dataplatform_cr.cfg` file before making the modifications.

For example, set the following values to avoid bringing up pods related to monitormetrics services when a Data Fabric cluster is created:

```
spec:monitoring:monitormetrics=false
spec:monitoring:opentsdb:count=0
spec:monitoring:grafana:count=0
spec:monitoring:elasticsearch:count=0
spec:monitoring:kibana:count=0
```

NOTE: In HPE Ezmeral Runtime Enterprise 5.2, leaving `monitormetrics=true` in the CR template and subsequently changing it to `false` in the downloaded cluster CR might not stop the metrics pods.

After successful cluster creation, you may download the CR that was applied in the Kubernetes cluster using the HPE Ezmeral Runtime Enterprise web interface and can then either patch or modify and reapply it, as described in Upgrading and Patching Data Fabric Clusters on Kubernetes on page 621.

CAUTION:

All of the following cautions apply when modifying a template CR:

- Only advanced uses should modify the default values for keys related to HPE Ezmeral Data Fabric services in the template CR.
- The CR template is in YAML format. Preserve all indentations, spaces, and other punctuation.
- Disabling essential items (for example `admincli`) may cause the cluster to malfunction.
- When decreasing resource allocations for a service-pod, be sure to keep the resource allocation above the minimum required for that pod to function.

Method 2: Using bd_mgmt_config

Kubernetes Administrator users can modify configuration key values for a Data Fabric cluster in order to fine-tune that cluster.

CAUTION:

Key modification can cause performance loss and/or render the cluster inoperable. Do not modify the default key values unless you are familiar with the keys and how changing their values can affect the Data Fabric cluster.

Only change the value when modifying a configuration key. Always preserve the key name and format (e.g. Tuple, integer, string, etc.).

Environment Setup

To modify a key, you must first execute the following commands on the Controller host to set up the environment:

```
ERTS_PATH=/opt/bluedata/common-install/bd_mgmt/erts-*/bin
NODETOOL=/opt/bluedata/common-install/bd_mgmt/bin/nodetool
```

```
NAME_ARG=`egrep '^-s?name' $ERTS_PATH/../../releases/1/vm.args`
RPCCMD="$ERTS_PATH/escript $NODETOOL $NAME_ARG rpcterms"
```

Key Value Lookup

To look up the value of a configuration key, execute the following command:

```
$RPCCMD bd_mgmt_config lookup "<configuration_key_name>."
```

For example, the command:

```
$RPCCMD bd_mgmt_config lookup "datafabric_cldb_cpu_req_limit_percents."
```

Returns something similar to:

```
{35,75}
```

Modifying a Key Value

To change the value of a configuration key, execute the following command:

```
$RPCCMD bd_mgmt_config update "<configuration_key_name>. <value>."
```

For example, the command:

```
$RPCCMD bd_mgmt_config update "datafabric_cldb_cpu_req_limit_percents.
{50,70}."
```

Returns (if successful):

```
ok
```

Available Keys

The following configuration keys are available:

- `{datafabric_cldb_wakeup_timeout, 1500}`

  This integer value specifies how long the HPE Ezmeral Runtime Enterprise bootstrap add-on for HPE Ezmeral Data Fabric Kubernetes Edition must wait after Data Fabric CR creation/application until the cluster pods have come up, in seconds. Periodic status checks occur during this time period. Cluster creation fails if the cluster does not come up during this period.

- `{datafabric_cldb_cpu_req_limit_percents, {35, 75}}.`

  This tuple value influences the `requestcpu` and `limitcpu` for an intended CLDB pod specified in the Data Fabric CR. The `{X, Y}` tuple denotes the `{requestcpu, limitcpu}` values as percentages of the number of logical CPU cores in the system info of a CLDB node. The new or updated Data Fabric CR will specify X% of the node's logical CPU cores as the `requestcpu` for a CLDB pod and Y% as the `limitcpu` for a CLDB pod.

- `{datafabric_cldb_mem_req_limit_percents, {60, 75}}.`

  This tuple value influences the `requestmemory` and `limitmemory` for an intended CLDB pod specified in the Data Fabric CR. The `{X, Y}` tuple denotes the `{requestmemory, limitmemory}` values as percentages of the total available memory in a CLDB node's system info. The new or updated Data Fabric CR will specify X% of the node's total available memory as the `requestmemory` for a CLDB pod, and Y% as the `limitmemory` for a CLDB pod.

- `{datafabric_mfs_cpu_req_limit_percents, {40, 70}}.`

  This tuple value influences the `requestcpu` and `limitcpu` for an intended MFS Group specified in the Data Fabric CR. The `{X, Y}` tuple denotes the `{requestcpu, limitcpu}` values as percentages of the number of logical CPU cores in an MFS node's system info. The new or updated Data Fabric CR will specify X% of the node's logical CPU cores as the `requestcpu` for each MFS Group, and Y% as the `limitcpu` for each MFS Group.

- `{datafabric_mfs_mem_req_limit_percents, {60, 75}}.`

  This tuple value influences the `requestmemory` and `limitmemory` for an intended MFS Group specified in the Data Fabric CR. The `{X, Y}` tuple denotes the `{requestmemory, limitmemory}` values as percentages of the total available memory in an MFS node's system info. The new or updated Data Fabric CR will specify X% of the node's total available memory as the `requestmemory` for each MFS Group, and Y% as the `limitmemory` for each MFS Group.

- `{datafabric_hilowperf_disktype_capacity_ratio, {2, 3}}.`

  This configuration key is only relevant when nodes that can be used to schedule a Data Fabric cluster CLDB or MFS pod have multiple disk types (e.g. hard disk, SSD, or NVMe) among the node's persistent disks. Normally, HPE Ezmeral Data Fabric on Kubernetes only allows a node may to be represented by one disk type when it is considered for scheduling a CLDB or MFS pod.

  This tuple value denotes a capacity ratio, `x/y`, which guides ECP policy in how the `disktype` and `diskcount` are specified in the `diskinfo` section of the specification for a CLDB pod-set or an MFS group. The Data Fabric CR will specify a higher-performing disk type to represent a node, if that disk type is present in relatively-sizable capacity.

  If the capacity of the higher-performing disk type is `x/y` or more of the capacity of a lower-performing disk type (both disk types must be present among the node's persistent disks), then the node will be counted as having the higher `disktype`. The `diskcount` will equal the actual number of persistent disks of the higher-performing disk type that are present on the node. Thus, setting a low value for `x/y` (such as `1/100`) can help force a preference for the higher-performing disk type.

  If the higher-performing disk type is less than `x/y` of the lower-performing disk type, then the lower `disktype` will represent that node. If `m` disks of the higher type and `n` disks of the lower type are present in the node, the `diskcount` for the node will equal `m+n`, by convention.

  Adjusting this value allows a user to force a higher-performing or a lower-performing disk type to be used to represent nodes used for CLDBs or MFSs.

  - **Example 1:** If `{x, y}` is `{1, 2}`; a node's persistent disks include `p` NVMe disks totaling 500 GB; `q` SSDs, totaling 5 TB; `r` HDDs, totaling 20 TB: the node will be counted as having a `disktype` of `HDD` with a `diskcount` of the sum, `p+q+r`, of the counts of the disk types.

- **Example: 2:** If `{x, y}` is `{1, 2}`; a node's persistent disks include p NVMe disks, totaling 500 GB; q SSDs, totaling 800 GB; r HDDs, totaling 1.2 TB: the node will be counted as having a `disktype` of `NVMe` with a `diskcount` of p, the actual number of NVMe disks present.
- **Example: 3**: If `{x, y}` is `{1, 2}`; a node's persistent disks include p NVMe disks, totaling 200 GB; q SSDs, totaling 800 GB; `r` HDDs, totaling 1.2 TB: the node will be counted as having a `disktype` of `SSD` with a `diskcount` of `p+q`, the sum of the counts of the NVMe disks and SSDs present. In this example, changing `{x, y}` to `{1, 5}` would count the node as a `disktype` of `NVMe`, and a `diskcount` of p. Changing `{x, y}` to `{1, 1}` would count the node as a `disktype` of `HDD`, with a `diskcount` of `p+q+r`.

NFS Support

HPE Ezmeral Runtime Enterprise supports the NFSv3 service in MFS pods.

NOTE: NFSv4 is not currently supported.

Enabling or Disabling NFSv3

To enable NFSv3, add the `nfs: true` entry to the `dataplatform` YAML file. The `full.yaml` and `simple.yaml` example files include this entry by default.

```
gateways:
  nfs: true
  mast: true
  objectstore:
    image: objectstore-2.0.0:202101050329C
    zones:
      - name: zone1
        count: 1
        sshport: 5010
        size: 5Gi
        fspath: ""
        hostports:
           - hostport: 31900
        requestcpu: "1000m"
        limitcpu: "1000m"
        requestmemory: 2Gi
        limitmemory: 2Gi
        requestdisk: 20Gi
        limitdisk: 30Gi
        loglevel: INFO
```

NFSv3 support is disabled if either:

- You specify `nfs: false`.
- The `nfs: < true | false >` entry is not present.

Considerations

When working with NFSv3 in HPE Ezmeral Runtime Enterprise:

- The default share is `/mapr`. To change the default share, see Customizing NFS, below.
- If the MFS pod hosting the NFSv3 service stops and then restarts on a different node, then the NFSv3 mounts will stop working.
- The default NFS server port is 2049. Do not change the default port.

- You must mount the NFS drive manually; auto-mount is not supported.

Known Issues

The following issues are known to exist with `maprcli` commands for NFS:

- `maprcli setloglevel nfs -loglevel` or `maprcli trace info` or `tracelevel: DEBUG` - Command does not work or returns an exception. The default trace levels are:
  - `NFSD : INFO`
  - `NFSDProfile : INFO`
  - `NFSExport : ERROR`
  - `NFSHandle : ERROR`
- `maprcli nfsmgmt` - No response from the NFS server.
- `maprcli node services -nfs start|stop|restart|enable|disable` - Command has no effect.
- `maprcli alarm list` - An alarm is generated that indicates the wrong NFS version.

Customizing NFS

You can customize NFS behavior by modifying the `exports` and `nfsserver.conf` sections of the config map (`template-mfs-cm.yaml`), which is located at `bootstrap/customize/templates-dataplatform/template-mfs-cm.yaml`. Any configuration changes must be made before bootstrapping. For example, to change the share name, you must change the `/mapr (rw):` entry in the `exports:` section of the config map:

```
exports: |
    # Sample Exports file

    # for /mapr exports
    # <Path> <exports_control>

    #access_control -> order is specific to default
    # list the hosts before specifying a default for all
    #  a.b.c.d,1.2.3.4(ro) d.e.f.g(ro) (rw)
    #  enforces ro for a.b.c.d & 1.2.3.4 and everybody else is rw

    # special path to export clusters in mapr-clusters.conf. To disable
exporting,
    # comment it out. to restrict access use the exports_control
    #
    /mapr (rw)

    #to export only certain clusters, comment out the /mapr & uncomment.
    #/mapr/clustername (rw)
    .
    .
    .
nfsserver.conf: |
    # Configuration for nfsserver

    #
    # The system defaults are in the comments
    #
```

```
    # Default compression is true
    #Compression = true

    # chunksize is 64M
    #ChunkSize = 67108864

    # Number of threads for compression/decompression: default=2
    #CompThreads = 2

    #Mount point for the ramfs file for mmap
    #RamfsMntDir = /ramfs/mapr
    .
    .
    .
```

Please refer to Managing the HPE Ezmeral Data Fabric NFS Service for information about managing the NFS service (link opens an external website in a new browser tab/window).

Configuring Client Access

To configure clients to connect to NFS on the Kubernetes cluster network, locate the node on which the MFS pod is running, such as the `mfs-group1-0` pod shown below. The network location where NFS is running is the `NODE` value for the MFS pod:

```
[root@[redacted] ~]# kubectl get pod mfs-group1-0 -n dataplatform -o wide
NAME            READY   STATUS    RESTARTS   AGE     IP             NODE                                  NOMINATED NODE   READINESS GATES
mfs-group1-0    1/1     Running   0          5d18h   10.244.2.211   [redacted].net   <none>           <none>
[root@[redacted] ~]# _
```

In this example, the command to mount your share on the data-fabric cluster in Kubernetes is:

```
user-vbox2:~$ m2-sm2028-15-n4.mip.storage.hpecorp.net:/mapr
```

Pod Sizing Fields in CRs

Pod sizing fields in HPE Ezmeral Data Fabric on Kubernetes tenant and Data Fabric Custom Resources (CRs) have consistent names, usages, and meanings across all sections of the CR.

Pod Sizing Fields in CRs

HPE Ezmeral Data Fabric on Kubernetes tenant and Data Fabric Custom Resources (CRs) include some frequently-used settings in several sections. These settings have consistent meanings and usages across all sections and are therefore consolidated in the following section and referenced in other articles. You can see default values for all possible uses of the pod sizing fields in the sample CRs:

- `examples/Data Fabric/cr-full.yaml`
- `examples/tenants/cr-tenant-full-internal-hcp.yaml`

The pod sizing fields are:

- `requestcpu` - string - CPU amount to reserve for the pod, in the format `([1 - 9][0-9]+m)`. For example: `200m`.
- `limitcpu` - string - Maximum CPU for the pod, in the format `([1 - 9][0-9]+m)`. For example: `200m`.
- `requestmemory` - string - Amount of memory to reserve for the pod, in the format `([1 - 9]+Gi)`. For example: `4Gi`.
- `limitmemory` - string - Maximum memory amount for the pod, in the format `([1 - 9]+Gi)`. For example: `4Gi`.

last-updated: May 06, 2024

- `requestdisk` - string - Amount of ephemeral storage space to reserve for the pod. Default is `5Gi`.
- `limitdisk` - string - Maximum amount of ephemeral storage space to reserve for the pod. Default is `20Gi`.
- `loglevel` - string - Log level for the pod container. Value can be `ERROR`, `INFO` (default), or `DEBUG`.

See Managing Compute Resources (link opens an external website in a new browser tab/window).

Node Labels
HPE Ezmeral Data Fabric on Kubernetes `nodeservice` pods create labels on each node of the Kubernetes cluster.

Node Labels

The bootstrapping process deploys `nodeservice` pods to the `hpe-nodesvc` namespace as a daemonSet. The `nodeservice` pods create the following labels on each node of the Kubernetes cluster:

- `hpe.com/compute` - Whether (true; default) or not (false) to use the node for compute engines. `false` is a weak scheduler hint that is ignored if there is no other place to store pods.
- `hpe.com/exclusivecluster` - Storage cluster for which to use the node. Value can be `none` (default; uses the node for any storage cluster pod in the Kubernetes environment) or `<storage-cluster-name>` (only use the node for pods in the specified storage cluster).
- `hpe.com/dataplatform` - Whether (`true`; default) or not (`false`) to use the node for storage cluster pods. `false` is a weak scheduler hint that is ignored if there is no other place to store pods.
- `hpe.com/usenode` - Whether (`true`; default) or not (`false`) to use the node for installing the storage cluster or running compute engines. If `usenode` is `true` and both `compute` and `dataplatform` are `false`, then neither storage services nor compute spaces are installed on the node; however, the node might be used when there are capacity issues.

The `nodeservice` pods create the following annotations on each node of the Kubernetes cluster:

- `hpe.com/status`
- `hpe.com/createdpvs` - `True`/`False` - Internal use only. Flags the creation of persistent volumes.
- `hpe.com/decommission` - `Yes`/`No` - Unused.
- `hpe.com/validationversion` - Bootstrap utility version last run in the environment.
- `hpe.com/fulldisklist` - Complete list of available and used disks detected on the node, including the disk size.
- `hpe.com/hddlist` - List of available HDDs on the node.

  `hpe.com/maintenance` - `Yes`/`No` - Unused.
- `hpe.com/modifypvs` - `Yes`/`No` - Unused.
- `hpe.com/nodetopology` - Node topology, in the format `/<racknumber>/<nodename>`. Default is `rack1`.
- `hpe.com/nvmelist` - List of available NVMEs on the node.
- `hpe.com/physicalnodeid` - Example: `51ee6ec19ac3af40`.

- `hpe.com/rack` - Rack number label. Default is `rack1`.
- `hpe.com/sddlist` - List of available SDDs on the node.
- `hpe.com/validationerrors` - Whether (errors) or not (none) errors occurred during node validation.
- `hpe.com/validationstatus` - Status of the validation operation. Value can be either validated (completed successfully) or validating (operation currently running). Do not modify this property.
- `hpe.com/validationtimestamp` - Unused.

Command Reference: `edf update cluster`

The `edf update cluster` command updates components in HPE Ezmeral Data Fabric on Kubernetes clusters.

Syntax

```
edf update cluster
```

Description

The `edf update cluster` command updates HPE Ezmeral Data Fabric on Kubernetes cluster components that require an ordered shut down and restart process, such as CLDB, ZooKeeper, and MFS.

Usage

This command must be executed from the `admincli` pod of an HPE Ezmeral Data Fabric on Kubernetes cluster.

Example

```
kubectl exec -it admincli-0 -n <pod-namespace> /bin/bash
edf update cluster
```

Command Reference: `edf shutdown cluster`

The `edf shutdown cluster` command shuts down core components in Kubernetes HPE Ezmeral Data Fabric clusters and prevents them from resuming operations.

Syntax

```
edf shutdown cluster
```

Description

The `edf shutdown cluster` command shuts down HPE Ezmeral Data Fabric on Kubernetes cluster components that require an ordered shut down process, and prevents those components from completing the restart process. Examples of such components include CLDB pods and MFS pods.

When you use the `edf shutdown cluster` command, pods are shut down and are rebooted, but the pods are put into a wait state immediately after the reboot, which prevents the pods from becoming operational.

Usage

This command must be executed from the `admincli` pod of a Kubernetes Data Fabric cluster.

Use this command to stop operations on a Data Fabric cluster when you want to perform maintenance or upgrade procedures on HPE Ezmeral Data Fabric on Kubernetes.

If you want to troubleshoot one or more core component pods (such as CLDB or MFS), see the `edf startup pause` command. The `edf startup pause` prevents the component pods from completing the startup sequence after a pod restart, and pauses indefinitely until startup is manually resumed using the `edf startup resume` command.

To resume operations on the pod, see the `edf startup resume` command.

Example

```
kubectl exec -it admincli-0 -n <pod-namespace> /bin/bash
edf shutdown cluster
```

Related reference

Command Reference: edf startup {pause | resume} on page 719
The `edf startup pause` command flags core HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS, such that they will to enter into a nonfunctional state when they restart. The pods resume their startup sequence only after the `edf startup resume` command is executed.

Command Reference: `edf startup {pause | resume}`

The `edf startup pause` command flags core HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS, such that they will to enter into a nonfunctional state when they restart. The pods resume their startup sequence only after the `edf startup resume` command is executed.

Syntax

```
edf startup {pause | resume}
```

Description

The `edf startup {pause | resume}` command pauses or resumes the restart of core HPE Ezmeral Data Fabric on Kubernetes cluster components, such as CLDB and MFS pods.

Parameters

`pause`	Pauses the startup of certain HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS pods. When you `run edf startup pause`, there is no impact to a running pod. When one or more pods of this type are restarted (either manually or automatically by Kubernetes due to an issue), the startup sequence for the pod is paused. The pods wait for the `edf startup resume` command to be issued to complete the startup sequence and become functional again.
`resume`	Resumes the startup sequence of certain HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS pods. The command does not reboot pods.

Usage

This command must be executed from the `admincli` pod of the HPE Ezmeral Data Fabric on Kubernetes.

The `edf startup pause` command is intended for use when troubleshooting HPE Ezmeral Data Fabric on Kubernetes, in situations where you want the core component pods to enter into a nonfunctional state when they restart. This command does not stop pods that are currently running.

To perform an orderly shutdown of a Data Fabric cluster, see the `edf shutdown cluster` command.

Use the `edf startup resume` command to resume normal operations on a cluster after the `edf startup pause` or the `edf shutdown cluster` command has been executed.

You can check the status of the pods by executing the `edf report ready` command.

Example

```
kubectl exec -it admincli-0 -n <pod-namespace> /bin/bash
edf startup pause
...
edf startup resume
```

Related reference

Command Reference: edf shutdown cluster on page 718
The `edf shutdown cluster` command shuts down core components in Kubernetes HPE Ezmeral Data Fabric clusters and prevents them from resuming operations.

Command Reference: edf report ready on page 720
The `edf report ready` command reports the readiness of control plane Kubernetes pods in HPE Ezmeral Data Fabric clusters.

Command Reference: `edf report ready`

The `edf report ready` command reports the readiness of control plane Kubernetes pods in HPE Ezmeral Data Fabric clusters.

Syntax

```
edf report ready
```

Description

The `edf report ready` command checks whether or not certain control-plane HPE Ezmeral Data Fabric on Kubernetes cluster components, such as CLDB and MFS pods, are ready for an upgrade or maintenance procedure.

When you use the `edf shutdown cluster` command, pods are shut down and are rebooted, but the pods are put into a wait state immediately after the reboot, which prevents the pods from becoming operational. You use the `edf report ready` to determine if pods have rebooted and are ready for you to continue with your upgrade or maintenance procedure.

Usage

This command must be executed from the `admincli` pod of a Kubernetes Data Fabric cluster.

This command can take a couple of minutes to complete. You might notice a delay between the display of the second and the third lines of the output. If the pods are ready for upgrade, you can proceed with upgrade or other maintenance tasks.

Example

The following example shows the output when the pods are not ready:

```
edf report ready
2021/06/14 23:22:34 [edf reports]: [INFO] Checking if pods are stabilized for
upgrade. This may take a minute or two.
2021/06/14 23:22:35 [edf reports]: [INFO] Valid MapR user ticket found,
skipping ticket generation
2021/06/14 23:24:31 [edf reports]: [ERROR] Pods are not ready for upgrade
2021/06/14 23:24:31 [edf reports]: [ERROR] Check out /tmp/
report-20210614232234 for details
```

The following example shows the output when the pods are ready:

```
edf report ready
2021/06/14 23:28:01 [edf reports]: [INFO] Checking if pods are stabilized for
upgrade. This may take a minute or two.
2021/06/14 23:28:02 [edf reports]: [INFO] Valid MapR user ticket found,
skipping ticket generation
2021/06/14 23:29:52 [edf reports]: [INFO] Pods are ready
```

Related reference

Command Reference: edf startup {pause | resume} on page 719
The `edf startup pause` command flags core HPE Ezmeral Data Fabric on Kubernetes components, such as CLDB and MFS, such that they will to enter into a nonfunctional state when they restart. The pods resume their startup sequence only after the `edf startup resume` command is executed.

GPU and MIG Support

This topic provides information about support for NVIDIA GPU and MIG devices on HPE Ezmeral Runtime Enterprise.

GPU Support

HPE Ezmeral Runtime Enterprise supports making NVIDIA Data Center CUDA GPU devices available to containers or virtual nodes for use in CUDA applications.

- For information about the GPU devices supported by HPE Ezmeral Runtime Enterprise, see Support Matrixes on page 54.
- NVIDIA driver version 470.57.02 or later is required on hosts that have GPUs, regardless of whether those GPUs support Multi-Instance GPU (MIG).
- For information about the available device versions, see CUDA CPUs.

HPE Ezmeral Runtime Enterprise supports GPUs on Kubernetes nodes. The underlying hosts must be running an operating system and version that is supported on the corresponding version of HPE Ezmeral Runtime Enterprise. See OS Support on page 85.

Support for MIG-enabled devices is subject to additional requirements and restrictions. See MIG Support on page 722.

NOTE:

HPE Ezmeral Runtime Enterprise 5.3.5 and later releases deploy updated versions of the NVIDIA runtime and other required NVIDIA packages, and has changed the node label used to identify hosts that have GPU devices. When upgrading from a release of HPE Ezmeral Runtime Enterprise prior to 5.3.5, you must remove hosts that have GPUs (regardless of whether they are MIG-enabled) from HPE Ezmeral Runtime Enterprise before the upgrade, and then add those hosts after the upgrade to HPE Ezmeral Runtime Enterprise is complete.

For more information about using GPU resources in Kubernetes pods in HPE Ezmeral Runtime Enterprise, see Using GPUs in Kubernetes Pods on page 727.

MIG Support

The Multi-Instance GPU (MIG) feature from NVIDIA virtualizes the GPU such that applications can use a fraction of a GPU to optimize resource usage and to provide workload isolation.

HPE Ezmeral Runtime Enterprise supports MIG as follows:

- NVIDIA A100 MIG instances are supported on Kubernetes Worker hosts running RHEL 7.x , CentOS 7.x, and SLES as listed in OS Support on page 85.
- MIG support requires NVIDIA driver version 470.57.02 or later.
- On hosts that have multiple A100 GPU devices, each GPU device can have a different MIG configuration.

  For example, on a system that has four A100 GPUs, the MIG configurations might be as follows:

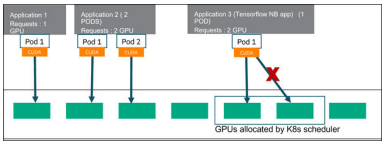
  - GPU 0: MIG disabled
  - GPU 1: 7 MIG 1g.5gb devices
  - GPU 2: 3 MIG 2g.10gb devices
  - GPU 3: 1 MIG 4g.20gb, 1 MIG 2g.10gb, 1 MIG 1g.5gb
- The NVIDIA GPU operator is not supported.
- No GPU metrics on A100 MIG instances are available. NVIDIA recommends using DCGM-Exporter. (link opens an external website in a new browser tab or window).
- Information about hosts listed on the **Host(s) Info** tab of the Kubernetes **Cluster Details** screen includes the number of GPU devices. The **More Info** link displays information about the MIG instances.
- The NVIDIA device plugin version 0.9.0, which is deployed by the `nvidia-plugin` Kubernetes add-on, does not support multiple compute instances (CI) for the same GPU instance (GI). Therefore, do **not** configure MIG with profiles that start with *X*c, such as 1c.3g.20gb or 2c.3g.20gb.
- As stated in the NVIDIA Multi-Instance GPU User Guide, "MIG supports running CUDA applications by specifying the CUDA device on which the application should be run. With CUDA 11, only enumeration of a single MIG instance is supported." This restriction means that applications that use CUDA can use only the first MIG device applied to a pod.

  For example, consider the case in which an A100 GPU is configured with 7 MIG devices, with Pod1 assigned to MIG device 0 and Pod2 assigned to MIG device 1. Then, a Tensorflow notebook application pod is created, and that pod specifies two GPUs. The following occurs:

  - The new pod is assigned MIG devices 2 and 3.
  - When invoked from inside the pod, the `nvidia-smi -L` command returns two devices, indexed as 0 and 1.

    For example:

    ```
    kubectl exec -n tenant1 tf-gpu-nb-controller-mvz7p-0 -- nvidia-smi -L
    GPU 0: A100-SXM4-40GB (UUID: GPU-5d5ba0d6-d33d-2b2c-524d-9e3d8d2b8a77)
      MIG 1g.5gb      Device  0: (UUID:
    MIG-c6d4f1ef-42e4-5de3-91c7-45d71c87eb3f)
      MIG 1g.5gb      Device  1: (UUID:
    MIG-cba663e8-9bed-5b25-b243-5985ef7c9beb)
    ```



- However, the Tensorflow `tf.config.list_physical_devices('GPU')` request returns only one device:

```
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

- Quotas on (tenant) namespaces for GPUs are applied by the `nvidia.com/gpu` specifier, which applies to physical GPUs and MIG instances in `single` strategy only. For example, specifying a quota of three devices of 1g.5gb is not supported.

For more information about the MIG feature, including application considerations, see the following from NVIDIA (links open an external website in a new browser window or tab):

- Multi-Instance GPUs
- MIG Support on Kubernetes
- NVIDIA Multi-Instance GPU User Guide

Host GPU Driver Compatibility

The host OS NVIDIA driver must be compatible with the CUDA library version required by the application.

For example:

- Tensorflow has information about tested build configurations for GPUs, which includes version compatibility information for Python, CUDA, and so forth.
- The KubeDirector Notebook application that is included with HPE Ezmeral Runtime Enterprise 5.4.x releases is installed with CUDA version 11.4.3. The Python Training and Python Inference applications also use CUDA 11.x.

The driver and CUDA package bundles from NVIDIA may not support every GPU listed here CUDA CPUs (link opens an external website in a new browser tab or window). You might need to download and install the driver and compatible CUDA toolkit for your specific GPU model separately.

For information about requirements for MIG support, see MIG Support on page 722.

CUDA Toolkit

For RHEL and SLES hosts, you can download and install an OS-specific CUDA toolkit (link opens an external website in a new browser tab or window). The toolkit can be useful for building applications. The toolkit might not be needed on the host itself.

The NVIDIA driver version determines the supported CUDA toolkit versions, as described in CUDA Toolkit and Compatible Driver Versions (link opens an external website in a new browser tab or window).

You can choose the NVIDIA driver version when configuring on-premises resources. When you add Amazon EC2 hosts with GPUs (such as in a hybrid deployment), then the NVIDIA driver is installed as part of the AMI for the EC2 instance that supports each node.

Deploying GPUs in HPE Ezmeral Runtime Enterprise

For information about deploying GPU and MIG in HPE Ezmeral Runtime Enterprise, see the following:

- GPU Driver Installation on page 838.

- Deploying MIG Support on page 840

GPU and MIG Resources in Kubernetes Applications

For information about using GPU resources in Kubernetes applications and pods, see Using GPUs in Kubernetes Pods on page 727.

Viewing GPU and MIG Devices Using the GUI

View GPU and MIG device information on the **Host(s) Info** tab of the Kubernetes cluster details screen in HPE Ezmeral Runtime Enterprise.

Prerequisites

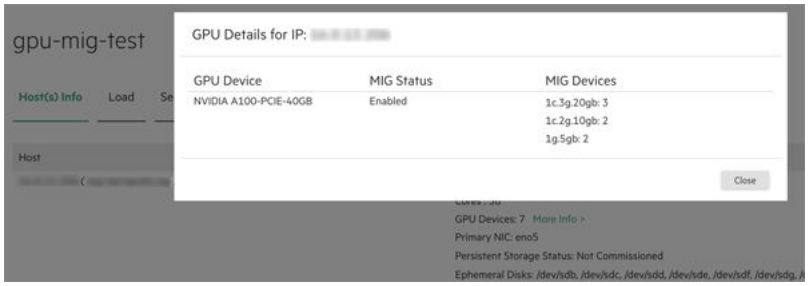
Required access rights: Platform Administrator

Procedure

- To use the GUI:

  On the **Host(s) Info** tab of the **Kubernetes Cluster Details** screen, locate the host. The GPU information is in the **Details** column entry.

  If the GPU supports MIG, when you click the **More Info** link, **GPU Details** dialog shows information about the MIG configuration. For example:



Viewing GPU and MIG Devices Using kubectl Commands

View GPU and MIG device information using `kubectl` commands.

Prerequisites

Required access rights: Platform Administrator

Procedure

- To verify that the Kubernetes pod recognizes the GPU resources, enter the following command:

```
kubectl get nodes --selector=nvidia.com/gpu.count -Lnvidia.com/
gpu.count -Lnvidia.com/gpu.product -Lnvidia.com/mig.strategy
```

  The output of the command lists the nodes that have GPU devices. For each node, it lists the GPU product name and, for MIG-enabled GPUs, the configured MIG strategy.

  For example:

```


NAME                      STATUS   ROLES    AGE   VERSION    GPU.COUNT
GPU.PRODUCT   MIG.STRATEGY
dev04.mycorp.net          Ready    worker   22d   v1.20.11   1
Tesla-P4      mixed
```

- To identify the GPU and MIG resources—if any—in a given node, use the `kubectl describe node <node-name>` command.

  The output of the `kubectl describe node <gpu-node>` command varies as follows:

MIG-enabled GPU, `mixed` strategy

If the host has GPUs that are MIG-enabled using a `mixed` strategy, the system returns something like the following:

```
...
Capacity:
cpu:                48
ephemeral-storage:  1049136384Ki
hugepages-1Gi:      0
hugepages-2Mi:      0
memory:             131523060Ki
nvidia.com/mig-1g.5gb:   1
nvidia.com/mig-2g.10gb:  1
nvidia.com/mig-3g.20gb:  1
pods:               110
```

MIG-enabled GPU, `single` strategy

If the host has GPUs that are MIG-enabled using a `single` strategy, the output is similar to the hosts that have GPUs that are not MIG-enabled, except that the number of GPUs is greater than one:

```
...
Capacity:
nvidia.com/gpu:          7
...
Allocatable:
nvidia.com/gpu:          7
...
```

GPU is not MIG-enabled

If the host has GPUs that are not MIG-enabled, the system returns something like the following:

```
...
Capacity:
cpu:                48
ephemeral-storage:  1049136384Ki
hugepages-1Gi:      0
```

	`hugepages-2Mi:  0`
	`memory:  131523060Ki`
	`nvidia.com/gpu:  1`
	`pods:  110`
	`...`

Host does not have a GPU	If the host does not have a GPU, then the `nvidia.com/gpu` field does not appear.

Viewing GPU and MIG Devices Using nvidia-smi Commands

View GPU and MIG device information using `nvidia-smi` commands.

Prerequisites

Required access rights: Platform Administrator

About this task

The NVIDIA System Management Interface (`nvidia-smi`) is a command line utility that enables you to view and modify GPU device state. It is also used to configure MIG devices. For more information about `nvidia-smi`, see the following NVIDIA documentation (links open an external website in a new browser tab or window):

- MIG Support on Kubernetes
- NVIDIA Multi-Instance GPU User Guide

Procedure

- To view the MIG devices on a node, you can use the `nvidia-smi -L` command.

  For example, the following output shows a single physical GPU device with three MIG instances, each of which has a different MIG configuration (also called a MIG profile):

  ```
  sudo nvidia-smi -L

  GPU 0: NVIDIA A100-PCIE-40GB (UUID:
  GPU-b5e82144-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
    MIG 3g.20gb     Device  0: (UUID:
  MIG-cc1de538-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
    MIG 2g.10gb     Device  1: (UUID:
  MIG-202913a0-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
    MIG 1g.5gb      Device  2: (UUID:
  MIG-01efa7b8-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
  ```

  See also Troubleshooting MIG on HPE Ezmeral Runtime Enterprise on page 731.

Changing the MIG Configuration

To change the MIG configuration on a host, remove the host from HPE Ezmeral Runtime Enterprise, make the configuration changes on the host, and then add the host to HPE Ezmeral Runtime Enterprise and to the Kubernetes cluster.

Prerequisites

Required access rights: Platform Administrator

Procedure

1. Remove the host from the Kubernetes cluster.

   See Expanding or Shrinking a Kubernetes Cluster on page 483.

2. Delete the host from HPE Ezmeral Runtime Enterprise.

   See Decommissioning/Deleting a Kubernetes Host on page 555.

3. Use the `nvidia-smi` tool to change the MIG configuration.

   See Deploying MIG Support on page 840 and, if needed, the NVIDIA documentation.

4. Add the host to HPE Ezmeral Runtime Enterprise as a Kubernetes Worker.

   See Kubernetes Worker Installation Overview on page 528.

Using GPUs in Kubernetes Pods

This topic describes how to identify and request GPU and MIG resources, and how to use node labels and the Kubernetes nodeAffinity feature to constrain the pods that are eligible for scheduling.

Identifying GPU Resources

You can view GPU and MIG resources in HPE Ezmeral Runtime Enterprise using the GUI or by using `kubectl` or `nvidia-smi` commands. See GPU and MIG Support on page 721.

Requesting GPU Resources

A Kubernetes application can request GPU resources in its YAML file, and these resources will be scheduled accordingly.

HPE Ezmeral Runtime Enterprise taints GPU hosts to try to eliminate having non-GPU pods scheduled on hosts with GPUs. However, GPU-equipped hosts will be used for non-GPU pods if no other resources are available.

There are two key parts to specifying a GPU resource in the YAML file:

- Specifying the correct key name in the `resources:` specification. For GPUs in HPE Ezmeral Runtime Enterprise, that key name is: `nvidia.com/gpu`

  For example:

  ```
  resources:
    limits:
      nvidia.com/gpu: 2
  ```

- Setting the NVIDIA_DRIVER_CAPABILITIES environment variable to the value: `compute,utility`

  For example:

  ```
  env:
   -
    name: "NVIDIA_DRIVER_CAPABILITIES"
    value: "compute,utility"
  ```

  If this is a KubeDirector application with GPU support, such as **Jupyter Notebook with ML toolkits**, when you select a nonzero GPU count in the UI, HPE Ezmeral Runtime Enterprise adds the NVIDIA_DRIVER_CAPABILITIES environment variable to the KD app YAML automatically. Otherwise, you can add the environment variable manually.

You include these items in any native Kubernetes resource that includes a Container object (link opens an external website in a new browser tab or window), including pods and higher-level pod-creating resources such as Deployment, StatefulSet, and DaemonSet.

In a KubeDirectorCluster specification, you include these items in the RoleSpec (link opens an external website in a new browser tab or window) of the role that accesses GPUs.

To specify MIG resources, see Requesting MIG Resources on page 729.

Using nodeAffinity

You might want to restrict the application to run on a specific GPU type because of availability or cost considerations in your business environment. For example, using an A100 GPU might have a different billing rate than other types of GPUs.

You can use a combination of node labels and the Kubenetes nodeAffinity feature (link opens an external website in a new browser tab or window) to constrain which nodes pods are eligible to be scheduled on.

Using nodeAffinity to Select By GPU Type

You might want to restrict the application to run on a specific GPU type because of availability or cost considerations in your business environment. For example, using an A100 GPU might have a different billing rate than other types of GPUs.

You can use a combination of node labels and the Kubenetes nodeAffinity feature (link opens an external website in a new browser tab or window) to constrain which nodes pods are eligible to be scheduled on.

The `nodeAffinity` feature includes an expressive matching language, and the ability to specify a preference instead of a hard requirement. You can also use the match expressions and operators to express an anti-affinity.

The procedure, in concept, is the following:

1. If needed, the Kubernetes Cluster Administrator or Platform Administrator can label the nodes to which you want to apply preferences or restrictions.

   If you want to use an existing default node label, you do not need to create and apply label key-value pairs to nodes, but you do need the Kubernetes Cluster Administrator or Platform Administrator to supply you with the list of node labels.

   A Kubernetes Cluster Administrator or Platform Administrator can get a valid list of keys and values of node labels by querying with `kubectl` commands. For an example, see Listing the nvidia.com Node Labels on page 731.

   For example, in HPE Ezmeral Runtime Enterprise, nodes that have GPUs have a set of default node labels, one of which has the key: `nvidia.com/gpu.product`. One of the valid values of that key is `Tesla-P4`.

   However, you might want to enable users that create applications to specify the appropriate category of GPU without knowing the exact model identifier of the GPU. For example, you might want to label one or more nodes as having "general-purpose" or "higher-performance" GPUs, using node labels such as `gputype=general-purpose`. In your deployment, you might apply the same label to hosts that have one of several GPU models.

2. Specify the `nodeAffinity` in the `affinity` field.

   Any native Kubernetes resource that includes a PodSpec object (link opens an external website in a new browser tab or window) can put an affinity field into that object. This includes pods and higher-level pod-creating resources such as Deployment, StatefulSet, and DaemonSet.

   In a KubeDirectorCluster specification, you include the affinity field in the RoleSpec (link opens an external website in a new browser tab or window).

In the following example, `nodeAffinity` expresses a preference to schedule RESTserver pods in nodes with a Tesla-P4 GPU.

Specifying `preferredDuringSchedulingIgnoredDuringExecution` instead of `requiredDuringSchedulingIgnoredDuringExecution` indicates that, if a preferred node is not available at the time the pod is scheduled, the pod may be scheduled in a node that is not eligible according to the `matchExpressions`.

```
...
    affinity:
      nodeAffinity:
        preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
            matchExpressions:
            - key: nvidia.com/gpu.product
              operator: In
              values:
              - Tesla-P4
...
```

Requesting MIG Resources

As with requesting GPU resources, setting the NVIDIA_DRIVER_CAPABILITIES environment variable to `compute,utility` is required. However, the way you specify the MIG instance differs in both resource requests and in the standard `nvidia.com/gpu.product` node label values.

For applications that support specifying resources for MIG-enabled GPUs, the way you specify the MIG instance differs depending on the Kubernetes MIG strategy chosen by the Platform Administrator.

`single` strategy

If the `single` strategy is used, when you request resources, you specify the number of MIG instances in the same way as for physical GPUs devices.

For example:

```
...
    resources:
      limits:
        nvidia.com/gpu: 1
...
```

If you have different nodes with different MIG configurations, you can use the `nodeAffinity` field to specify the node that has MIG configuration you want to use.

The following example uses the standard `nvidia.com/gpu.product` key to require a particular MIG configuration. If a node with that configuration is not available, the pod will not be scheduled.

```
...
      resources:
        limits:
          nvidia.com/gpu: 1
      env:
       -
        name:
NVIDIA_DRIVER_CAPABILITIES
        value: 'compute,utility'
```

```
...
    affinity:
      nodeAffinity:

requiredDuringSchedulingIgnoredDuringE
xecution:
        - weight: 1
          preference:
            matchExpressions:
            - key: nvidia.com/
gpu.product
              operator: In
              values:
              -
A100-SXM4-40GB-MIG-1g.5gb
...
```

`mixed` strategy

If the `mixed` strategy is used, when you request resources, you specify and enumerate MIG devices by their fully qualified name in the form:

```
nvidia.com/
mig-<slice_count>g.<memory_size>gb
```

If the `mixed` strategy is used, the value of standard `nvidia.com/gpu.product` node label is the physical GPU.

```
...
      resources:
        limits:
          nvidia.com/mig-3g.20gb: 1
      env:
       -
        name:
NVIDIA_DRIVER_CAPABILITIES
        value: 'compute,utility'
...
    affinity:
      nodeAffinity:

requiredDuringSchedulingIgnoredDuringE
xecution:
        - weight: 1
          preference:
            matchExpressions:
            - key: nvidia.com/
gpu.product
              operator: In
              values:
              - A100-SXM4-40GB
...
```

NOTE:

As stated in "Device Enumeration" in the NVIDIA Multi-Instance GPU User Guide (link opens an external website in a new browser window or tab): "MIG supports running CUDA applications by specifying the CUDA device on which the application should be run. With CUDA 11, only enumeration of a single MIG instance is supported."

Therefore, an application can access only one GPU MIG instance (the first instance applied to the pod), even if the pod spec specifies a limit larger than one.

Listing the `nvidia.com` Node Labels

The following command queries all nodes for the node labels that have a label key that starts with `nvidia.com`. You must have Kubernetes Cluster Administrator or Platform Administrator rights to execute this command.

```
kubectl get nodes -o json | jq '.items[].metadata.labels |
with_entries(select(.key | startswith("nvidia.com")))'
```

The command is useful to obtain the valid `nodeAffinity` key-value pairs.

Troubleshooting MIG on HPE Ezmeral Runtime Enterprise

Troubleshooting tips for verifying MIG installation and configuration in Kubernetes deployments of HPE Ezmeral Runtime Enterprise.

These troubleshooting tips apply to the deployment of MIG devices in Kubernetes deployments of HPE Ezmeral Runtime Enterprise.

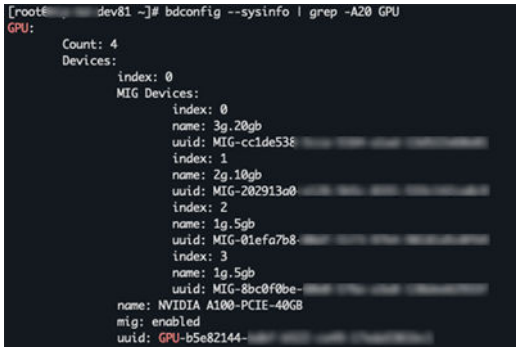
These tips are meant to supplement troubleshooting information available from NVIDIA, such as the following (link opens an external website in a new browser tab or window):

- MIG Support on Kubernetes
- NVIDIA Multi-Instance GPU User Guide

Verifying Matching bdconfig and nvidia-smi Output

On the GPU host, verify that the information about GPU and MIG returned by `bdconfig` matches the GPU and MIG information returned by `nvidia-smi`.

Example `bdconfig --sysinfo` command and output:



Example `nvidia-smi -L` command and output:

```
sudo nvidia-smi -L

GPU 0: NVIDIA A100-PCIE-40GB (UUID: GPU-b5e82144-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
  MIG 3g.20gb     Device  0: (UUID: MIG-cc1de538-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
```

```
  MIG 2g.10gb     Device  1: (UUID: MIG-202913a0-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
  MIG 1g.5gb      Device  2: (UUID: MIG-01efa7b8-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
  MIG 1g.5gb      Device  3: (UUID: MIG-8bc0f0be-xxxx-xxxx-xxxx-xxxxxxxxxxxx)
```

Verifying GPU Node Labels

On worker nodes that have MIG-enabled GPUs, verify the node labels:

- `"hpe.com/mig,strategy": "single",` or `"hpe.com/mig,strategy": "mixed",`
- The `nivida.com/gpu.product` label specifies a MIG-enabled GPU and MIG configuration. For example:

  ```
  "nivida.com/gpu.product": "NVIDIA-A100-PCIE-40GB-MIG-1g.5gb"
  ```

Verifying That the Required Pods Are Running

Verify that the `nvidia-device-plugin`, `gpu-feature-discovery`, `nfd-worker`, and `nfd-master` pods are running on all nodes that have MIG-enabled GPUs.

```
kubectl get nodes
kubectl -n kube-system get pods -o wide | grep nvidia
kubectl -n kube-system get pods -o wide | grep nfd
```

Verifying GPU Resources on Worker Nodes

Use the kubectl `describe node` command to verify that the MIG resources are allocatable on worker nodes.

The following example shows output when the `mixed` strategy is configured:

```
Addresses:
  InternalIP:
  Hostname:                                                    corp.net
Capacity:
  cpu:                          112
  ephemeral-storage:            12004360Mi
  hugepages-1Gi:                0
  hugepages-2Mi:                0
  memory:                       263633096Ki
  nvidia.com/mig-1g.5gb:        2
  nvidia.com/mig-2g.10gb:       1
  nvidia.com/mig-3g.20gb:       1
  pods:                         110
Allocatable:
  cpu:                          106400m
  ephemeral-storage:            11328735393468
  hugepages-1Gi:                0
  hugepages-2Mi:                0
  memory:                       237167816Ki
  nvidia.com/mig-1g.5gb:        2
  nvidia.com/mig-2g.10gb:       1
  nvidia.com/mig-3g.20gb:       1
  pods:                         110
```

Verifying the Number of NVIDIA DaemonSets

The NVIDIA plugin pods are automatically configured on hosts with GPU resources. They are not present on non-GPU hosts. The NVIDIA plugin pod enables GPU reservation in application YAML files and is deployed as four DaemonSets:

- `nvidia-device-plugin-mixed`
- `nvidia-device-plugin-single`
- `gpu-feature-discovery-mixed`
- `gpu-feature-discovery-single`

To list the NVIDIA device plugin DaemonSets, execute the following commands on the Kubernetes master:

```
kubectl get -n kube-system ds -l app.kubernetes.io/name=nvidia-device-plugin
```

```
kubectl get -n kube-system ds -l app.kubernetes.io/
name=gpu-feature-discovery
```

The following example shows the output when there is one GPU node configured to use the `mixed` strategy:

```
kubectl get -n kube-system ds -l app.kubernetes.io/name=nvidia-device-plugin

NAME                          DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE
SELECTOR                                      AGE
nvidia-device-plugin-mixed   1       1       1     1
1                                                        11d
nvidia-device-plugin-single  0       0       0     0
0                                                        11d

kubectl get -n kube-system ds -l app.kubernetes.io/name=gpu-feature-discovery

NAME                          DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE
SELECTOR                                      AGE
gpu-feature-discovery-mixed  1       1       1     1          1
feature.node.kubernetes.io/pci-10de.present=true 11d
gpu-feature-discovery-single 0       0       0     0          0
feature.node.kubernetes.io/pci-10de.present=true 11d
```

Verifying MIG Configuration After Host Reboot

You can verify that the MIG configuration was restored after a reboot by executing the following command:

```
sudo service bds-nvidia-mig-config status
```

The output of the command is similar to the following:

```
Redirecting to /bin/systemctl status bds-nvidia-mig-config.service
bds-nvidia-mig-config.service - Oneshot service to re-create NVIDIA MIG
devices
   Loaded: loaded (/usr/lib/systemd/system/bds-nvidia-mig-config.service;
enabled; vendor preset: disabled)
   Active: active (exited) since Sat 2022-12-10 18:34:11 PST; 1 weeks 3
days ago
 Main PID: 2164 (code=exited, status=0/SUCCESS)
    Tasks: 0
   Memory: 0B
   CGroup: /system.slice/bds-nvidia-mig-config.service
Dec 10 18:33:01 mynode-88.mycorp.net systemd[1]: Starting Oneshot service
to re-create NVIDIA MI...
Dec 10 18:34:11 mynode-88.mycorp.net python[2164]: MIG command 'nvidia-smi
mig -i 0 -lgi' failed...
Dec 10 18:34:11 mynode-88.mycorp.net python[2164]: Failed getting current
MIG configuration on G...
Dec 10 18:34:11 mynode-88.mycorp.net python[2164]: Got stored GPU MIG
configuration. Trying to r...
Dec 10 18:34:11 mynode-88.mycorp.net python[2164]: Restoring MIG
configuration on GPU 0: '[{u'start...
Dec 10 18:34:11 mynode-88.mycorp.net python[2164]: MIG configuration on GPU
0 restored successfully.
Dec 10 18:34:11 mynode-88.mycorp.net systemd[1]: Started Oneshot service to
re-create NVIDIA MIG...
```

bds-nvidia-mig-config Service

`bds-nvidia-mig-config` service is a systemd service that preserves the MIG device configurations across system reboots.

You can check its status and examine its logs by executing commands such as the following:

```
systemd status bds-nvidia-mig-config
```

```
journalctl -u bds-nvidia-mig-config
```

More information

Using GPUs in Kubernetes Pods on page 727
This topic describes how to identify and request GPU and MIG resources, and how to use node labels and the Kubernetes nodeAffinity feature to constrain the pods that are eligible for scheduling.

Licensing

Licenses

The Hewlett Packard Enterprise licensing mechanism enforces the licensing terms.

You may have multiple coexisting licenses. A newer license does not delete an older license, but is cumulative. The **License** tab of the **System Settings** screen displays summary of each installed license on the total number of licensed cores, latest expiration date, and so forth.

Licenses Apply to CPU Cores

Licenses apply to a certain number of CPU cores for a certain period of time. Each host in the deployment contains one or more CPU cores, and the cores in each host apply to the licensed maximum. You cannot assign only some of the CPU cores in a host to the deployment.

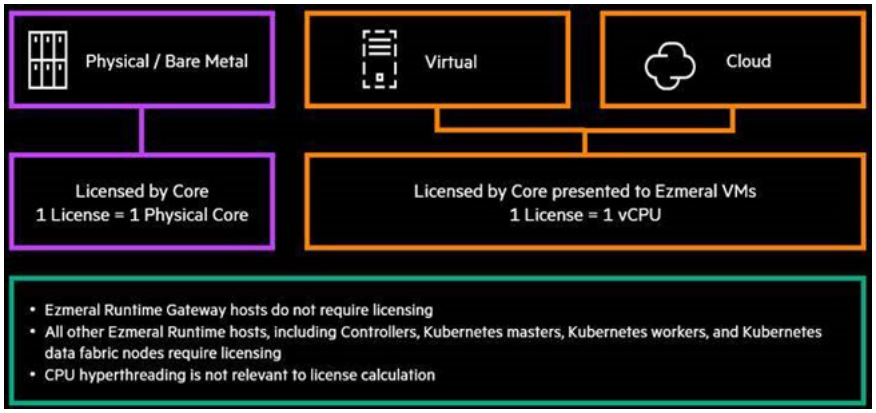
For example, if your license is valid for 100 CPU cores and your infrastructure uses hosts with 8 CPU cores per host, then you can add a total of 12 hosts with 96 cores to the deployment; you cannot add only 4 of the 8 cores in a 13th host. In this example, to add the 13th host, you would need to expand your license to allow at least 104 CPU cores.

Licenses are cumulative. For example, if you have two licenses of the same type where one license allows 50 CPU cores and the other allows 30 CPU cores, then you will be able to use up to 80 CPU cores under that type of license.

Hewlett Packard Enterprise licenses the total number of host CPU cores that can be used. If you attempt to add one or more hosts whose CPU cores would exceed the licensed maximum, then the affected hosts will display the status **Unlicensed**, and you will not be able to continue installing the hosts. To resolve this issue, either add a new license that allows the increased number of CPU cores, or delete the hosts you are trying to add (See Decommissioning/Deleting a Kubernetes Host on page 555).

Definition of CPU Cores

HPE Ezmeral Runtime Enterprise is licensed by the number of unique cores available to the kernel in the OS on which the HPE Ezmeral Runtime Enterprise software is directly installed, regardless of the number of threads in each core.



For licensing purposes, cores and core capacity is formally defined in HEWLETT PACKARD ENTERPRISE SOFTWARE END USER SUBSCRIPTION AGREEMENT on page 87.

License Inclusions

To determine what products, features, and functions are included in a license, see What's Included on page 87.

License Expiration

A license file specifies the expiration date; however, you can add, modify, or remove a license at any time.

An alert is added to the Alerts list in the web interface when a license is approaching its expiration date, and after a license has expired.

CAUTION:

If the HPE Ezmeral Instant-On and all other evaluation licenses expire before before a purchased license has been applied, then the deployment will go into Lockdown mode (see Lockdown Mode on page 916). The Platform Administrator will not be able to exit lockdown mode until a purchased license is applied.

When a purchased license expires, the deployment does not go into Lockdown mode. An alert is added to the Alerts list. Contact Hewlett Packard Enterprise to obtain a new license.

Information in a License

A license contains the following information:

- **Controller ID:** Unique ID of the Controller host. You will need to provide this number to Hewlett Packard Enterprise to request a new or updated/extended license.
- **Name:** Name of the license.
- **Number of cores:** Number of CPU cores that can exist at any one time in this deployment. The total CPU cores reported by the combined physical hosts are counted against the total number of licensed cores.
- **Validity:** Last day on which the current license will be valid, in MM-DD-YYYY format. A warning bar appears in web interface screens when the license is approaching its expiration data.
- **Version:** Version of HPE Ezmeral Runtime Enterprise.
- **License File:** Name of the current license file.

Related tasks

Upgrading from HPE Ezmeral Runtime Enterprise Essentials on page 911
Upgrade from HPE Ezmeral Runtime Enterprise Essentials to the full-featured HPE Ezmeral Runtime Enterprise or to HPE Ezmeral ML Ops by uploading a license. No additional steps are required.

Related reference

License Tab on page 798
The License tab enables the Platform Administrator to manage HPE Ezmeral Runtime Enterprise licenses.

More information

Lockdown Mode on page 916

HPE Ezmeral Instant-On License

The HPE Ezmeral Instant-On license is an evaluation license that is included when you install HPE Ezmeral Runtime Enterprise.

The HPE Ezmeral Instant-On license is a type of evaluation or demo license that is included when you install HPE Ezmeral Runtime Enterprise. With the HPE Ezmeral Instant-On license, you can try out various HPE Ezmeral Runtime Enterprise and HPE Ezmeral ML Ops features including Spark functions from within an ML Ops tenant.

CAUTION:

If the HPE Ezmeral Instant-On and all other evaluation licenses expire before before a purchased license has been applied, then the deployment will go into Lockdown mode (see Lockdown Mode on page 916). The Platform Administrator will not be able to exit lockdown mode until a purchased license is applied.

The HPE Ezmeral Instant-On license expires but it cannot be deleted. If you delete it, it is recreated automatically.

Related concepts

Licensing on page 734

Adding Licenses

Prerequisites

- You have purchased a license and obtained the license file.

  To purchase a license, contact Hewlett Packard Enterprise.

- If this is an HPE Ezmeral Runtime Enterprise Essentials deployment, ensure that you have HPE Ezmeral Runtime Enterprise Essentials license. Except for the HPE Ezmeral Instant-On license, after license that includes the full-featured HPE Ezmeral Runtime Enterprise license is applied, the deployment cannot be changed to HPE Ezmeral Runtime Enterprise Essentials.

- **Required access rights:** Platform Administrator

About this task

Tyipically, you add a license to do one of the following:

- Replace a trial license, such as HPE Ezmeral Instant-On with a purchased license
- Renew an existing or expired license
- License additional features, such Spark or ML Ops features
- Increase the number of licensed CPU cores

Licenses are cumulative.

For example, if you have two licenses of the same type where one license allows 50 CPU cores and the other allows 30 CPU cores, then you will be able to use up to 80 CPU cores under that type of license.

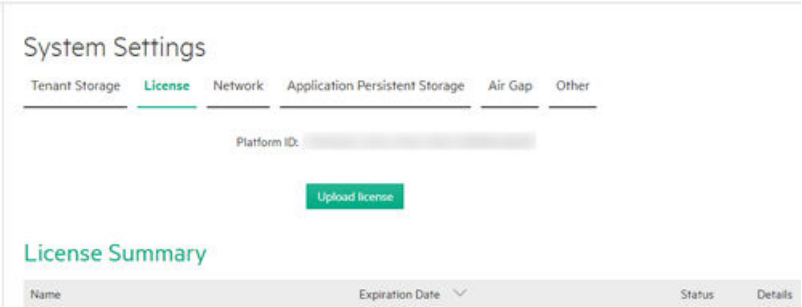
CAUTION:

If the HPE Ezmeral Instant-On and all other evaluation licenses expire before before a purchased license has been applied, then the deployment will go into lockdown mode (see Lockdown Mode on page 916). The Platform Administrator will not be able to exit lockdown mode until a purchased license is applied.

Procedure

1. Open the **System Settings** screen, and then select the **License** tab.

   For more information about these screens, see The System Settings Screen and License Tab).



2. Click the **Upload license** button to navigate to and select a new license file.

   The specified license file uploads, and a confirmation message appears.

Related concepts

Licensing on page 734

Related reference

License Tab on page 798

The License tab enables the Platform Administrator to manage HPE Ezmeral Runtime Enterprise licenses.

Global Settings

Platform Administrator can manage the global settings that affect the entire deployment.

Please also see:

- Platform Administrator Overview on page 570 for a list of articles that describe managing the Big Data, AI, and/or ML settings for your deployment.
- Kubernetes Overview for a list of articles that describe managing Kubernetes.

Enabling SSL Connections

This procedure describes how to enable SSL connections in HPE Ezmeral Runtime Enterprise deployments for which SSL was not enabled during the initial deployment process.

Prerequisites

You are logged into the active Controller host as the user account that was used to install HPE Ezmeral Runtime Enterprise.

About this task

If you followed the instructions in Adding an SSL Certificate on page 843 during the HPE Ezmeral Runtime Enterprise deployment process, you do not need to complete this task.

Use this procedure to enable SSL connections on an existing HPE Ezmeral Runtime Enterprise deployment.

Procedure

1. Either generate or obtain an SSL certificate that includes the correct set of hostnames in the Common Name (CN) or Subject Alternative Name (SAN) field:

   Include the following hostnames:

   - HPE Ezmeral Runtime Enterprise Controller hostname.
   - Common HPE Ezmeral Runtime Enterprise Gateway hostname.
   - If the deployment has more than one Gateway host, include the additional HPE Ezmeral Runtime Enterprise Gateway hostnames.

   If Platform HA is enabled, also include the following hostnames:

   - The HPE Ezmeral Runtime Enterprise Shadow Controller hostname.
   - If your deployment uses a cluster IP address, then also include the hostname associated with that cluster IP address.

2. Place both the host SSL certificate and the private key on the Controller host.

   If your deployment has a Shadow Controller, ensure that you create the the same directory paths on the Shadow controller that you create on the Controller. The files are copied from the Controller to the Shadow Controller, but the copy operation will fail if the path does not exist on the Shadow Controller.

   The certificate and key files must be readable by the webserver process, according to Linux file permissions and the SELinux configuration.

   - **RHEL/CentOS:**

     A standard way to do this is to assign 644 permissions to the certificate and key files and place them in the `/etc/pki/tls/certs` directory.

     However, you can place the files in the directory of your choice.

- **SLES:**

  The commands in this procedure assume you are using the following standard directories:

  ```
  /etc/pki/tls/certs
  /etc/pki/tls/private
  ```

  However, you can place the files in the directories of your choice.

  Create the following folders on the parent directory:

  ```
  mkdir -p /etc/pki/tls
  mkdir -p /etc/pki/tls/certs
  mkdir -p /etc/pki/tls/private
  ```

  Assign 755 permissions to the certificate and key files:

  ```
  chmod 755 /etc/pki/tls
  chmod 755 /etc/pki/tls/certs
  chmod 755 /etc/pki/tls/private
  ```

3. Execute `ssl.sh`, specifying the file paths to the certificate and key files.

   ```
   /opt/bluedata/bundles/hpe-cp-*/startscript.sh -a
   ssl --ssl-cert=<filepath-to-cert> --ssl-priv-key=<filepath-to-key> --ss
   l-ca-data=<filepath-to-ca-data>
   ```

   The `--ssl-ca-data=<filepath-to-ca-data>` argument is optional. The argument specifies the filepath to the certificate authority data.

   The script supports both HA and non-HA environments.

   Example:

   ```
   /opt/bluedata/bundles/hpe-cp-*/startscript.sh -a
   ssl --ssl-cert=/etc/pki/tls/certs/server.crt --ssl-priv-key=/etc/pki/tls/
   private/server.key
   ```

4. Verify that the HPE Ezmeral Runtime Enterprise web interface accepts HTTPS connections and that unsecure HTTP connections are no longer accepted.

5. If your HPE Ezmeral Runtime Enterprise deployment has one or more existing Kubernetes clusters, change the secret that used by the `hpecp-agent` operator to communicate with the control plane for creating services:

   On each Kubernetes Master node, execute the following commands:

   ```
   URL=$(kubectl -n hpecp get secrets/hpecp-session-secret -o
   jsonpath='{.data.k8s-cluster-services-url}' | base64 --decode)
   MOD_URL=$(echo -n $URL | sed 's/http/https/g' | base64 -w 0)
   kubectl -n hpecp patch
   secret hpecp-session-secret --type='json' -p="[{\"op\" :
   \"replace\" ,\"path\" : \"/data/k8s-cluster-services-url\" ,\"value\" :
   \"$MOD_URL\"}]"
   ```

   The preceding commands fetch the current secret, change `http` to `https`, and then update the secret with the modified option.

6. In the **Gateway Settings** tab, enable SSL termination. You can use either the same SSL certificate file you created or obtained at the beginning of this procedure, or a separate SSL certificate file.

More information

Gateway Settings Tab on page 757

Enabling Platform High Availability

Platform High Availability (HA) protects your HPE Ezmeral Runtime Enterprise a failure of the Controller host. Hewlett Packard Enterprise recommends that you enable HA for the HPE Ezmeral Runtime Enterprise Controller before you create Kubernetes clusters.

Prerequisites

Required access rights: Platform Administrator

New Deployments: In a new deployment of HPE Ezmeral Runtime Enterprise, the prerequisites to enabling platform HA are the following:

- You have completed installing HPE Ezmeral Runtime Enterprise and completed Platform Controller Setup on page 861 on the Controller host.
- You have two hosts that conform to the requirements for controller hosts and to the high-availability requirements listed in Host Requirements. These two hosts will become the Shadow Controller and the Arbiter.
- Hewlett Packard Enterprise recommends enabling platform High Availability shortly after initial installation, before adding a large number of Kubernetes hosts.

  Hewlett Packard Enterprise recommends enabling platform High Availability before creating any Kubernetes clusters, including an HPE Ezmeral Data Fabric on Kubernetes cluster. Kubernetes clusters that were created before enabling platform HA might not send data to the correct host after an HA failover. If you want to enable platform HA without deleting existing Kubernetes clusters, contact Hewlett Packard Enterprise Support for assistance.
- If the Controller and the Shadow Controller hosts are to be on the same subnet, in order for the cluster IP address to function correctly, the external switch connecting the hosts to the network must support gratuitous ARP.
- If a cluster IP address is not provided, the Controller and the Shadow Controller are not required to be on the same subnet.

Changing HA Hosts: If you want to change the hosts used for Shadow Controller and Arbiter roles after platform HA has been enabled, you must disable HA protection and then re-enable HA protection using the updated IP addresses and hostnames.

Re-enabling Platform HA: If you are re-enabling platform HA after disabling platform HA in an existing deployment, the prerequisites are the following:

- If platform HA was disabled while the Shadow Controller host was offline, when the faulty hardware is replaced and HA protection is re-enabled, both of the following are required:
  - The original Arbiter host must be redesignated as the Arbiter.
  - The new Shadow Controller host must use the same IP address as the previous Shadow Controller host.
- If platform HA was disabled while the Arbiter host was offline, when the faulty hardware is replaced and HA protection is re-enabled, both of the following are required:
  - The original Shadow Controller host must be redesignated as the Shadow Controller.

- The new Arbiter host must use the same IP address as the previous Arbiter host.

About this task

When enabling platform High Availability on page 132 for a new HPE Ezmeral Runtime Enterprise deployment, you will add two hosts. The hosts you add become the Shadow Controller and Arbiter hosts. The hosts can not be used for any other purpose.

Procedure

1. If you have not already done so, add the hosts that will become the Shadow Controller and Arbiter hosts to the deployment.

   See Adding the Shadow Controller and Arbiter Hosts on page 742.

2. Enter Lockdown mode as described in Lockdown Mode on page 916.

3. On the **Controllers & HA** screen, select **Enable HA**.

4. Enter values in **Cluster IP**, **Cluster Name**, or both, as appropriate:

   - If the Controller and Shadow Controller hosts are in different subnets, then you must leave the **Cluster IP** field blank. By leaving both the **Cluster IP** and **Cluster Name** fields blank, you can access the web interface by navigating to `http://<gateway_ip>` or `https://<gateway_ip>`, as appropriate, where `<gateway_ip>` is the IP address of a Gateway host. See Gateway Hosts on page 106.

   - If the Controller and Shadow Controller hosts are on the same subnet, then you can enter an available IP address to use as the cluster IP address in the **Cluster IP** field.

     The cluster IP address must be in the same subnet as the Controller host and cannot be in use by any other resource.

     If you do not supply a cluster IP address, if you have defined a cluster name in the **Cluster Name** field, then you can access the web interface by navigating to `http://<cluster-name>` or `https://<cluster-name>`, as appropriate. This cluster name must be mapped to the cluster IP address in a user-accessible DNS server. You can also access the web interface using a Gateway IP address.

5. Use the **Shadow Controller** and **Arbiter Node** menus to select a host for each role.

   If the deployment has three hosts, then after you select a host as Shadow Controller or Arbiter, the remaining host is automatically assigned to the other role.

   If there are more than three hosts, no automatic assignment occurs. You can select the Shadow Controller and Arbiter hosts in any order.

   You cannot remove or modify the Shadow Controller or Arbiter host while platform HA is enabled.

6. Click **Submit**.

   The **Controllers** tab displays the message **HA Setup in progress**. This process may take up to 30 minutes to complete, depending on a number of factors. During HA setup, this page reloads and you are signed out. To see updated status, sign in and view this page.

   If you want more detailed information about the setup process, click the **Details** button to open the **HA Setup Details**.

   After the setup process completes, a message appears informing you that HPE Ezmeral Runtime Enterprise is running in High Availability mode, and reminding you to begin using the cluster IP address or cluster name to sign in to the web interface.

   If you installed the Network Manager service while installing the base OS on the hosts, then this service will stop because it conflicts with the High Availability monitoring services.

7. Click **Click here to migrate to Cluster Name** link in the message.

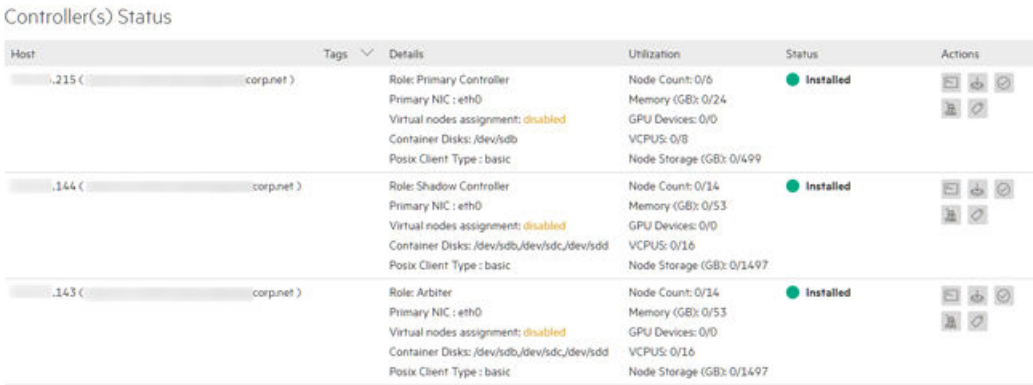
   Clicking the **Click here to migrate to Cluster Name** link in this message logs you out of the web interface and returns you to the sign-in screen using the cluster IP address.

8. Sign in to HPE Ezmeral Runtime Enterprise.

9. Exit Lockdown mode as described in Lockdown Mode on page 916.

Results

The newly added Shadow Controller and Arbiter will appear in the **Controllers & HA** screen.



After enabling HA, Hewlett Packard Enterprise recommends that you use either the cluster IP address or cluster name to sign into the web interface. Doing so will automatically connect you to the Controller host (during normal operation) or the Shadow Controller host (when a Controller host failure triggers HA protection). If the Controller host fails, then you will not be able to access the web interface using the IP address of that host.

If enabling High Availability fails, then the fields in the **HA Setting** section of the **Controllers & HA** screen reappear, and the deployment continues to run with a single Controller host. Contact Hewlett Packard Enterprise Support for assistance.

Related reference

High Availability on page 132
High availability (HA) in deployments of HPE Ezmeral Runtime Enterprise is divided into platform controller HA, gateway HA, and cluster HA.

Adding the Shadow Controller and Arbiter Hosts

This article describes adding hosts to be used as the Shadow Controller and Arbiter in deployments of HPE Ezmeral Runtime Enterprise. Hosts are assigned the roles of Shadow Controller and Arbiter when you enable Plaform High Availability.

Prerequisites

- **Required access rights:** Platform Administrator
- You have two hosts that conform to the requirements for controller hosts and to the high-availability requirements listed in Host Requirements.

  Hewlett Packard Enterprise recommends that the Controller and Shadow Controller hosts share the same configuration (CPU, RAM, storage, OS, etc.).

  CAUTION:

  Installing HPE Ezmeral Runtime Enterprise on any host that does not meet all applicable requirements may lead to unpredictable behavior and/or data loss.

- If you want the installer for HPE Ezmeral Runtime Enterprise to automatically configure firewall rules to open the required ports listed in Port Requirements on page 809, install the and enable the `firewalld` service before you add the host.

About this task

The following procedure describes how to add the hosts that will become the Shadow Controller and Arbiter hosts to HPE Ezmeral Runtime Enterprise.

These control plane hosts are not Kubernetes hosts. The hosts can not be used for any other purpose. You assign the hosts the Shadow Controller or Arbiter role in the procedure Enabling Platform High Availability on page 740.

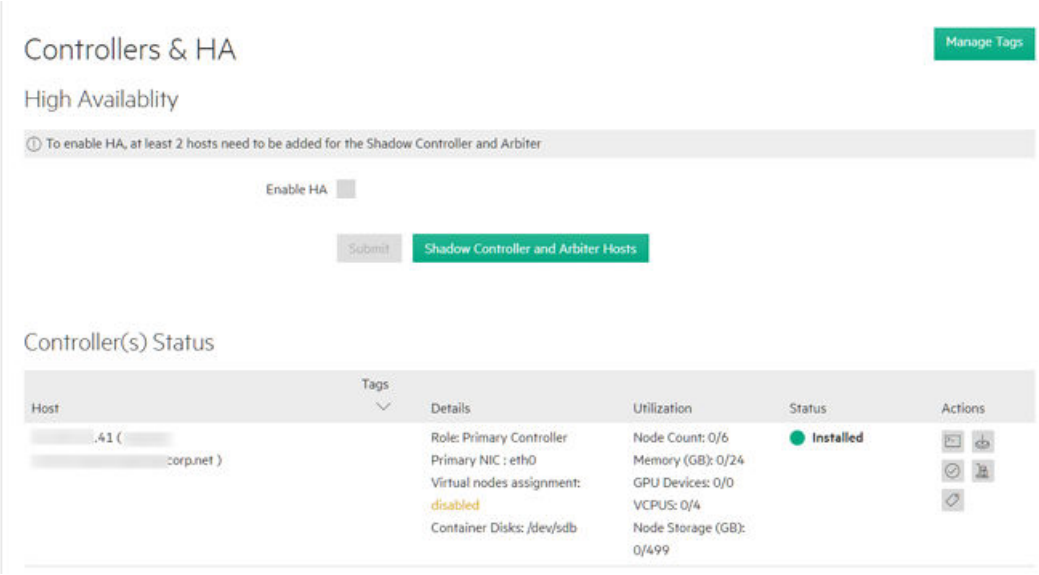
CAUTION:

HPE Ezmeral Runtime Enterprise performs numerous configuration changes to the host during installation that are required in order for the platform to function. These changes are not completely reversible and might impact any other applications and processes that are currently running on the host.

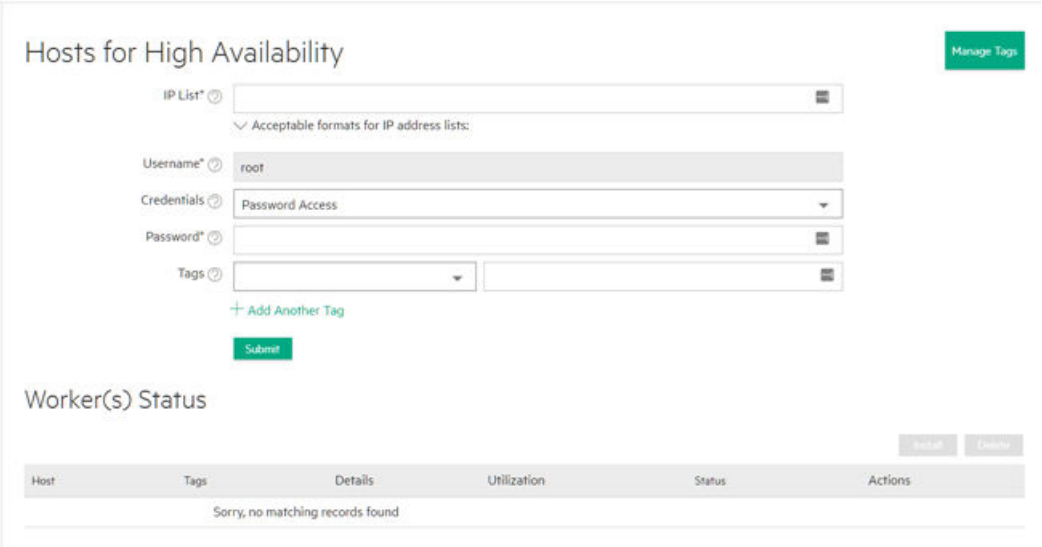
To avoid possible disruptions to your business process, Hewlett Packard Enterprise strongly recommends that you install HPE Ezmeral Runtime Enterprise on a host that is not being used for any other purpose.

Procedure

1. Install HPE Ezmeral Runtime Enterprise on the hosts.
   - If your environment is running the SSHD service, add the public key. See Installing Hosts Using Passwordless SSH on page 750.
   - If your environment does not allow key-based SSH login, see Agent-Based Host Installation on page 746.
2. In the **High Availability** section of the **Controllers & HA** screen, click **Shadow Controller and Arbiter Hosts**.



The **Hosts for High Availability** screen appears.



Select the Hosts

3. Enter the IP addresses of the hosts that you are adding in the **IP List** field.

4. Select the credentials that will be used to access the host.

   - **Agent-based installation:** If you installed the agent on the hosts as described in Agent-Based Host Installation on page 746, then you will not see any credential or key options. Proceed to the next step.

   - **Password access:** In the **Credentials** menu, select **Password Access**. and then enter the password for the hosts you are adding in the **Password** fields. The password must be valid for the username in the **User name** field.

   - **SSH Key:** If the hosts already have a public key installed to allow password-free access (see Installing Hosts Using Passwordless SSH on page 750), upload the private key:

     a. In the **Credentials** menu, select **SSH Key Based Access**

b. Click the **Browse** button to open a standard **File Upload** dialog, then browse for and open the key file.

c. If the key requires a pass phrase, enter that phrase in the **Passphrase** field.

The uploaded private key will be used for initial host access only, and the key will not be permanently stored.

5. (Optional) Apply host tags to the hosts.

   For more information about host tags, see About Tags on page 545.

6. Click **Submit**.

   The hosts that you are adding appear in the **Worker(s) Status** table.

   When the **Status** for each host is **Bundle completed**, proceed to the next step.

Define Storage for the Hosts

7. Define the storage for each host.

   Hosts that will become Shadow Controller or Arbiter hosts must have ephemeral storage (node storage) defined.

   a) In the **Actions** column for the host, click the **Edit** icon (pencil).

      The **Advanced Worker** settings dialog appears.

   b) In **Select one or more available disk(s) for Node Storage**, select the drives that you want to add.

   c) Click **Set**.

      The selected drives are added to the deployment.

Install the Hosts

8. Enter lockdown mode.

   See Lockdown Mode on page 916.

9. Install the hosts in the HPE Ezmeral Runtime Enterprise deployment:

   a) Verify the host fingerprint (MD5 hash). See Public Key Infrastructure on page 134 for information about the PKI.

   b) Select the hosts to install in the **Worker(s) Status** table, and then click the **Install** button.

      A confirmation dialog appears.

   c) Click **OK** to proceed.

Results

While the installation proceeds, the **Install Scheduled** and then the **Installing** bar appear in the **Worker(s) Status** table for the selected hosts. This status changes to **Installed** when the installation is complete.

The hosts are now ready to be assigned to the Shadow Controller or Arbiter role. See Enabling Platform High Availability on page 740.

If host installation fails because of a security error, then check the local times on the Controller and the hosts you are adding. If these times are significantly different, then set the local time on the new host to match the local time on the Controller host, and then begin the installation process again.

Related tasks

Enabling Platform High Availability on page 740
Platform High Availability (HA) protects your HPE Ezmeral Runtime Enterprise a failure of the Controller host. Hewlett Packard Enterprise recommends that you enable HA for the HPE Ezmeral Runtime Enterprise Controller before you create Kubernetes clusters.

Related reference

The Controllers & HA Screen on page 754
The **Controllers & HA** screen enables Platform Administrators to configure platform high availability, and add the Shadow Controller and Arbiter hosts, as needed.

More information

Hosts for High Availability Screen on page 751
Lockdown Mode on page 916

Agent-Based Host Installation

Prerequisites

- Hewlett Packard Enterprise recommends that you update the host to latest OS packages (e.g. yum update) before installing HPE Ezmeral Runtime Enterprise.
- These instructions assume that the Controller host was installed with the option `--worker-agent-install`. If that was not done and if you do not want to reinstall the Controller host with that option specified, then please contact Hewlett Packard Enterprise Technical Support for possible options.

About this task

If your environment does not allow key-based SSH, then you must run the command line agent described in this procedure on each host before adding the host.

NOTE: If your environment does allow key-based SSH and the `PubkeyAuthentication` parameter is set to `true` on the Controller host, then you may bypass this procedure and proceed directly to adding the public key in Adding the Shadow Controller and Arbiter Hosts on page 742.

Procedure

1. If you encountered any errors while pre-checking and/or installing HPE Ezmeral Runtime Enterprise on the Controller from the command line, then be sure to replicate the same remediation steps on each host you will be adding before proceeding with the installation.
2. Manually copy the HPE Ezmeral Runtime Enterprise binary (.bin) from `http://<controller-ip>/repos/common-hpe-cp-<os>-release-<version>-<build>.bin` to each host that you are adding, where:
   - `<controller_ip>` is the IP address of the Controller host.
   - `<os>` is the operating system (for example `rhel`).
   - `<version>` is the .bin version.
   - `<build>` is the specific .bin build number.

   The remainder of this article refers to this .bin file as `<common>.bin`.
3. Make the .bin file executable by executing the command `chmod a+x <common>.bin.`

4. Download the `.parms` file from `http://<controller-ip>/repos/agent-install-worker.parms`

5. Modify the relevant settings in `/tmp/agent-install-worker.parms` to the appropriate values. The `.parms` file with these edits will be used on every host.

   - **Set the Controller host parameter:**.

     - Because Platform high availability is not yet enabled, in the `Platform HA not configured` section, do the following:

       a. Uncomment the `HAENABLED` line, set `HAENABLED` (Platform High Availability Enabled) field to `false`

       ```
       ##################################################################
       ##############
       #                          Platform HA not
       configured                           #
       # Ensure the appropriate parameters are uncommented and set in
       this section    #
       # when Platform HA is not
       enabled.                                                    #
       ##################################################################
       ##############

            ## Is PLHA enabled?
               #HAENABLED=false
       ```

       b. Uncomment the line that contains the `CONTROLLER` setting and provide the Controller host IP address.

       ```
       ## Controller node's IP address.
          #CONTROLLER=<Controller IP address>
       ```

       c. Uncomment the line that contains the `CONTROLLER_HOSTNAME` setting and provide the Controller hostname.

       ```
       ## Controller node's FQDN.
          #CONTROLLER_HOSTNAME=<FQDN of controller>
       ```

     - If the deployment uses a Cluster IP address, then you must uncomment the following setting and set `CLUSTERIP` (Cluster IP address); otherwise, you can leave it commented.

       ```
       ## The cluster IP address.
          #CLUSTERIP=<Cluster IP address>
       ```

       Uncomment the following and provide the Controller IP address.

       ```
       ## Controller node's IP address. A failover to okay but, his node
       must be alive
       ## for a worker to be added.
          #CONTROLLER=<Controller IP address>
       ```

       Uncomment the following and provide the Controller hostname.

       ```
       ## Controller node's FQDN.
          #CONTROLLER_HOSTNAME=<FQDN of controller>
       ```

- **Set the installation userid and groupid parameters:** If you have already a defined an HPE `userid` and `groupid` system account on the Controller host, then you will need to set the `BLUEDATA_USER` and `BLUEDATA_GROUP` values accordingly.

```
#########################################################################
#########
#                         Installation user and
group                          #
# All nodes in the HPE physical cluster must be installed the same
user.       #
# Specify this if the common bundle is not being executed by the same
user as  #
# the user that will be running the HPE services. Please refer to
the          #
# System requirements guide for information on permissions required
for a      #
# non-root user to install and run HPE
software.                                #
#########################################################################
#########

   #BLUEDATA_USER=root
```

Note: Uncomment this and then provide the user id, as appropriate.

```
#BLUEDATA_GROUP=root
```

Note: Uncomment this and then provide the group id, as appropriate.

- **Set other miscellaneous parameters:** Set the following parameters to match the Controller host settings.

```
#######################################################################
#########
#                           Miscellaneous
parameters                           #
#
        #
#######################################################################
#########

   ## Automount root on the controller node. It must be the same on
the worker too.
      CONTROLLER_AUTOMOUNT_ROOT=/net/
```

Note: Modify this if needed.

```
## Bundle flavor used to install the controller. This may be either
'minimal' or
 ## 'full'
   CONTROLLER_BUNDLE_FLAVOR=minimal
```

Note: Modify this if needed.

```
## Skip configuring NTP? 'true' or 'false'
  #NO_NTP_CONFIG=false
```

Note: Modify this, as appropriate.

```
## If the controller was configured with proxy information, please
specify it
## for the worker too.

        #PROXY_URL=
```

Note: Set this if the Controller is configured with a proxy.

```
#NO_PROXY=
```

Set this if the Controller was configured with the `--no-proxy` option during installation.

```
## Controls whether the server should rollback to a clean state when
an error
## is encountered during installation. Setting it to 'false' helps
with debugging
## but the server should be manually cleaned up before re-attempting
the
## installation.

## Values: 'true' or 'false'.
    #ROLLBACK_ON_ERROR='false'

   # If the controller was configured with --dockerrootsize that is
different from 20
   # specify it here.
     DOCKER_ROOTSIZE=20
```

Note: Set this, if applicable.

6. Set the Erlang parameter.

```
ERLANG_COOKIE=value contained in <controller>$HOME/.erlang.cookie
```

7. Copy the modified version of the `.parms` file onto the new hosts.

8. On each host, execute the installer precheck using one of the following commands, where `<A.B.C.D>` is the IP address of the host, and `<name>` is the FQDN of the host:

   - `/tmp/<common>.bin --params /tmp/agent-install-worker.parms --nodetype worker --worker <A.B.C.D> --workerhostname <name>`

9. If needed, remediate any issues reported by the pre-check installer script, and then re-run the same pre-check script until all tests pass or until you have accounted for any warnings.

10. Copy the file `/opt/bluedata/keys/authorized_keys` from the Controller host to the same location on the new host, with the same owner/group, permissions, and SELinux context.

    This must be done after running the common install .bin.

Results

After the installation completes, you should see the message `Successfully prepared server as a HPE worker node.` Proceed to add the public key as described in See Installing Hosts Using Passwordless SSH on page 750.

If the installation fails, then erase HPE Ezmeral Runtime Enterprise from the host by executing the command `/tmp/<common>.bin --erase` (or `sudo /tmp/<common>.bin --erase`, or `erase SUDO_PREFIX="mysudo"; /tmp/<common>.bin --erase`. The instructions contained in Step 1 Troubleshooting on page 860 for the Controller host can also help you remediate problems on this host or hosts.

Installing Hosts Using Passwordless SSH

The topics in this section describe using the passwordless SSH method to install the Shadow Controller and Arbiter hosts in HPE Ezmeral Runtime Enterprise.

Prerequisites

- **Required access rights:** Platform Administrator
- The environment allows key-based SSH login.

About this task

You must upload the public key to the hosts before uploading the corresponding private key to HPE Ezmeral Runtime Enterprise to add those hosts via the web interface.

Procedure

1. Use a tool such as `ssh-keygen` to create a public key and a corresponding private key for each host. The keys must be in PEM format.

   For example, to use ssh-keygen on a Linux computer, enter the following command:

```
ssh-keygen -m PEM -t rsa #
```

2. Copy the `id_rsa.pub` file to the host.

3. Add the public key to the list of authorized keys for the root user by executing a command similar to the following:

```
root worker# cat id_rsa.pub >> /root/.ssh/authorized_keys
```

4. Test the key by executing the following command (where `worker` is the hostname or IP address of the Worker host) from the Controller host:

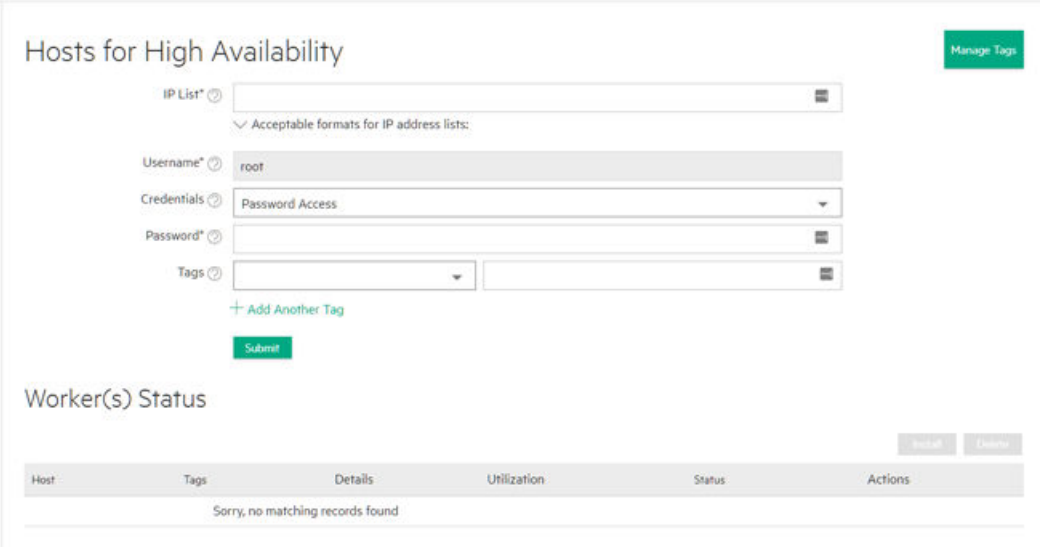
```
ssh -i id_rsa root@worker
```

   This command should log the root user into the Worker host without being prompted for a password.

Results

The public key is installed on the host. You can continue to **Select the Hosts** in Adding the Shadow Controller and Arbiter Hosts on page 742.

Hosts for High Availability Screen



The upper portion of this screen contains the following functions:

- **Manage Tags:** Clicking this button opens the **Tags** screen, which allows you to view, add, and delete host tags that are available in this deployment of HPE Ezmeral Runtime Enterprise. See The Tags Screen.

- **IP List**: Enter the IP addresses of the hosts that will become the Shadow Controller and Arbiter nodes. Enter multiple IP addresses separated by commas, such as the following:

  ```
  10.10.1.1, 10.10.1.2
  ```

- **Tags:** Click to select an existing host tag. For example, if the hosts reside in different racks, you can use a tag called `rack` to specify the host location, such as `rack_a`, `rack_b`, or `rack_c`.

  To select a tag, use the menu to select the tag to add, and then enter the desired value in the text field. If you add a tag by mistake, click the **Delete** icon (trash can) to remove the tag. You can also add one or more additional tags by clicking the **Add Another Tag** link and repeating this process for each tag you want to assign to the hosts. You may only assign one value per tag.

- **Credentials:** Enter the credentials to be used to access the host. Credentials are either a valid username and password or an SSH key.
- **Submit** Click to begin the process of adding the specified hosts to the deployment.

The lower portion of this screen contains the **Install** and **Delete** buttons, and the **Worker(s) Status** table.

- **Install:** Selecting one or more hosts in the following table and then clicking this button installs the selected hosts, if they have not already been installed.
- **Delete:** Selecting one or more hosts in the following table and then clicking this button removes the selected hosts from the deployment. You may also delete an individual host by clicking the **Delete** icon for that host.

  If platform High Availability is enabled, then you cannot delete a Controller, Shadow Controller, or Arbiter host.

The **Worker(s) Status** table displays the following information and functions for the hosts that become the Shadow Controller and Arbiter host:

- **Host:** IP address and hostname of the host.
- **Tags:** Lists any tags assigned to the host and the value assigned to each tag.
- **Details:** This column displays the following information:
  - **Role:** Role the host is playing in the deployment, such as **Shadow Controller**, or **Arbiter**.
  - **Memory (GB):** Amount of RAM available to the host.
  - **GPUs:** Number of GPU devices available to the host, if any.
  - **Cores:** Number of CPU cores available to the host.
  - **Virtual nodes assignment:** For Controller, Shadow Controller, and Arbiter hosts, virtual nodes assignment is always **disabled**.
  - **Storage status:** Type and status of local shared-storage service on this host, if any.
- **Utilization:**
- **Status:** Status of the host.
  - **Connecting:** HPE Ezmeral Runtime Enterprise is attempting to connect to the listed host.
  - **Running bundle:** HPE Ezmeral Runtime Enterprise has successfully connected to the listed hosts and is preparing the host.
  - **Bundle completed:** HPE Ezmeral Runtime Enterprise has completed preparing the listed host. f you added the hosts by mistake, you may remove them by clicking the **Delete** icon (trash can).
  - **Unlicensed:** If adding the hosts would cause the total number of CPU cores to exceed the amount of cores allowed by your HPE Ezmeral Runtime Enterprise license, then this status will appear in an orange bar, and you will not be able to continue installing the host. To resolve this issue, either add a new license that allows the increased number of CPU cores (see License Tab on page 798), or delete the hosts you are trying to add.
- **Actions:** The following functions are available:
  - **Update Tags:** Clicking the **Update Tags** icon (tag) for a host opens the **Update Tags** dialog, which allows you to add, edit, and remove tags for that host. See Updating Tags for a Host.

- **Delete:** Deletes the host.

  If platform High Availability is enabled, then you cannot delete a Controller, Shadow Controller, or Arbiter host.

Disabling Platform High-Availability

Prerequisites

- **Required access rights:** Platform Administrator
- All hosts must be online.

  If you want to disable HA protection while the Shadow Controller or Arbiter host is offline, contact Hewlett Packard Enterprise Support for assistance. The Support team will perform some manual operations to allow the management service to ignore the offline host and allow you to proceed with disabling HA Protection.

About this task

Disabling platform HA protection has the following effects:

- The cluster IP address is disabled; you must sign in using the Controller IP address or a Gateway.
- The HPE Ezmeral Runtime Enterprise deployment is no longer protected against Controller host failure.

Procedure

1. Enter Lockdown mode as described in Lockdown Mode on page 916.
2. On the **Controlers & HA** screen, clear the **Enable HA** check box.

   A confirmation popup appears. Click **OK** to confirm.
3. Click **Submit** to begin disabling HA protection.

   You will be signed out automatically.
4. Sign in to the web interface using the Controller IP address.

   The **HA** tab will display the message **HA disable in progress**. Disabling HA might require up to 30 minutes to complete. If you want more information about the HA disable process, click the **Details** button to open the **HPE HA Disable Details** dialog.
5. After the HA disable process completes, exit site lockdown as described in Lockdown Mode on page 916.

Disabling HA can be rejected for one of several reasons:

- **Shadow Controller failure:** Normally, disabling HA protection will delete the database replicas on the Shadow Controller and Arbiter hosts. If the Shadow Controller host is offline due to a hardware failure, then database deletion will fail.

  If disabling HA fails because the Shadow Controller is offline, contact Hewlett Packard Enterprise Support for assistance.

  When the faulty hardware is replaced and HA protection is re-enabled, the original Arbiter host must be redesignated as the Arbiter, and the new Shadow Controller host must use the same IP address as the previous Shadow Controller host.

- **Arbiter failure:** Normally, disabling HA protection will delete the database replicas on the Shadow Controller and Arbiter hosts. If the Arbiter host is offline due to a hardware failure, then database deletion will fail.

  If disabling HA fails because the Arbiter is offline, contact Hewlett Packard Enterprise Support for assistance.

  When the faulty hardware is replaced and HA protection is re-enabled, the original Shadow Controller host must be re-designated as the Shadow Controller, and the new Arbiter host must use the same IP address as the previous Arbiter host.

Related reference

High Availability on page 132
High availability (HA) in deployments of HPE Ezmeral Runtime Enterprise is divided into platform controller HA, gateway HA, and cluster HA.

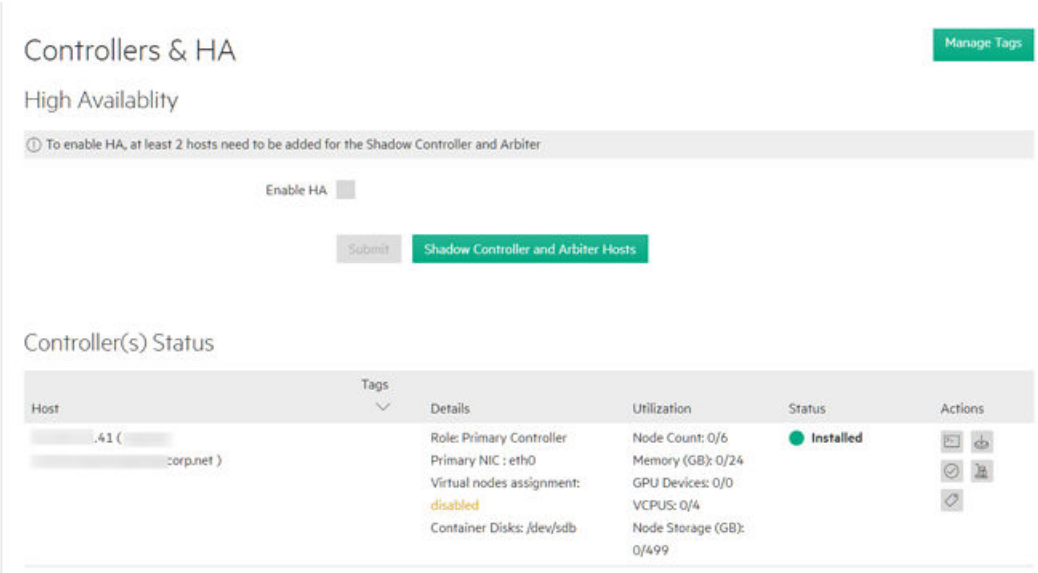
The Controllers & HA Screen

The **Controllers & HA** screen enables Platform Administrators to configure platform high availability, and add the Shadow Controller and Arbiter hosts, as needed.

Selecting **Controllers & HA** in the main menu opens the **Controllers & HA** screen, which enables the Platform Administrator to configure platform High Availability (HA) and add the hosts that will become the Shadow Controller and Arbiter hosts.

Enabling platform HA protects HPE Ezmeral Runtime Enterprise in the event of a Controller host failure.

Enabling platform High Availability protection does not protect Kubernetes clusters. For information about platform high availability compared to cluster high availability, see High Availability on page 132.



HA Setting

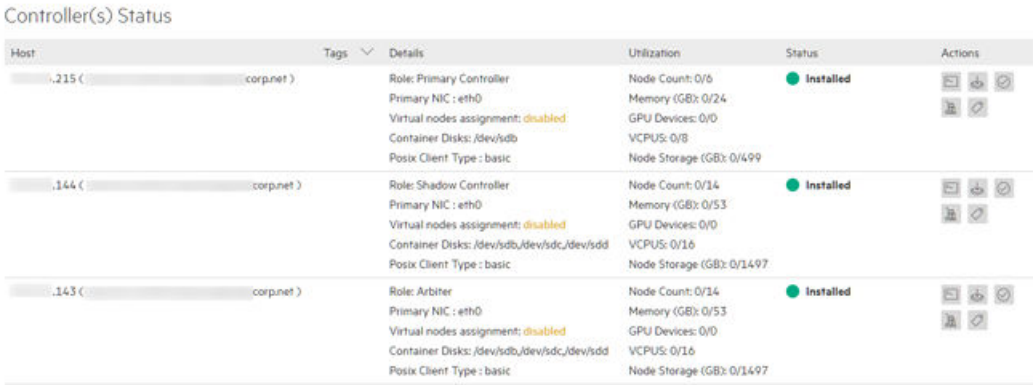
The **HA Setting** section contains the **Enable HA** check box. When selected, a form is displayed. Enabling HA is not attempted until you complete and submit the form.

See the following:

- Enabling Platform High Availability on page 740
- Disabling Platform High-Availability on page 753

Controller(s) Status

The **Controller(s) Status** section displays information about the Controller host. If platform HA is enabled, the **Controller(s) Status** displays information about the Controller, Shadow Controller, and Arbiter hosts.



Gateway LB

The topics in this section describe the settings and tasks related to the gateway load balancers in HPE Ezmeral Runtime Enterprise.

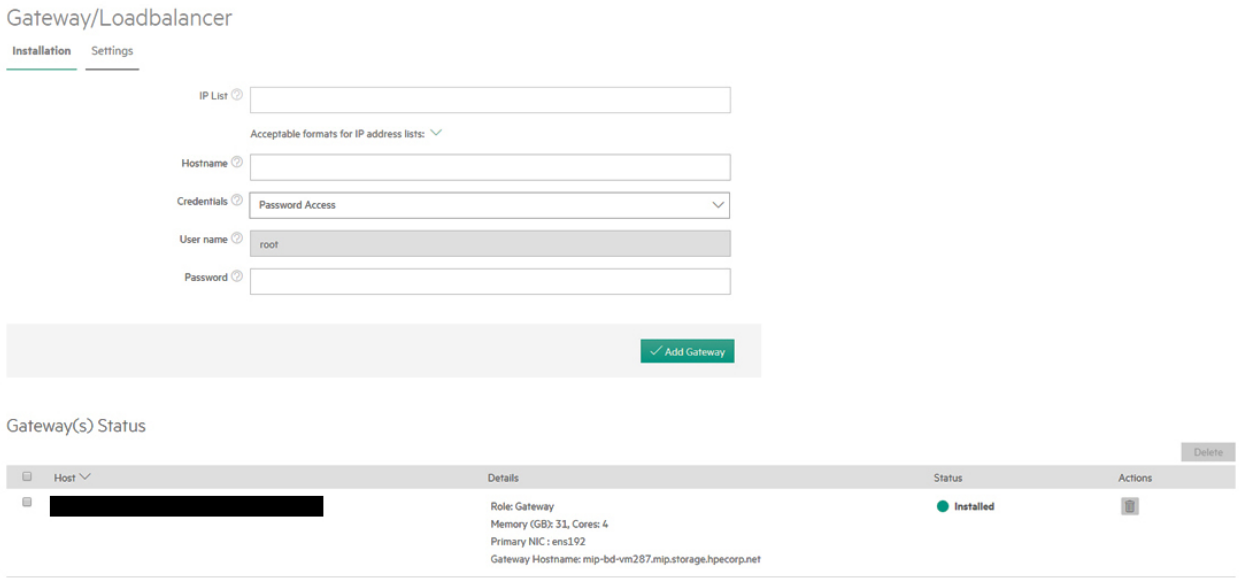
The Gateway/Load Balancer Screen

Selecting **Gateway LB** in the main menu opens the **Gateway/Load Balancer** screen, which enables the Platform Administrator to perform the following functions:

- **Manage Gateway hosts:** See Gateway Installation Tab.
- **Manage Gateway settings:** See Gateway Settings Tab.

Gateway Installation Tab

The **Gateway Installation** tab of the **Gateway/Load Balancer** screen (see The Gateway/Load Balancer Screen) lists the Gateway hosts in the deployment and allows you to install and remove Gateway hosts.



NOTE: This screen only lists Gateway hosts.

For information about working with Kubernetes hosts, see The Kubernetes Hosts Installation Screen and Kubernetes Worker Installation Overview.

This upper portion of this screen contains the following functions:

- **IP List:** Enter the IP addresses for one or more Gateway Worker hosts in the **Worker IP** field.
- **Hostname:** When you add one or more Gateway Worker hosts, you must specify a hostname in the **Hostname** field. The **Gateway Hostname** must be **all lower case** set as per the Linux hostname naming convention. If you specify one hostname for more than one Gateway IP address, then either the DNS server or external load balancer will load-balance requests to the hostname among all of the Gateway hosts on a round-robin basis. For example, if three Gateway hosts are sharing a hostname, then Users 1-3 will access virtual nodes/containers via Hosts 1-3, respectively, User 4 will access virtual nodes/containers using Host 1, and so on. You may add additional Gateway hosts to an existing set at any time by specifying the IP addresses of the Gateway hosts you are adding and then entering an existing Gateway hostname. You can use Gateway hostnames in one of two ways:
  - Configure the Gateway hostname in the corporate DNS server to resolve to the IP addresses of the Gateway hosts.
  - Configure an external load balancer with rules to point traffic to the IP addresses of the Gateway hosts. These rules are not enforced.

NOTE: Clusters created before the addition of Gateway hosts will not receive service endpoints on those hosts.

- **Credentials:** This is where you add either a valid username and password or SSH key in order to access the Gateway Worker hosts being added.
- **Add Gateway:** Clicking this button begins the process of adding the specified Gateway hosts. See Installing a Gateway Host.

The lower portion of this screen contains the **Delete** buttons, and the **Gateways Status** table.

- **Delete:** Selecting one or more Gateway hosts in the following table and then clicking this button removes the selected Gateway hosts. See Deleting a Gateway Host. You may also delete an individual Gateway host by clicking the **Delete** icon for that host, as described below.

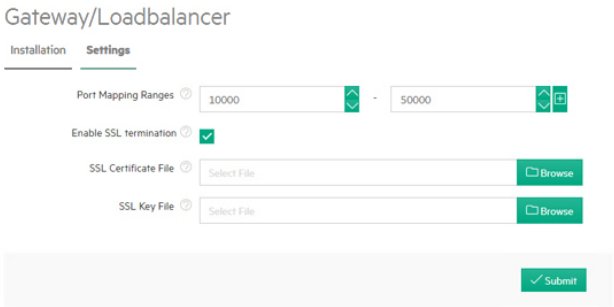
The table displays the following information and functions for each Gateway host:

- **Host:** IP address and hostname of the Worker host.
- **Details:** This column displays the following information:
  - **Role:** Role the host is playing, which will be **Gateway**.
  - **Memory (GB):** Amount of RAM available to the host.
  - **Cores:** Number of CPU cores available to the host.
  - **Gateway Hostname:** Hostname of the Gateway host.
  - **Status:** Status of the Gateway host. This column will say **Installed** for all fully-installed Gateway Worker hosts.
- **Actions:** The following function is available:
  - **Delete:** Clicking the **Delete** icon (trash can) for a Gateway host removes that host. See Deleting a Gateway Host.

last-updated: May 06, 2024

Gateway Settings Tab

The **Gateway Settings** tab of the **Gateway/Load Balancer** screen (see The Gateway/Load Balancer Screen) allows you to specify a port mapping range for use with Gateway hosts (see Gateway Hosts) and to configure Gateway host SSL termination for non-secure (HTTP) cluster services running in pods.



This tab has the following functions:

- **Port Mapping Range:** The **Port Mapping Range** fields allow you specify a custom range of ports to use for accessing services via Gateway hosts when using a private, non-routable network. These ports must be reserved for exclusive use by the deployment. The maximum allowable port range is 10000-50000. When working with port ranges:
  - To add a port range, click the **Add** icon (plus sign) next to a port range.
  - To remove a port range, click the **Remove** icon (minus sign) next to the port range you with to remove.
  - To assign a single port, enter the same number in the start and end fields. For example, to reserve port 10100, then enter 10100 twice, as shown above.
  - Port ranges must be non-contiguous. For example, if you add ports 20000 to 20500 in one range and then add ports 20501 to 21000 in another range, then these ranges will be combined into a single range that consists of ports 20000-21000.
  - Any range that overlaps with an existing range will be ignored. In the above example, if you add the range 20400-25000, then that range will not be added, nor will it add ports 21001-25000 to the range 20000-21000.

NOTE: You must remove all Kubernetesclusters before modifying the port range settings.

- **SSL Termination:** Checking this check box configures the Gateway hosts to provide SSL termination for non-secure (HTTP) cluster services running in virtual nodes (containers).
- **SSL Certificate File:** When the **SSL Termination** check box is checked, this field allows you to specify an HTTPS certificate file. Clicking the **Browse** button allows you to navigate to and select a new or replacement certificate. This may be a self-signed certificate, if desired; however, this may trigger HTTPS warnings in your web browser.

NOTE: Encrypted (password-protected) certificates or keypairs for SSL termination are not supported. SSL termination will fail if you add an encrypted certificate.

- **SSL Key File:** When the **SSL Termination** check box is checked, this field allows you to specify an RSA private key file. Clicking the **Browse** button allows you to navigate to and select a new or replacement RSA key file.

Click the **Submit** button when you have finished making changes to the gateway settings.

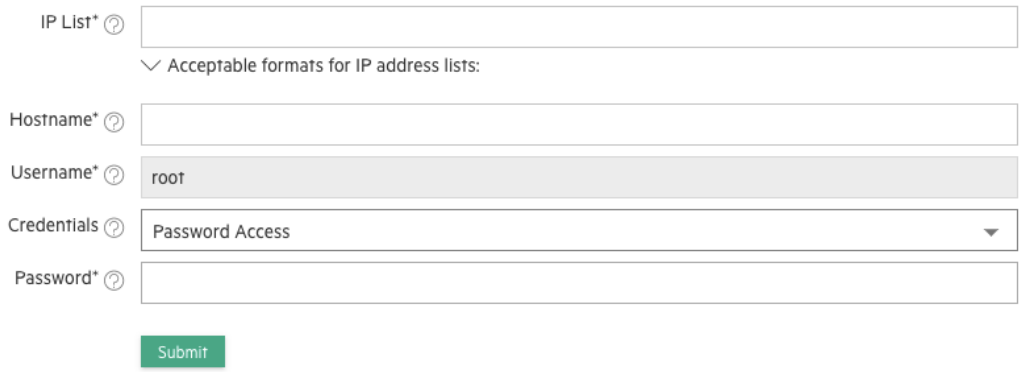
NOTE: Gateway hosts will perform SSL tunneling (as opposed to SSL termination) for cluster services that have explicit HTTPS endpoints.

Installing a Gateway Host

To add one or more Gateway hosts, you will use the top portion of the **Gateway/Load Balancer** screen (see The Gateway/Load Balancer Screen).

Before adding one or more Gateway hosts, ensure that the hosts conform to the requirements described in Host Requirements on page 813.

If the `firewalld` service is installed and enabled on the Controller, and the `firewalld` service is installed and enabled on all hosts before they are added to the deployment, the installer for HPE Ezmeral Runtime Enterprise automatically configures firewall rules to open the required ports.



To select the hosts:

1. If you do not see the **User name** and **Password** fields, then follow the instructions found in Agent-Based Gateway Installation; otherwise, proceed to Step 2.

2. Enter the IP addresses of the Gateway hosts that you are adding in the **IP List** field. You may select one or more hosts as follows:

   - **Single IP address:** Enter a properly formatted IP address, such as `10.10.1.1`. This will add a single host.

   - **Multiple IP addresses:** Enter the first three octets of the IP addresses, and then separate each digit of the fourth octet with a comma, such as `10.10.1.1,2,5,8`. In this example, four Gateway hosts with IP addresses of 10.10.1.1, 10.10.1.2, 10.10.1.5, and 10.10.1.8 will be added.

   - **Multiple IP addresses:** Enter multiple IP addresses separated by commas, such as `10.10.1.1`, `10.10.1.2`, `10.10.1.5`, `10.10.1.8`. In this example, four Gateway hosts with the same IP addresses as the previous example will be added.

   - **IP address range:** Enter an IP address range, such as `10.10.1.1-8`. In this example, eight Gateway hosts with IP addresses from 10.10.1.1 to 10.10.1.8 will be added.

   - **Combination:** Use a combination of the above methods, such as `10.10.1.1`, `10.10.1.2,5,8,10.10.1.9-12`.

   NOTE: You may only perform one set of Gateway host additions to one or more hosts at once. To save time, consider adding all of the Gateway hosts at once by entering multiple IP addresses as described above.

3. Select how to access the Gateway hosts. Your available options are:

- **Password access:** Check the **Password Access** radio button and then enter the password for the Gateway hosts you are adding in the **Password** fields. The password must be valid for the username in the **User name** field.
- **SSH Key:** If the Gateway hosts already have a public key installed to allow password-free access, then you may check the **SSH Key based Access** radio button. Upload the private key by clicking the **Browse** button to open a standard **File Upload** dialog that allows you to browse for and select the key file. If the key requires a pass phrase, enter that phrase in the **Passphrase** field. The uploaded private key will only be used for initial host access and will not be permanently stored.

NOTE: If Gateway installation fails because of a security error, then check the local times on the Controller and Gateway Hosts. If these times are significantly different, then set the local time on the Gateway host to match the local time on the Controller host, and then begin the installation process again.

4. Click the **Add Gateway** button to install the selected Gateway hosts.

The selected Gateway hosts are installed. The **Gateway(s) Status** table displays the following information for each host you are adding:

- **Host:** IP address and hostname of the Gateway host.
- **Details:** Information about the Gateway host (RAM, CPU cores, etc.).2
- **Status:** Current status of the Compute host, which updates as the installation progresses. This will appear as one of the following:
  - **Connecting:** HPE Ezmeral Runtime Enterprise is attempting to connect to the listed Gateway hosts.
  - **Running bundle:** HPE Ezmeral Runtime Enterprise has successfully connected to the listed Gateway hosts and is preparing the hosts.
  - **Bundle completed:** HPE Ezmeral Runtime Enterprise has completed preparing the listed Gateway hosts, which are ready to be added to the deployment. If you added the hosts by mistake, you may remove them by clicking the **Delete** icon (trash can).
  - **Installed:** The Gateway host is available for use.
- **Actions:** Once the Gateway hosts are reviewed, a **Delete** icon (trash can) will appear next to that Gateway. See Deleting a Gateway Host.

Troubleshooting

If you experience issues when installing a Gateway host, then access the following logs:

- **Controller host:**
  - **Gateway Installer log:**`/var/log/bluedata/install/addworker.out_.log`.
  - **Xtrace file:** This file is a verbose, line-by-line description of the exact commands used by the script to both get data and determine the outcome of each test. This file will be stored in `/var/log/bluedata/addworker/install.out_.log.xtrace`.
- **Gateway host**
  - **Gateway setup log:** `/var/log/bluedata/install/worker_setup_<timestamp>`
  - **Gateway Xtrace set-up file:** `/var/log/bluedata/install/worker_setup_<timestamp>.xtrace`

Begin reading these logs from top to bottom.

Stop at the first ERROR you find. This first error can often cause further problems downstream, and taking a start-to-finish approach (instead of working your way back from the tail end of the log file) may help you solve one error that in turn resolves a series of cascading errors. If the problem is obvious, then correct the problem and re-run the installer.

If you are unable to resolve the problems on your own, then contact Hewlett Packard Enterprise for support. You may be asked to provide the these installer logs and xtrace files.

Agent-Based Gateway Installation

If your environment does not allow password-less SSH, then you must run the command line agent described in this article on each Gateway host being added to your deployment before adding the host(s) using the web interface.

NOTE: These instructions assume that the Controller host was installed with the option `--worker-agent-install`. If that was not done and if you do not want to reinstall the Controller host with that option specified, then please contact HPE Technical Support for possible options.

NOTE: If your environment does allow password-less SSH and the `PubkeyAuthentication` parameter is set to `true` on the Controller host, then follow the instructions in Installing a Gateway Host.

To install the agent on each Gateway host:

1. If you encountered any errors while pre-checking and/or installing HPE Container Platform on the Controller from the command line, then be sure to replicate the same remediation steps on each Gateway host you will be adding before proceeding with the installation.

2. Manually copy the HPE Container Platform Enterprise binary (.bin) from `<controller-ip>/opt/bluedata/bundles/common-cp-<version>-<build>.bin` to each Gateway host that you will adding, where:

   - `<controller_ip>` is the IP address of the HPE Container Platform Controller host.
   - `<version>` is the HPE Container Platform version.
   - `<build>` is the specific HPE Container Platform build number.

   NOTE: If you cannot download the file via http, then you may retrieve it from `/opt/bluedata/bundles` on the Controller host.

   NOTE: The remainder of this article will refer to this .bin file as `<common>.bin`.

3. Make the .bin file executable by executing the command `chmod a+x <common>.bin`.

4. Copy the `.erlang.cookie` file from the Controller host to the Gateway host(s) you are adding with the same owner/group, permissions, and SELinux context. This file is located in the home directory of the user who installed HPE Ezmeral Runtime Enterprise. This step is required to allow secure communications between hosts.

5. Download the `.parms` file from `http://<controller-ip>/repos/agent-install-worker.parms`

6. Modify the relevant settings in `/tmp/agent-install-worker.parms` to the appropriate values. The `.parms` file with these edits will be used on every Gateway host.

- **Set the Controller host parameter:** The Controller parameter settings vary based on whether or not the deployment has platform HA enabled.
  - If platform HA is not enabled, then you must set the `HAENABLED` (Platform High Availability Enabled) field to `false` and provide both the Controller IP address and hostname in the `Platform HA not configured` section.

```
##################################################################
###########
                        #                           Platform HA not
configured                            #
                        # Ensure the appropriate parameters are
uncommented and set in this section    #
                        # when Platform HA is not
enabled.                                                  #

##################################################################
###########

                        ## Is PLHA enabled?
                        #HAENABLED=false
```

Note: Uncomment this.

```
## Controller node's IP address.
                        #CONTROLLER=<Controller IP address>
```

Note: Uncomment this and provide the Controller host IP address.

```
## Controller node's FQDN.
                        #CONTROLLER_HOSTNAME=<FQDN of controller>
```

*Note: Uncomment this and provide the Controller hostname. The **Controller hostname** must be **all lower case** set as per the Linux hostname naming convention.*

- If platform HA is enabled, then you must set the `HAENABLED` (Platform High Availability Enabled) field to `false` and provide both the IP address and hostname for the Controller, Shadow Controller, and Arbiter hosts in the `Platform HA configured` section.

  Further, if the deployment uses a Cluster IP address, then you must set `CLUSTERIP` (Cluster IP address); otherwise, you can leave it commented.

```
#######################################################################
###########
                        #                              Platform HA
configured                              #
                        # Ensure the appropriate parameters are
uncommented and set in this section    #
                        # when Platform HA is not
enabled.                                               #

#######################################################################
###########

                        ## Is Platform HA enabled?
                        #HAENABLED=true
```

  Note: Uncomment this.

```
## The cluster IP address.
                        #CLUSTERIP=<Cluster IP address>
```

  Note: Uncomment this if the deployment uses a Cluster IP address.

```
## Controller node's IP address. A failover to okay but, his node
must be alive
                        ## for a worker to be added.
                        #CONTROLLER=<Controller IP address>
```

  Note: Uncomment this and provide the Controller IP address.

```
## The original shadow controller node's IP address. This node must
be alive for
                        ## the worker node to be added.
                        #SHADOWCTRL=<Shadow IP address>
```

  Note: Uncomment this and then provide the Shadow IP address.

```
## The arbiter node's IP address. This node must be alive for the
worker node to
                        ## be added.
                        #ARBITER=<Arbiter IP address>
```

  Note: Uncomment this and then provide the Arbiter IP address.

```
## Controller node's FQDN.
                        #CONTROLLER_HOSTNAME=<FQDN of controller>
```

  Note: Uncomment this and then provide the Controller hostname.

```
## Shadow controller node's FQDN.
                        #SHADOW_HOSTNAME=<FQDN of Shadow>
```

*Note: Uncomment this and then provide the Shadow hostname. The **Shadow hostname** must be **all lower case** set as per the Linux hostname naming convention.*

```
## Arbiter node's FQDN.
                        #ARBITER_HOSTNAME=<FQDN of Arbiter>
```

*Note: Uncomment this and then provide the Arbiter hostname. The **Arbiter hostname** must be **all lower case** set as per the Linux hostname naming convention.*

- For a Gateway set:

```
NODE_TYPE=proxy
                        GATEWAY_NODE_IP=<gateway_ip>
                        GATEWAY_NODE_FQDN=<gateway_host_fqdn>
```

- **Set the Erlang parameter:**

```
ERLANG_COOKIE=value contained in <controller>$HOME/.erlang.cookie
```

- **Set the installation userid and groupid parameters:** If you have already a defined HPE `userid` and `groupid` system account on the Controller host, then you will need to set the `BLUEDATA_USER` and `BLUEDATA_GROUP` values accordingly.

```
##########################################################################
#########
                #                        Installation user and
group                           #
                # All nodes in the HPE physical cluster must be
installed the same user.  #
                # Specify this if the common bundle is not being
executed by the same user as  #
                # the user that will be running the HPE services.
Please refer to the     #
                # System requirements guide for information on
permissions required for a      #
                # non-root user to install and run HPE
software.                          #

##########################################################################
#########

                #BLUEDATA_USER=root
```

Note: Uncomment this and then provide the user id, as appropriate.

```
#BLUEDATA_GROUP=root
```

Note: Uncomment this and then provide the group id, as appropriate.

- **Set other miscellaneous parameters:** Set the following parameters to match the Controller host settings.

```
#########################################################################
#########
                #                            Miscellaneous
parameters                            #

#
        #

#########################################################################
#########

                ## Automount root on the controller node. It must be
the same on the worker too.
                CONTROLLER_AUTOMOUNT_ROOT=/net/
```

Note: Modify this if needed.

```
## Bundle flavor used to install the controller. This may be either
'minimal' or
                ## 'full'
                CONTROLLER_BUNDLE_FLAVOR=minimal
```

Note: Modify this if needed.

```
## Skip configuring NTP? 'true' or 'false'
                #NO_NTP_CONFIG=false
```

Note: Modify this, as appropriate.

```
## If the controller was configured with proxy information, please
specify it


                ## for the worker too.


                #PROXY_URL=
```

Note: Set this if the HPE Controller is configured with a proxy.

```
## Controls whether the server should rollback to a clean state when
an error
                ## is encountered during installation. Setting it to
'false' helps with debugging
                ## but the server should be manually cleaned up before
re-attempting the
                ## installation.

                ## Values: 'true' or 'false'.
                #ROLLBACK_ON_ERROR='false'

                # If the controller was configured
with --dockerrootsize that is different from 20
                # specify it here.
                DOCKER_ROOTSIZE=20
```

Note: Set this, if applicable.

7. Copy the modified version of the `.parms` file onto every new Gateway host.

8. On each Gateway host, execute the installer binary using the following command, where <gateway_ip> is the IP address of the host, and <gateway_hostname> is the FQDN of the host:

```
./ common-cp-<version>-<build>.bin /tmp/
agent-install-worker.parms --nodetype proxy --gateway-node-ip
<gateway_ip> --gateway-node-hostname <gateway_hostname>
```

   where:

   - `<version>` is the .bin version.
   - `<build>` is the .bin build number
   - `<gateway_ip>` is the IP address of the Gateway host.
   - `<gateway_hostname>` is the hostname of the Gateway host. *The **Gateway Hostname** must be **all lower case** set as per the Linux hostname naming convention*.

   NOTE: HPE recommends to update to latest OS packages (e.g. yum update) before installing the HPE Ezmeral Runtime Enterprise product.

9. If needed, remediate any issues reported by the above installer script, and then re-run the same installer script until all tests pass or until you have accounted for any warnings.

10. Copy the file `/opt/bluedata/keys/authorized_keys` from the Controller host to the same location on the new Worker host, with the same owner/group, permissions, and SELinux context. This must be done after executing the common install .bin.

    After the installation completes, you should see the message `Successfully configured a Gateway node.`

    If the installation fails, then erase HPE Ezmeral Runtime Enterprise from the host by executing the command `/tmp/<common>.bin --erase` (or `sudo /tmp/<common>.bin --erase`, or `SUDO_PREFIX="mysudo"; /tmp/<common>.bin --erase`. The instructions contained in Step 1 Troubleshooting for the Controller host can also help you remediate problems on this host or hosts.

    If the installation succeeds, then proceed to Step 2 in Installing a Gateway Host. Be sure to only specify the IP address(es) that you added using this agent-based installation method. You can ignore Step 3, because agent-based installations do not required credentials.

Deleting a Gateway Host

Deleting a Gateway host completely removes it from the deployment. To delete one or more Gateway host(s):

1. Access the **Gateway/Load Balancer** screen (see The Gateway/Load Balancer Screen).

2. In the **Gateway(s) Status** table, either:

   - Remove a single Gateway host by clicking the **Delete** icon (trash can) for the host you want to remove.
   - Remove multiple Gateway hosts by selecting the affected hosts and then clicking the **Delete** button above the table.

HPE Ezmeral Runtime Enterprise removes the selected Gateway host(s).

The User Authentication Screen

The User Authentication screen enables the Platform Administrator to configure user authentication settings in HPE Ezmeral Runtime Enterprise.

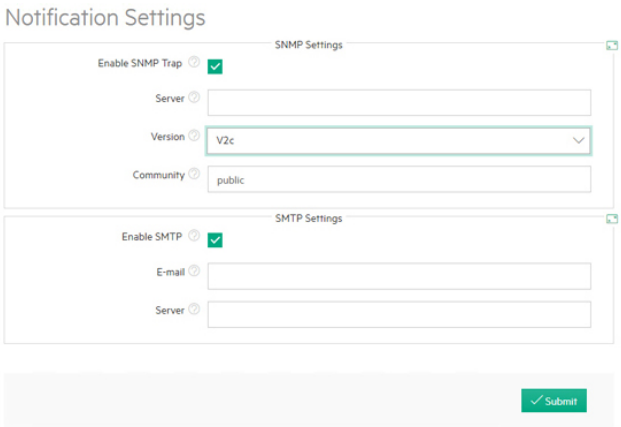
Selecting **Authentication** in the main menu opens the **User Authentication** screen, which enables the Platform Administrator to configure user authentication settings. See Configuring User Authentication Settings.

The Notification Settings Screen

The Notification Settings screen enables Platform Administrators to configure the HPE Ezmeral Runtime Enterprise deployment to deliver Nagios alerts.

Nagios monitors the state of services running on the Controller and Worker hosts. Nagios can be configured to send alerts when it detects that a service has failed or has been restarted. These notifications are in addition to the information presented in the **Services** tab of the Platform Administrator **Dashboard** screen (See Dashboard - Platform Administrator on page 570).

Selecting **Notifications** in the main menu opens the **Notification Settings** screen, which enables you to configure the deployment to deliver Nagios alerts via SNMP trap and/or SMTP (email).



This screen has the following functional areas:

- **SNMP Settings:** Configures SNMP trap settings. See SNMP Settings.
- **SMTP Settings:** Configures email settings. See SMTP Settings.

To configure notifications:

1. Check the **SNMP Settings** and/or **SMTP Settings** check box(es).
2. Enter the appropriate parameters.
3. Click the **Submit** button to save your changes.

   The **Verify** button will appear at the top of the tab when your changes have been saved.
4. Verify your notification settings as described in Verification, below.

SNMP Settings

The **SNMP Settings** area of the **Notification** tab allows you to specify the IP address or FQDN and any additional parameters required to connect to a server that receives and displays SNMP traps from Nagios. This area contains the following functions:

- **Enable SNMP Trap:** Checking this check box enables Nagios alert delivery via SNAP traps. Some or all of the fields described below when this check box is checked.

- **Server:** Enter either the IP address or FQDN of the SNMP server that will receive and display the SNMP traps from Nagios.
- **Version:** Use the Version pull-down menu to select the SNMP version to use (**V2c** or **V3**).
- **Community:** This field only appears when the **SNMP Version** is set to **V2c**. Enter the SNMP server community in this field. This field is not available when the Version is set to V3.
- **Engine ID:** Enter the ID of the SNMP engine in this field.
- **Security Level:** This pull-down menu allows you to specify the security level to use. The available options are:
  - **authPriv:** The username must be both authenticated and private. This is the highest security level.
  - **authNoPriv:** The username must be authenticated but not private. This is a medium security level.
  - **noAuthNoPriv:** The username is neither authenticated nor private. This is the lowest security level.
- **Username:** Valid username recognized by the SNMP server.
- **Authentication Protocol:** This field only appears when the **Security Level** is set to either **authNoPriv** or **authPriv**. Use this pull-down menu to select the protocol to use when authenticating the **Username** (**SHA** or **MD5**).
- **Authentication Passphrase:** This field only appears when the **Security Level** is set to either **authNoPriv** or **authPriv**. Enter the passphrase that will be used to authenticate the **Username** in this field.
- **Privacy Protocol:** This pull-down menu only appears when the **Security Level** is set to **authPriv**. Use this pull-down menu to select the privacy protocol to use for the **Username** provided above (**AES** or **DES**).
- **Privacy Passphrase:** This field only appears when the **Security Level** is set to **authPriv**. Enter the privacy passphrase that will be used for the **Username** in this field.

See Verification, below.

SMTP Settings

The **SMTP Settings** area of the **Notification** tab allows you to configure emailed alerts from Nagios.

NOTE: You may also configure email alerts within the Nagios interface, as described in Setting Up Nagios Email Alerts.

This area contains the following functions:

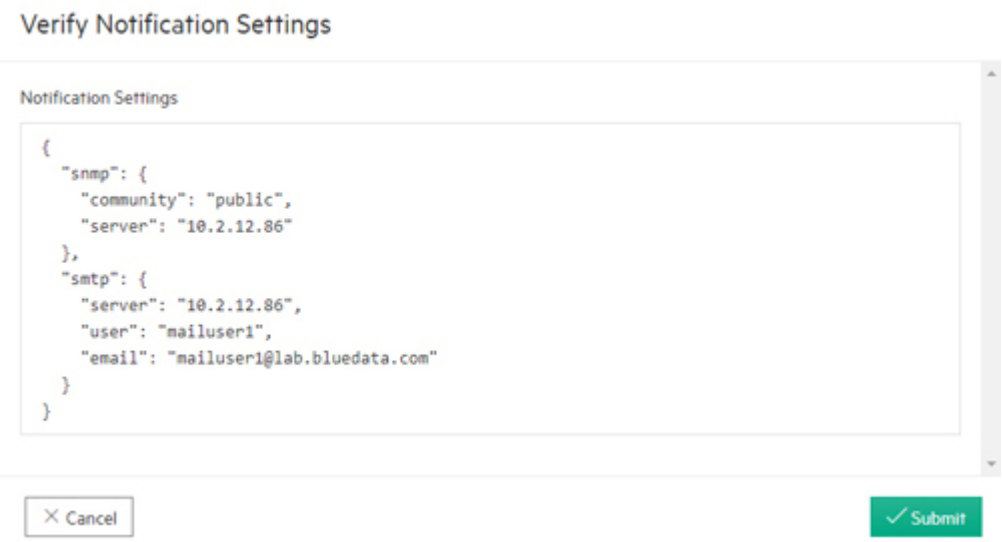
- **Enable SMTP:** Checking this check box enables alert delivery via email. Clearing this check box disables email Nagios notifications from HPE Ezmeral Runtime Enterprise.
- **E-mail:** This field appears when the **Enable SMTP** check box is checked. Enter a valid email address in this field.
- **Server:** This field appears when the **Enable SMTP** check box is checked. This field is only necessary if the Controller is not able to resolve an email address to a mail server, i.e. if the email domain name does not have an MX record in the DNS server that is visible to the Controller.
- **Username:** Use this field to change the email sender ID. If this is not set, then the **From:** field will be populated with `admin@<servername>`.

- **Password:** Use this in conjunction with the **Username** field if it is necessary to authenticate with the target email server in order to send an email from the specified user.

For example, if your email address is `itadmin@example.com`, and `example.com` has an MX record in a DNS server that is visible to the HPE Ezmeral Runtime Enterprise Controller, then you need only specify the **E-mail** field. However, if you want to send an email from yourself to a mail alias via a service such as Google Gmail, then then **E-mail** address would be something like it-all@gmail.com. In this example, the **Server** is `smtp.gmail.com:587`, the **Username** is itadmin@gmail.com and the **Password** is the one you normally use to log in to Gmail. See Verification, below.

Verification

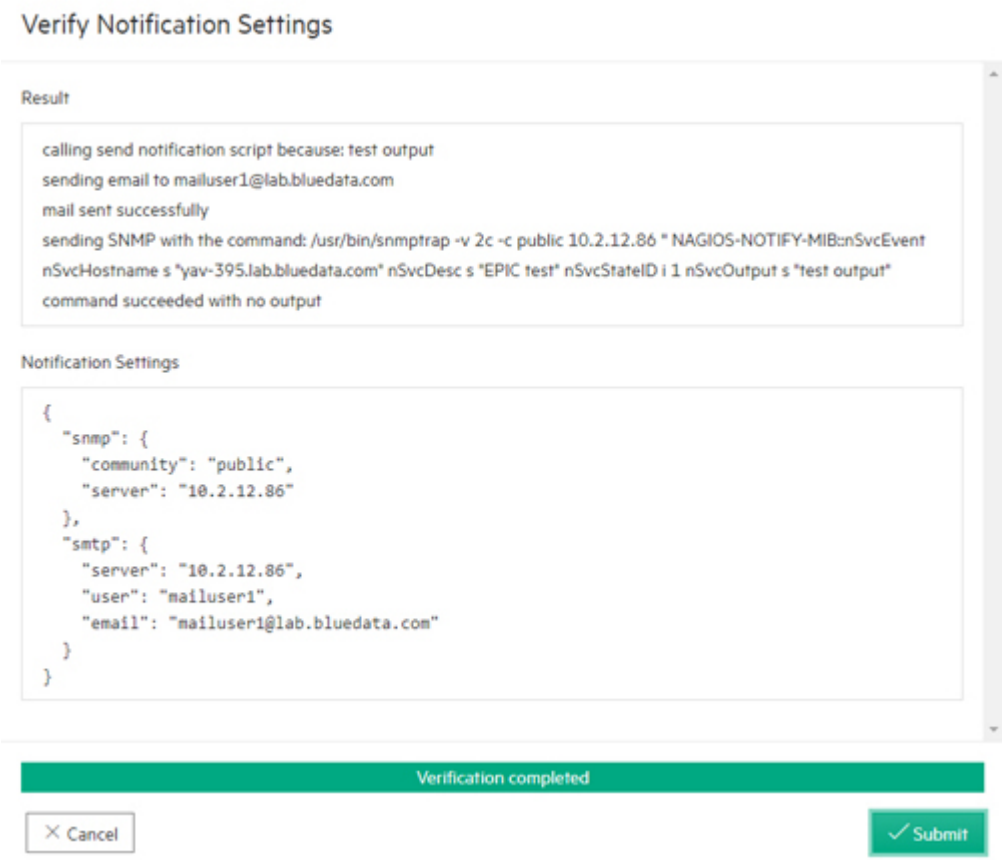
The **Verify** button appears once you have configured SNMP and/or SMTP notifications as described in SNMP Settings and SMTP Settings and then clicked the **Submit** button. Clicking this button opens the **Verify Notification Settings** popup.



This popup has the following functions:

- **Notification Settings**: This area of the popup displays a JSON blob with your SNMP and/or SMTP notification settings.

- **Submit:** Review the JSON blob and then click the Submit button to verify your settings. A test SNMP trap and/or SMTP email will be sent depending on the configuration parameters to the targets that have been specified. The **Verify Notification Settings** popup will display the results of the verification in the **Results** section. Any errors that appear should contain details on the nature of the mis-configuration. SNMP is a UDP protocol and does not always provide detailed messages. Be sure to check for typos in **Engine ID** or **Passphrase** fields.



NOTE: SNMP and SMTP alerts only apply to services on the base hosts. They do not apply to any services running inside virtual nodes/containers. Some applications in virtual containers (e.g. Cloudera Manager) natively support notifications; check the documentation f or those solutions for details.

NOTE: Nagios only sends a failure after three consecutive detections of a particular service being down, in order to avoid false failures. Also, only one failure notification will be sent for services that are determined to be “flapping” or starting and stopping frequently within a given short time period.

SMTP Settings - Troubleshooting

When you enter the necessary configuration details and click the **Verify** button, the test mail is displayed. Sometimes, you may face one of the following issues:

- **ERROR email failed: timed out**: If you enter an invalid detail or due to network issue, HPE Ezmeral Runtime Enterprise may fail to connect to the specified SMTP server on the given port. When HPE Ezmeral Runtime Enterprise fails to connect, the web UI (and the log file at `/srv/bluedata/nagios/notification.log` on the current primary controller) displays **ERROR email failed: timed out** message.
- **Configuring recipients email addresses for alarms**: To configure email addresses, do the following:
  1. Create a Public distribution list (PDL) within the email service of your organization’s and add individual users as necessary.
  2. Enter the PDL name in the **E-mail** field of SMTP settings.
  3. Click **Verify** button and check the result.

User Management

The topics in this section describe the settings and tasks related to the managing users in HPE Ezmeral Runtime Enterprise.

Platform Administrators can manage the following user settings:

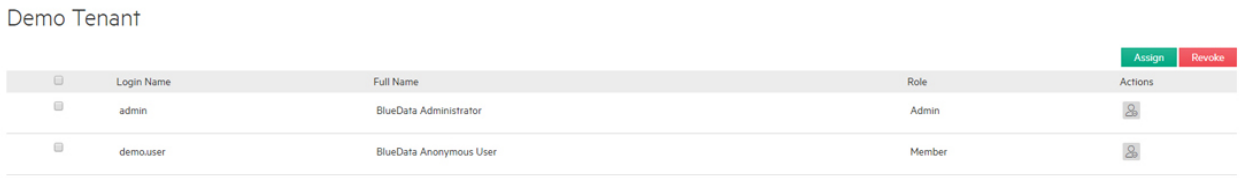
- **Viewing User Assignments:** Viewing the tenants, projects, and roles assigned to each user. A user may have one role per Kubernetes tenant, one role per Big Data tenant, and one role per AI/ML project. See Viewing User Assignments.
- **Assigning/Revoking User Roles:** These articles describe how to assign and revoke user roles based on the authentication method used by the deployment or tenant:
  - **Local authentication:** See Assigning/Revoking User Roles (Local).
  - **LDAP/AD authentication:** See Assigning/Revoking User Roles (LDAP/AD/SAML).
- **User Management:** The **User Management** screen allows you to view and manage users and user sessions. See The User Management Screen and Managing User Sessions.
- **User Details:** Clicking a user in the **User Management** screen opens the **User Details** screen for that user. See The User Details Screen.
- **Adding a New User:** Describes how to add a new user to via the internal user database (as opposed to via LDAP/AD). See Creating a New User (Local).
- **Removing a User:** See Deleting a User.
- **Authentication Settings:** The Configuring User Authentication Settings article describes how to configure user authentication.
- **Accessing LDAP/AD Logs:** You can access detailed logs of LDAP/AD activity as described in Accessing LDAP/AD/SAML Logs.

Viewing User Assignments

If you are a Tenant or Project Administrator, then selecting **Users** in the main menu opens the **Tenant Details** screen, which displays the users who are assigned to the current tenant/project.

NOTE: Platform Administrators who select **Users** in the main menu will access the **User Management** screen. See The User Management Screen.

The **Tenant Details** screen appears as follows when the platform is configured to use platform authentication.



This screen contains the following buttons:

- **Assign:** Clicking this button opens the **Assign Users** screen. See Assigning/Revoking User Roles (Local).
- **Revoke:** Selecting one or more users in the table and then clicking this button revokes the selected users from the tenant/project. A popup warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without revoking the roles for the affected users.

NOTE: If you revoke a user by mistake, you can reassign them to the tenant or project using the **Assign Users** screen. See Assigning/Revoking User Roles (Local).

NOTE: If you use LDAP/AD to authenticate users, then you will manage user assignments on the authentication server, as described in Assigning/Revoking User Roles (LDAP/AD/SAML).

The table on this screen contains the following information and functions:

- **Login Name:** Login name of the user.
- **Full Name:** Full name of the user. This will be blank for LDAP/AD users, or if no name was entered when adding the user.
- **Role:** Role of the user within the tenant or project (**Admin** or **Member**). A user may have one role per tenant or project.
- **Revoke:** Clicking the **Revoke** icon (person) in the **Actions** column revokes the selected user's access to the tenant/project. A popup warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without revoking the user's role for the tenant or project.

NOTE: Users who do not have a role in the current tenant or project will not appear on this screen. These users will appear on the **Assign Users** screen when you click the **Assign** button.

Assigning/Revoking User Roles (Local)

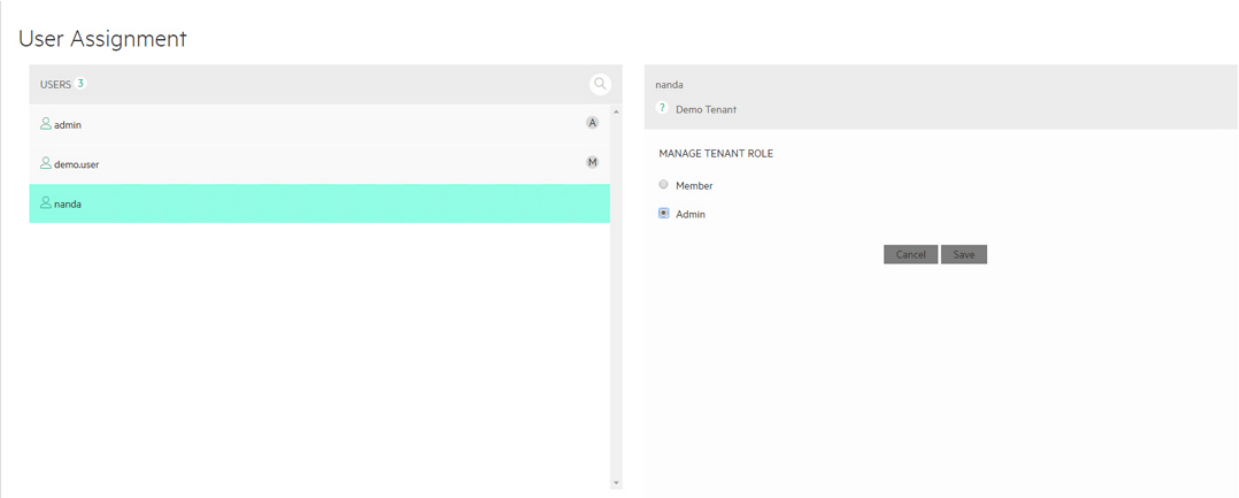
If the deployment **Local** user authentication across either the entire platform or in the current tenant (see The User Authentication Screen and Kubernetes Tenant/Project External Authentication on page 456), then the process of assigning and revoking user roles varies based on your role, as follows:

- **Tenant Administrator:** You can use the **Assign User** screen to assign the Member or Admin roles to users within your own tenant. The appearance and functionality of this screen varies slightly based on your role, as described in Tenant Administrator View.
- **Platform Administrator:** You can use the **Assign User** screen to assign the Member or Admin roles to users across all tenants in the deployment and can also assign the Platform Administrator role. The appearance and functionality of this screen varies slightly based on your role, as described in Platform Administrator View.

NOTE: If you use LDAP/AD to authenticate users, then you will manage user assignments on the authentication server as described in Assigning/Revoking User Roles (LDAP/AD/SAML).

Tenant Administrator View

If you are a Tenant Administrator, then clicking the **Assign** button in the **Tenant Details** screen or selecting **Assign Users** in the **Quick Access** menu opens the **Assign Users** screen. This screen allows you to assign, change, or revoke user access to the current tenant. The **Assign Users** screen appears as shown here for a Tenant Administrator.



To assign a user to the current tenant or change the user's role within the current tenant (such as from Member to Tenant Administrator or vice versa):

1. On the left side of the screen, select the user you want to assign in the **USERS** list. You may also click the **Search** icon (magnifying glass) and then start typing the username into the **Filter** field, and the list of users will update in real time based on your entry.
   - An **A** icon appears by each user who has the Tenant Administrator role assigned to them for the current tenant. A tenant may have multiple administrators. You may either downgrade the role of that user to Tenant Member or remove access to this tenant altogether.
   - An **M** icon appears by each user who has the Tenant Member role assigned to them for the current tenant. A tenant may have multiple members. You may either upgrade the role of that user to Tenant Administrator or remove access to this tenant altogether.
   - Users who do not have any role in the current tenant may be granted either the Tenant Member or Tenant Administrator role. No icon appears next to these users.

   NOTE: It is possible to revoke all roles from a single user. A user with zero assigned roles will not appear in any of the **Tenant Details** screens, but will appear in the **Assign Users** screen. A user must have at least one assigned role in order to be able to log in to the deployment.

2. Selecting a user enables the **User** section on the right side of the screen. The name of the tenant to which you are assigning the user also appears below the username.

3. Check the appropriate radio button to assign a role to the selected user. The available options are:
   - **Member:** Makes the user a non-administrative member of the current tenant.
   - **Admin:** Makes the user a Tenant Administrator of the current tenant.

   NOTE: This function does not store user passwords. The built-in user database or your existing external authentication server will handle user passwords.

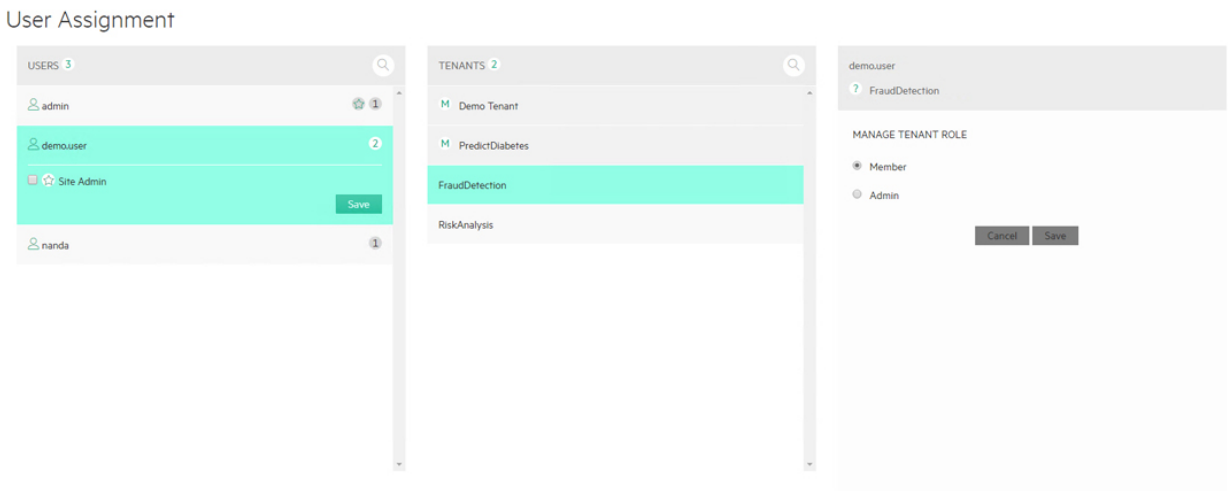
4. If the selected user already has Member or Tenant Administrator access to the current tenant, you will see a **Remove from this Tenant** button at the bottom right of the **Assign Users** screen. Clicking this button revokes the user's role and prevents them from being able to access the current tenant.

A confirmation dialog appears; click **OK** to proceed with the revocation or **Cancel** to cancel.

NOTE: If you revoke a user role and that user has no other role in any other tenant, then that user will not be able to log in until they are assigned at least one role.

Platform Administrator View

If you are a Platform Administrator, then selecting a user followed by clicking the **Assign** button in the **User Details** screen opens the **Assign Users** screen, which allows you to assign, change, or revoke user access across all tenants and to assign/remove the Site Admin role from one or more users. The **Assign Users** screen appears as shown here for a Platform Administrator.



This screen enables you to:

- Assign the Member or Tenant Administrator role to users. See Assigning Member/Tenant Administrator Roles on page 773.
- Assign the Site Admin role to users. See Assigning the Platform Administrator Role on page 774.

Assigning Member/Tenant Administrator Roles

To assign a user role:

1. On the left side of the screen, select the user you want to assign in the **USERS** list. You may also start typing the username into the **Filter** field, and the list of users will update in real time based on your entry.
   - An star icon appears by each user who has the Platform Administrator role assigned to them.
   - The **TENANTS** column displays all of the tenants.
   - When you select a user in the **USERS** column, an **A** icon appears in the **TENANTS** column next to each tenant in which the selected user has the Tenant Administrator role assigned to them for that tenant.
   - When you select a user in the **USERS** column, an **M** icon appears in the **TENANTS** next to each tenant in which the selected user has the Member role assigned to them for that tenant.
   - No icon appears next to any tenant(s) for which the selected user has no role.

   NOTE: It is possible to revoke all roles from a single user. A user with zero assigned roles will not appear in any of the **Tenant Details** screens, but will appear in the **Assign Users** screen. A user must have at least one assigned role in order to be able to log in.

2. Selecting a user and a tenant enables the **User** section on the right side of the screen. Check the appropriate radio button to assign a role to the selected user. The available options are:
   - **Member:** Makes the user a non-administrative member of the selected tenant.
   - **Admin:** Makes the user a Tenant Administrator of the selected tenant.

   NOTE: A user may have one role per tenant. Please see Users and Roles on page 130 for an explanation of the available roles and the privileges associated with each role.

3. Click **Save** to save your changes.

   NOTE: This function does not store user passwords. The built-in user database or your existing external authentication server will handle user passwords.

4. If the selected user already has Member or Tenant Administrator access to the current tenant, you will see a **Remove from this Tenant** button at the bottom right of the **Assign Users** screen. Clicking this button revokes the user's role and prevents them from being able to access the current tenant.

A confirmation dialog appears; click **OK** to proceed with the revocation or **Cancel** to cancel.

NOTE: If you revoke a user role and that user has no other role in any other tenant, then that user will not be able to log in until they are assigned at least one role.

Assigning the Platform Administrator Role

The role assigned to Platform Administrators is called `Site Admin`.

Selecting a user in the **USERS** section also expands that user and displays a **Site Admin** check box for that user. This box is checked if the user already has the Site Admin role assigned to them. It is cleared if they do not have this role.

- Checking this check box and then clicking **Save** assigns the Site Admin role to the selected user, which gives that use Platform Administrator rights. A star icon appears next to this user in the **USERS** section.
- Clearing this check box and then clicking **Save** removes the Site Admin role from the selected user. The star icon disappears from this user in the **USERS** section.

NOTE: This function does not store user passwords. The built-in user database or your existing external authentication server will handle user passwords.

Assigning/Revoking User Roles (LDAP/AD/SAML)

If the Platform Administrator configured the deployment to use LDAP or Active Directory user authentication (see The User Authentication Screen on page 766), then there are two ways to assign/revoke user roles.

NOTE: If the platform handles user authentication, then you will manage user assignments on the authentication server as described in Assigning/Revoking User Roles (Local) on page 771.

LDAP/AD user accounts in the deployment fall into two groups:

- Auto-added from a tenant authentication group: See Automatically Added on page 774.
- Manually added by the Platform Administrator: See Manually Added on page 775.

Automatically Added

The tenant/project roles for users who have been automatically added via a tenant authentication group cannot be changed in the **User Assignment** screen. These users are based on LDAP/AD group

membership, and the deployment grants the roles specified by the tenant authorization groups every time one of these users logs in. The Platform Administrator can temporarily delete an automatically-added user, but the account will be re-created next time the user logs in. See Configuring User Authentication Settings on page 778.

To permanently remove such a user's role in a tenant or project, either remove that user's groups from the tenant's authorization groups, or change the user's group membership at the LDAP/AD server. If that user has a current session, then they will be able to continue accessing the deployment until that session expires; however, a Platform Administrator can end the session at any time, as described in Managing User Sessions on page 777.

Manually Added

The Platform Administrator may choose to manually add an LDAP/AD-based user account for various reasons, such as:

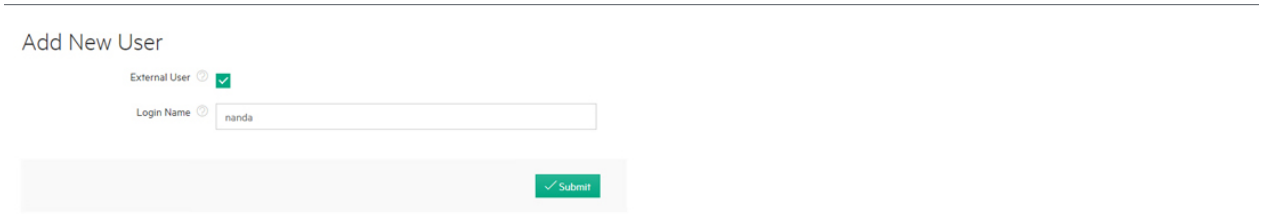
- If the user needs to be granted Platform Administrator privileges.
- If the deployment is not using group-based authentication for tenant/project and container access.
- If the tenant and/or project roles require manual management, as exceptions to group-based authentication settings.

If you are manually adding a user who already has an account that was automatically created by logging in and being granted group-level privileges, then you must first delete that existing user account and then re-add that user account manually, including granting the desired privileges.

NOTE: See User Authentication on page 126 for more information on how the deployment handles user authentication.

To manually add an external user:

1. Open the **User Management** screen (see The User Management Screen on page 791).
2. Click the **Add User** button to open the **Add New User** screen.
3. Check the **External User** check box.
4. Provide the login name of the user in the **Login** field.



5. Click the **Submit** button to save your changes.

   The **User Management** screen refreshes to include the name of the newly-added user.

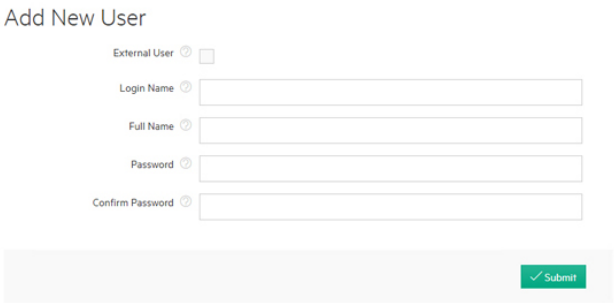
6. Click the **Details** button for the newly-added user to open the **User Details** screen for that user.

7. Click the **Assign** button, and then assign the desired tenant/project roles to the user as described in Assigning/Revoking User Roles (Local) on page 771.

Creating a New User (Local)

If you are a Platform Administrator and the deployment is set to use local authentication (see User Authentication), then clicking the **Add User** button in the **User Management** screen opens the **Add New User** screen.

When the deployment is configured to use platform authentication, then clicking the **Add User** button in the **User Management** screen opens the following **Add New User** screen:



NOTE: If the deployment is configured to use external authentication, then you will need to add the user to your LDAP/AD service. The new user will receive the roles mapped to their assigned group.

To create a new user:

1. If you are manually adding an external (LDAP/AD/SAML) user to the deployment, then see Manually Added; otherwise, proceed to Step 2.

2. Enter a unique user name in the **Login Name** field. This name is case sensitive.

3. Enter the full name of the user in the **Full Name** field.

4. Enter a password in the **Password** field, and then reenter the same password in the **Confirm Password** screen. Passwords are case sensitive. The user may change her or his password as described in Changing Your Password.

When you have finished entering the information for the new user, click **Submit** to save your changes. You may now assign the user to one or more tenants/projects, as described in Assigning/Revoking User Roles (Local).

NOTE: You may only create a new user via this function if the deployment is configured to use the local user database. If you use an external authentication server to manage logins, then you need to create the user account on that authentication server.

Deleting a User

Deleting a user immediately removes that user and prevent them from being able to access the deployment.

About this task

You cannot undelete a user.

Instead of deleting a user, consider simply unassigning them from all roles. This prevents the affected user from being able to log in unless and until you assign them a new role.

Procedure

1. Log in to the web interface as either a Platform Administrator in the **Site Admin** tenant or as a Tenant/Project Administrator of the tenant or project from which you want to delete the user.
2. Open either of the following screens:
3. The **User Management** screen, if you are a Platform Administrator. See The User Management Screen.
4. The **Tenant Details** screen, if you are a Tenant or Project Administrator. See Viewing User Assignments.
5. Either:
   - Select one or more users by checking the appropriate check boxes in the table, and then click the **Delete** button.
   - Click the **Delete** icon (trash can) for a user to delete that user.

   A warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without deleting the user.

Managing User Sessions

If you are a Platform Administrator, then you can view and terminate user sessions as follows:

1. Select **Users** in the main menu to open the **User Management** screen.
2. Open the **Sessions** tab.
3. If needed, you may:
   - Select one or more session(s) in the table and then click the **Delete** button.
   - Click the **Delete** icon for a session.

A confirmation popup appears. Click **OK** to delete the session or **Cancel** to exit without deleting the session.

A user whose session has been terminated will need to log in again as described in Launching and Logging In.

Configuring User Authentication Settings

The deployment can be configured to authenticate users on a platform-wide basis. You may also configure user authentication on a per-Kubernetes-cluster basis. User authentication may be handled by the authentication server or by one or more LDAP/AD severs across one or more domains. The following user authentication settings are thus supported:

- **Platform:** The entire deployment uses the same authentication settings, which may be configured as follows:
  - **Local:** The deployment uses the internal authentication server to authenticate users.
  - **Single LDAP/AD domain, with no failover protection:** The deployment uses a single LDAP/AD domain with a single server to authenticate users. There is no failover protection.
  - **Single LDAP/AD domain, with failover protection:** The deployment uses a single LDAP/AD domain with two or more servers configured, which provides failover protection in case a server goes down or becomes unreachable.
  - **Multiple LDAP/AD domains:** The deployment uses two or more LDAP/AD domains to authenticate users. Each domain may be configured to use either one server (no failover protection) or multiple servers (for failover protection).
- **Kubernetes cluster:** An individual Kubernetes cluster uses custom authentication settings, which may be configured as follows:
  - **Single LDAP/AD domain, with no failover protection:** The Kubernetes cluster uses a single LDAP/AD domain with a single server to authenticate users. There is no failover protection.
  - **Single LDAP/AD domain, with failover protection:** The Kubernetes cluster uses a single LDAP/AD domain with two or more servers configured, which provides failover protection in case a server goes down or becomes unreachable.
  - **Multiple LDAP/AD domains:** The Kubernetes cluster uses two or more LDAP/AD domains to authenticate users. Each domain may be configured to use either one server (no failover protection) or multiple servers (for failover protection).

Locating Authentication Settings

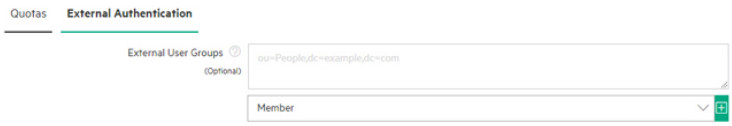
Configuring user authentication options takes place in one or more of the following locations in the web interface, depending on the configuration:

- The **User Authentication** screen (see The User Authentication Screen) allows the Platform Administrator to specify the following settings:
  - Whether or not multiple LDAP/AD domains are supported.
  - If multiple LDAP/AD domains are supported, whether or not the **Domain for Authentication** pull-down menu will appear on the **Login** screen.
- If LDAP/AD is enabled, then the **External Authentication** tab will appear in the Kubernetes **Edit Tenant** screen (see Editing an Existing Kubernetes Tenant or Project on page 454). Any changes you make here will only apply to the current tenant/project. See External Authentication Tab on page 779.
- In the **Step 3: Authentication** screen when creating a new Kubernetes cluster. See Creating a New Kubernetes Cluster on page 463 and Step 3: Authentication Screen (Kubernetes Clusters) on page 779.

See User Authentication for additional information on user authentication.

External Authentication Tab

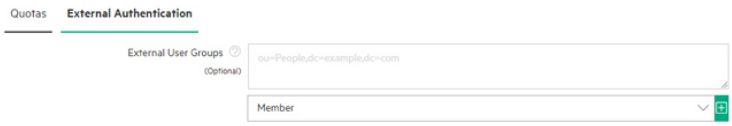
The **External Authentication** tab of the **Create Tenant** or **Edit Tenant** screen enables you to configure the user authentication options for the current tenant/project when HPE Ezmeral Runtime Enterprise uses platform-wide LDAP/AD user authentication (see Tenant Groups on page 779).



NOTE: This tab does not appear when platform-wide local authentication is configured.

Tenant Groups

The **External Authentication** tab of the **Create Tenant** or **Edit Tenant** screen enables you to specify the LDAP/AD groups that can access the tenant or project, if any.



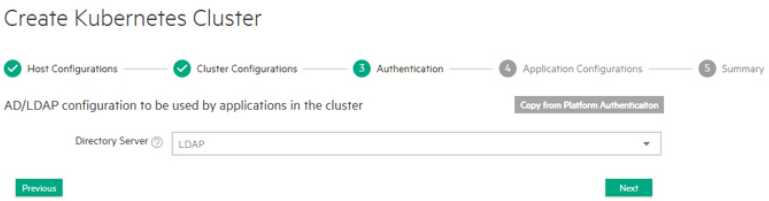
To assign one or more groups to a tenant, enter the group information in the **External User Groups** field, and then use the menu to select **Member** (if members of the group should have Member access to the tenant/project), or **Admin** (if members of the group should have Administrator access to the tenant/project). Each LDAP/AD group may have one tenant/project role. If needed, you may:

- Click the **Add Group** icon (plus sign) to add another LDAP/AD group.
- Click the **Remove Group** icon (minus sign) to remove an LDAP/AD group.
- To remove all LDAP/AD groups, click the **Remove Group** icon (if applicable), and then highlight the final remaining group and press [DEL].

When you have finished specifying group settings, continue creating or editing the tenant or project.

Step 3: Authentication Screen (Kubernetes Clusters)

The **Step 3: Authentication** screen allows the Kubernetes Administrator to specify whether the new Kubernetes cluster will use the same user authentication process configured for the HPE Ezmeral Runtime Enterprise deployment, or whether a different user authentication process will be configured for this particular Kubernetes cluster.



On this tab:

- Clicking **Next** applies the platform-wide user authentication configuration to the Kubernetes cluster.
- Clicking the **Copy from Platform Authentication** button copies the platform-wide authentication settings to this screen and allows you to edit these parameters as needed for this Kubernetes cluster.

- Manually entering user authentication parameters allows you to specify a completely different configuration to apply to this Kubernetes cluster.

If you are configuring user authentication options for this Kubernetes cluster, then see Configuring User Authentication Options on page 780, below.

Configuring User Authentication Options

These instructions apply as follows:

- If the deployment is configured to authenticate users on the platform level, then modifying these settings on the **User Authentication** screen (see The User Authentication Screen on page 766) will modify the user authentication settings across the entire deployment.
- If the deployment is configured to authenticate users on a per-Kubernetes-cluster basis, then modifying these settings on the **Step 3: Authentication** screen (see Creating a New Kubernetes Cluster on page 463) will modify the user authentication settings for this specific Kubernetes cluster.

To change the user authentication method:

1. If you want to use two or more authentication domains for authentication, then proceed to Step 2. Otherwise, skip to Step 8.
2. Check the **Enable Multi Domain** check box.

   A tab will appear for each domain. Each tab will be labeled either undefined (if the domain has not been configured) or with the name of the domain. Selecting a tab allows you to configure settings on a per-domain basis.

   NOTE: The configuration process described in this section is the same whether or not multiple domains are enabled. The only difference is that the tabs will appear when multiple domains are enabled, and selecting a tab will allow you to configure settings for that domain only. Modifying the settings for one domain will not affect the settings for any other domains.

3. If you want the **Login** page to display the **Domains for Authentication** pull-down menu in the **Login** screen (see Launching and Signing In on page 136), then check the **Show Domain in Login Page** check box. If this check box is blank, then:
   - The user may enter their username as `<username>@<domain>`, where `<username>` is their username and `<domain>` is the domain to use for authentication. If this box is checked and the user adds the domain when logging in, then this will override any selection they make with the **Domain for Authentication** menu.
   - The user may enter their username simply as `<username>` and HPE Ezmeral Runtime Enterprise will search for that user across all of the domains configured for the platform or Kubernetes cluster, as appropriate.
4. If desired, enter the domain regex in the **Domain Regex** field. This is a perl-like entry that extracts name and domain information from a login username. Click here for additional information (link opens an external website in a new browser tab/window). The regex entries are as follows:
   - **LDAP:** `'(?P<name>[^@]+)@?(?P<domain>[^@]*$)'`
   - **AD:** `'(((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?P<domain>.+$))|(^(?P<name>[^@\\]+)$))'`
5. Select the domain to configure, as follows:

- If you are configuring the first domain (where there is only one tab labeled undefined), then proceed to Step 7.
- If you are configuring an existing domain, then click the desired tab to begin editing that domain, and then proceed to Step 7.
- If you need to add a domain, then click the **Add Domain** icon (plus sign) on one of the existing tabs, and then proceed to Step 7.
- If you need to remove a domain, then click the **Remove Domain** icon (minus sign) on the tab that corresponds to the domain you are removing. For example, to delete the **LDAPS One** domain, click the **Remove Domain** icon on the **LDAPS One** tab.

  CAUTION: You cannot undelete a domain. If you remove a domain by accident, then you will need to reconfigure that domain, and users assigned to the removed domain will not be able to access container platform.

6. Enter a name that will be used to identify the service in the **Auth Service Identifier Name** field.
7. Select the desired type of authentication use using the **Authentication Type** pull-down menu. The available options are:
   - **Local**: Selecting this option configures the platform or Kubernetes cluster to use the built-in user database for user authentication. See The User Management Screen for information on managing the local user database. This option is not available if multiple domains were enabled in Step 2. Skip to Step 15.
   - **LDAP**: Selecting this option configures the platform, or Kubernetes cluster to use an existing external LDAP server. Proceed to Step 9.
   - **Active Directory**: Selecting this option configures the platform, or Kubernetes cluster to use an existing external Active Directory (AD) server. Proceed to Step 9.
8. Use the **Security Protocol** pull-down menu to select the security protocol that will be used to access the authentication server (**None**, **LDAPS**, or **Start TLS**). If you select **LDAPS** or **Start TLS**, ensure that the AD server supports TLS 1.3.
9. You must configure at least one authentication server to use with this domain.

   If desired, you may configure two or more servers for this domain, which will help ensure successful user logins if a server goes down or becomes unreachable. To configure the authentication servers to use with this domain:
   - All of the servers that you are configuring must meet all of the requirements listed in Server Failover Requirements on page 783.
   - Enter the hostname or IP address of the external LDAP/AD host in the left field of the **Service Locations** area.
   - Enter the port of the external LDAP/AD host in the right field of the **Service Locations** area.
   - To add another server, click one of the **Add Service Location** icons, and then add the hostname/IP address and port for that server in the appropriate fields.

- To remove a server, click the **Remove Service Location** icon (minus sign) for the server that you want to remove.
- If you check the **Reorder on Failover** check box, then the deployment will reorganize the list of configured servers to always try the last server that responded first. For example, assume four servers set up in the order A, B, C, D with this option enabled. Server A does not respond in a timely fashion, but B does. The server list will now appear in the order B, A, C, D. If A, B, and C do not respond but D does, then the new order will be D, B, A, C. Essentially, the first server to respond will be moved to the front of the list. Clearing this check box forces the deployment to check each server in the order in which it appears in this list. In this example, authentication will always proceed in the order A, B, C, D.

10. The binding type determines how the entered username is translated into a string that is understood by the LDAP/AD server. Select how the LDAP/AD user will be determined using the **Bind Type** pull-down menu. The available options are:
    - **Direct Bind**: This option derives the user's LDAP/AD name, also known as a user's distinguished name (DN), from the entered username and then attempts to authenticate the user using the entered password.
    - **Search Bind**: This option establishes a connection to the LDAP/AD server either anonymously or using a fixed account, searches for the authenticating user's DN, and then attempts to authenticate the user using the entered password.
    - Enter the LDAP/AD attribute used to retrieve user profiles in the **User Attribute** field. This will typically be `cn` for LDAP servers or `sAMAccountName` for AD servers.

    See Direct Bind (LDAP) on page 783, Direct Bind (AD) on page 784, or Search Bind (LDAP/AD) on page 785 for configuration instructions based on your selected server/binding options. Complete the appropriate configuration before proceeding to Step 12.

11. If you selected **LDAPS** or **Start TLS** in Step 9, then checking the **TLS Verify Peer** check box instructs the deployment to verify that the certificate of the LDAP/AD server has been signed by a known Certificate Authority (CA). When this option is selected,the entire user authentication certificate chain will be verified, and all applicable Certificate Revocation Lists (CRLs) will be scanned. If any portion of the certificate chain has been revoked at any level, then the affected LDAP/AD servers will not be queried, affected users will not be able to log in via any affected LDAP/AD servers and no certificates will be transmitted or received certificates. If you are copying global authentication when this option is enabled, you will need to re-upload the CA certificate.

12. Verify the authentication settings by clicking the **Verify** button at the top of the **User Authentication** tab or **External Authentication** tab to open the **Verify Authentication Settings** popup, entering a test username and password in the appropriate fields, selecting the domain to verify using the **Domain for Authentication** pull-down menu if multiple domains are configured, and then clicking **Submit**. A green bar with the message `User authorized successfully` appears if the configuration is correct. If the selected domain has multiple servers configured, then the verification process will stop as soon as one of those servers sends a response.

13. If you have **Search Bind** selected for LDAP or AD and want to enable SAML SSO, then check the **Enable SAML SSO** check box. See SSO on page 786 for configuration instructions. Otherwise, proceed to Step 14.

14. When you have finished configuring your authorization options, click **Submit** to save your changes. Your settings will be automatically verified.

Server Failover Requirements

The servers that are being used for failover protection in a single LDAP/AD domain must meet all of the following requirements:

- All servers must be online.
- All servers must be reachable from HPE Ezmeral Runtime Enterprise.
- All server certificates must be issued by the same Root Certificate Authority (CA).
- All servers must use the same security protocol (**LDAPS**, **Start TLS**, or **None**).
- All servers must either use the same user to bind or be anonymous.
- All servers must have the same search base.
- All servers must have the same username element (cn or sAMAccount).
- All servers must use the same bind type (Search or Direct).
- If this is an Active Directory (AD) domain, then all servers must have the same NT Domain.

Return to Step 10, above.

Direct Bind (LDAP)

Direct Bind for an LDAP server will always compose the user DN at login time by combining the given user name with the specified User Attribute and specified User Subtree DN. For example suppose the User Attribute is `cn` and the User Subtree DN is `dc=mycompany,dc=com`. When a user `bob` attempts to login, the authentication to the LDAP server will be attempted with the DN `cn=bob,dc=mycompany,dc=com`. If users have DNs that differ in portions other than just the user attribute, then Direct Bind is not a usable configuration.

NOTE: Direct Bind cannot be used with SSO.

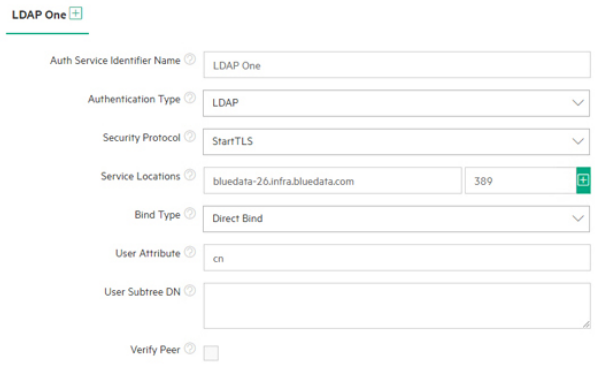
Direct Bind is only desirable if the LDAP server does not support anonymous search and there is no designated "service" user account that could be used to do searches for user objects. Search Bind is preferable for many LDAP configurations and also is preferable if LDAP will be used for container (virtual node) login authentication.

If you are using direct binding with an LDAP server, then you will need to specify the following parameters:

- Enter the LDAP attribute used to retrieve user profiles (such as **cn**) in the **User Attribute** field. Contact your LDAP Administrator for this information, if needed.
- Enter the LDAP subtree that will be used when searching for users in the **User Subtree DN** field. This is used to compose user object DNs at login time, as described above. It is also used as a "search base" that defines the scope which will be searched for user objects, when the object for an authenticated user is later fetched (using the credentials of that user) to determine the user's group memberships.

NOTE: These fields must match your existing LDAP parameters exactly. Contact your LDAP administrator for assistance.

This image shows sample direct bind LDAP settings.



Return to Step 12, above.

Direct Bind (AD)

If the **Yes** radio button for the **NT Domain Enabled** option is checked, then the **NT Domain** field must also be specified. In this configuration, the user DN will be formed from the username and the specified domain, as `username@domain`. If the **No** radio button is checked for **NT Domain Enabled**, then the user DN will be composed by combining the given user name with the specified **User Attribute** and specified **User Subtree DN**, in the same manner as described for Direct Bind (LDAP) on page 783, above.

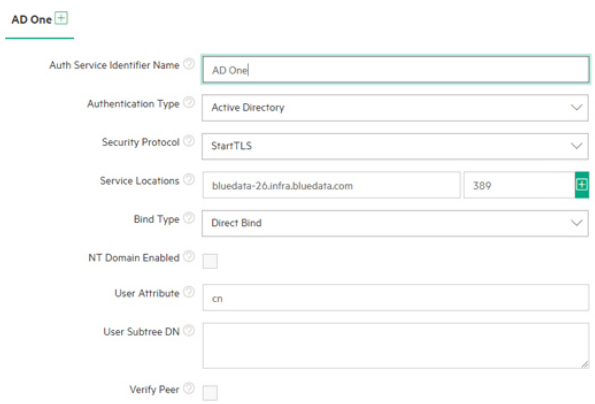
NOTE: Direct Bind cannot be used with SSO.

Direct Bind is only desirable if the AD server does not support anonymous search and there is no designated "service" user account that could be used to do searches for user objects. Search Bind is preferable for many AD configurations and also is preferable if AD will be used for container (virtual node) login authentication. Also, if **NT Domain Enabled** is not selected and users have DNs that differ in portions other than just the user attribute, then Direct Bind is not a usable configuration.

Regardless of whether **NT Domain Enabled** is enabled, you will need to specify the following additional parameters to use direct binding with an AD server:

- Enter the Active Directory attribute used to retrieve user profiles (such as **sAMAccountName**) in the **User Attribute field**. Contact your AD Administrator for this information, if needed.
- Enter the Active Directory subtree that will be used when searching for users in the **User Subtree DN** field. This is used to compose user object DNs at login time if **NT Domain Enabled** is not selected. It is also used as a "search base" that defines the scope which will be searched for user objects, when the object for an authenticated user is later fetched (using the credentials of that user) to determine the user's group memberships.

This image shows sample direct bind AD settings:



Return to Step 12, above.

Search Bind (LDAP/AD)

The Search Bind configuration for AD/LDAP integration does not make assumptions about the form of user DNs; however, this does require searching for a user object to find its DN, before that particular user's credentials can be authenticated. Therefore, the AD/LDAP server must either support anonymous searches of the directory scope that contains the user objects, or there must exist general service account credentials that can be used to bind to the server and search for users. Search Bind provides the most flexibility in the arrangement of user objects that can be supported. Search Bind is also necessary for container (virtual node) login authentication, unless the AD/LDAP server supports anonymous searches.

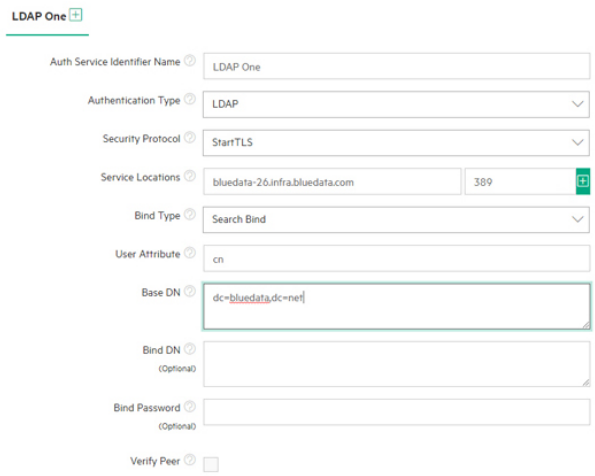
If you are using search binding with either LDAP or AD, then you will need to specify the following parameters:

- **User Attribute:** Enter the LDAP/AD attribute used to retrieve user profiles (such as **cn** for LDAP or **sAMAccountName** for AD) in this field.
- **Base DN:** Subtree in the LDAP/AD hierarchy within which to search for users.

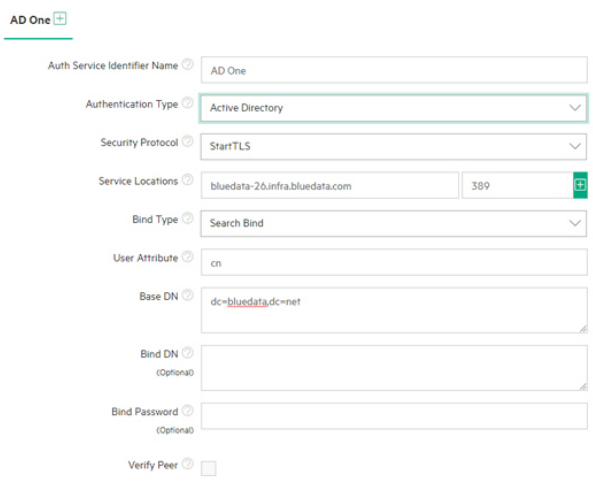
The following parameters are either optional (if the LDAP/AD server supports anonymous searches) or required (if the LDAP/AD server does not support anonymous searches):

- **Bind DN:** If the server does not allow anonymous binds, then enter the DN to bind to inside LDAP/AD to obtain permission to search for users.
- **Bind Password:** If the Bind DN requires a password, then enter that password (case sensitive) in this field.

This image shows sample search bind LDAP settings.



This image shows sample search bind AD settings.



Return to Step 12, above.

SSO

Single Sign On (SSO) allows a user to enter their credentials once (such as when arriving at the office in the morning), and then access all authorized resources without having to enter their credentials every time. If all of the following conditions are met, then you can configure the platform or Kubernetes cluster to allow users to log in to the interface without needing to enter their username and password:

- Hosts are running on RHEL/CentOS 7.x.

  NOTE: SLES SUSE operating system is not supported.

- Either:
  - The platform or Kubernetes cluster has been configured to use Search Bind with either LDAP or AD, as described in Search Bind (LDAP/AD) on page 785.
  - Your organization has enabled SSO based on the Security Assertion Markup Language (SAML) version 2.0 or later.

When the platform or Kubernetes cluster is configured to use SSO, an authorized user who accesses the web interface IP address will bypass the **Login** screen and go directly to the **Dashboard** screen for the tenant or project they most recently accessed. The SSO login process does forward the user to the Identity Provider (IdP) before forwarding them to the deployment. If the user already has an active IdP session, then they will see the appropriate **Dashboard** screen. If not, then they will see an IdP login screen and will be forwarded to the **Dashboard** screen after they provide their credentials. Users will access the web interface by navigating to either the hostname or IP address of one of the following:

- Controller host, if platform HA is not enabled.
- Cluster, if platform HA is enabled (see High Availability on page 132).

To allow SSO access:

1. Configure the Identity Provider (IdP) to allow SSO access: See either Configuring the Identity Provider on page 786.
2. Configure the deployment to allow SSO access by authorized users: See SSO on page 786, below.

Configuring the Identity Provider

1. Provide the following information to your IdP:

- **Audience:** This field is not required; however, providing the base URL of the SAML server is more secure than a blank entry. If you do enter a URL, then this URL must exactly match the **SAML Application Name** that you will specify in the deployment.
- **Recipient:** Enter `<name-or-ip>/bdswebui/login`, where:
  - If platform HA is not enabled, `<name-or-ip>` is either the hostname (FDQN) or the IP address of the Controller host, if platform HA is not enabled.
  - If platform HA is enabled, `<name-or-ip>` is either the cluster hostname (FQDN) or IP address.

    For HPE Ezmeral Runtime Enterprise 5.3.5 and later releases, to use SAML SSO with Jupyterhub Notebooks, you must specify the Controller gateway FDQN for `<name-or-ip>`. Do not specify an IP address.
- **Consumer URL Validator:** Enter `<name-or-ip>/bdswebui/login/`, where `<name-or-ip>` is one of the following:
  - `.*` - This is a valid generic entry, but is less secure. For example, `.*/bdswebui/login/`.
  - Either the FQDN or IP address of the Controller host, Controller gateway, or cluster, as described in **Recipient**. This entry is more secure than the generic entry. For example, `10.32.0.75/bdswebui/login/` or `MyPlatform-01.organization.com/beswebui/login/`.
- **Consumer URL:** Enter `<name-or-ip>/bdswebui/saml_login/`, where `<name-or-ip>` is either a generic or specific entry, as described above. This may also be described by your IdP as the **Single Sign On URL**, the **SAML Assertion Consumer Service URL**, or the **ACS URL**.
- **SAML domain:** If users must be authenticated against a specified domain, then you must configure the IdP to send the domain in the SAML Assertion. A SAML Assertion is an XML document that can contain arbitrary data. HPE Ezmeral Runtime Enterprise can use that arbitrary data for group assignment. However, because these "groups" and "roles" come over in the assertion, HPE Ezmeral Runtime Enterprise cannot guarantee them beforehand.

NOTE: Your IdP may use different labels for these parameters. Contact them for assistance, if required.

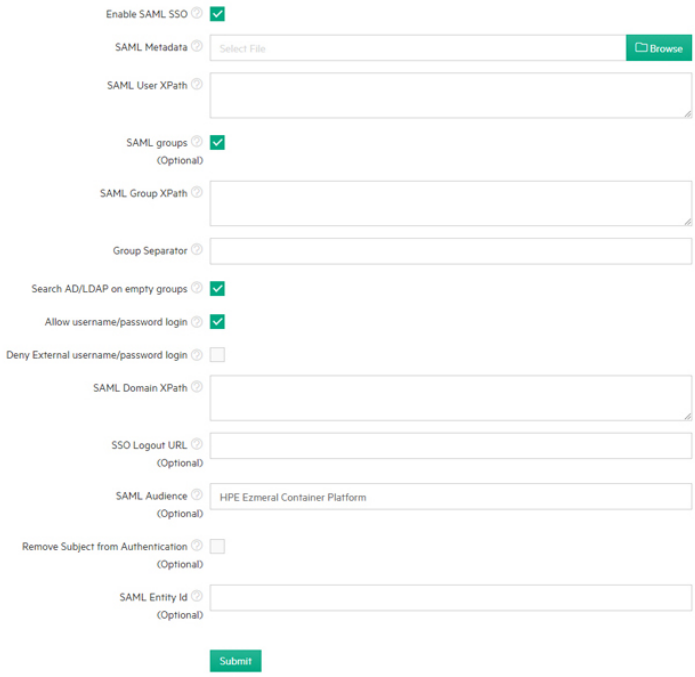
2. Your IdP will provide a SAML IdP XML metadata file. You will use this file when configuring HPE Ezmeral Runtime Enterprise for SSO, as described below.
3. Configure the deployment for SSO, as described in SSO on page 786.

Configuring HPE Ezmeral Runtime Enterprise for SSO

To configure the deployment for SSO:

1. Configure your IdP as described in Configuring the Identity Provider on page 786, above.
2. In the web interface, configure either LDAP or AD for Search Bind, and then check the **Enable SAML SSO** check box.

   The **User Authentication** tab or **External Authentication** tab expands to display the SSO options.



3. In the **SAML Metadata** field, either enter the complete path to the SAML IdP XML metadata file that you obtained from your IdP, or click the **Browse** button to open a standard **File Upload** popup that allows you to navigate to and select the file.

4. Enter the SAML user XPath in the **SAML User XPath** field. This path will have a format that may look like `//saml:Subject/saml:NameID/text()`.

5. If desired, enter an XPath in the **SAML Group XPath** field. This XPath points to a text-style field in the SAML Assertion that contains a single group or a list of groups separated by the string defined in the **Group Separator** field.

6. If desired, enter a list of characters in the **Group Separator** field that will separate group names in the SAML Authenticator. This should always be seen as a list of separators, and any separator in this list will never appear in the extracted group names.

7. Check the **Search AD/LDAP on empty groups** check box to have HPE Ezmeral Runtime Enterprise search the configured LDAP or AD server to determine user privileges if the user has already been authenticated via SAML and if that user did not have any groups in the SAML Assertion.

8. If you want to allow users to log in directly without the need to go through the SSO process, then check the **Allow username/password login** check box. This is a safety feature that ensures access if the SSO server goes down. It also allows you to log in if you make a mistake when configuring the platform or tenant/project for SSO. When this feature is enabled, you may access the web interface by navigating to `<controller_name-or-ip>/bdswebui/login?local`, and then entering either your local or LDAP/AD username and password.

   CAUTION: Hewlett Packard Enterprise strongly recommends allowing username and password logins, especially when performing the initial sso configuration. Failure to enable this option may result in all users being locked out of hpe ezmeral container platform. Only disable this option once you have verified that sso is properly configured and working properly.

9. If you want SAML to be the only method by which LDAP/AD users can log in, then check the **Deny External username/password login** check box. This is only recommended if users have been exclusively using SAML to authenticate to HPE Ezmeral Runtime Enterprise for a significant period of time.

10. If desired, define where to find the domain in the SAML Assertion using in the **SAML Domain XPath** field. For example, if the domain is in an **Attribute** field where the **Name** attributed is defined as `Domain`, then the XPath would look like this: `//saml:AttributeStatement/saml:Attribute[@Name="Domain"]/saml:AttributeValue/text()`

11. If desired, you may enter the complete URL (including the `http://` or `https://` prefix) to where a user will be directed when they log out in the **SSO Logout URL** field. If this field is left blank, then logged-out users will be redirected to a SAML-specific logout page.

12. If you provided the base URL of the SAML server in the IdP **Audience** field, then enter that exact URL in the **SAML Audience** field. If this field is left blank, then the Audience portion of the SAML assertion will not be validated.

13. If instructed by Hewlett Packard Enterprise Technical Support, then check the **Remove Subject from Authn** check box. Do not enable this option unless instructed to do so.

14. If needed, enter the entity ID to be used in authentication requests in the **SAML Entity Id** field. Leaving this field blank will use the machine (Controller host) name.

15. Return to Step 14, above.

Accessing LDAP/AD/SAML Logs

All queries sent to the configured LDAP/AD server can be logged. To enable this feature:

1. SSH into the Controller host.

2. Execute either of the following commands, as appropriate:

   - **LDAP/AD:**`/opt/bluedata/common-install/bd_mgmt/bin/bd_mgmt enable_management_logger authaudit`

   - **SAML:**`/opt/bluedata/common-install/bd_mgmt/bin/bd_mgmt enable_management_logger samldebug`

3. To locate LDAP/AD queries, search `/var/log/bluedata/bds-mgmt.log` for all instances of `authaud` (LDAP/AD) or `samldeb` (SAML).

This feature logs the following LDAP/AD activity:

- An LDAP/AD user who does not have an existing session attempts to log in.
- A Platform Administrator changes the LDAP/AD authentication parameters.
- A Platform Administrator verifies the LDAP/AD authentication parameters.
- A user is added who is being authenticated by an external LDAP/AD server.

This feature does not log the following activity because it does not require querying the LDAP/AD server:

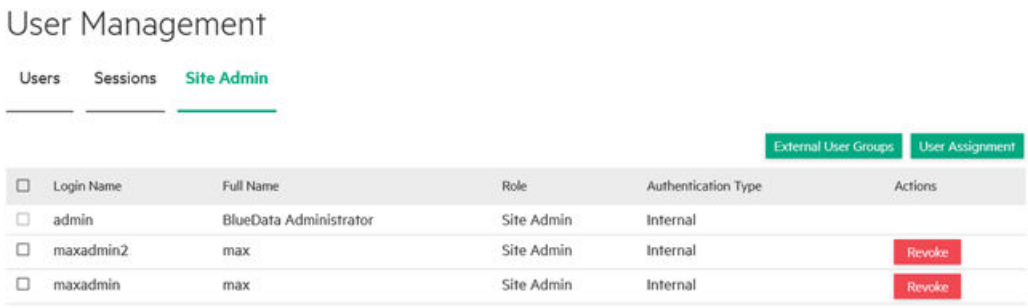
- An authenticated user attempts to log in.
- A user is added who is not being authenticated by an external LDAP/AD server.
- A user is assigned a role within a tenant or project.

Managing Platform Administrators

The topics in this section describe the settings and tasks related to the managing Platform Administrator (Site Admin) users in HPE Ezmeral Runtime Enterprise.

If the deployment is configured for LDAP/AD, then the **External Groups** button appears on the **Site Admin** tab of the **User Management** screen. Clicking this button opens the **Update Site Admin's User Groups** dialog, which enables you to specify LDAP/AD user groups that will be assigned the `Site Admin` role.

The `Site Admin` role has Platform Administrator rights.

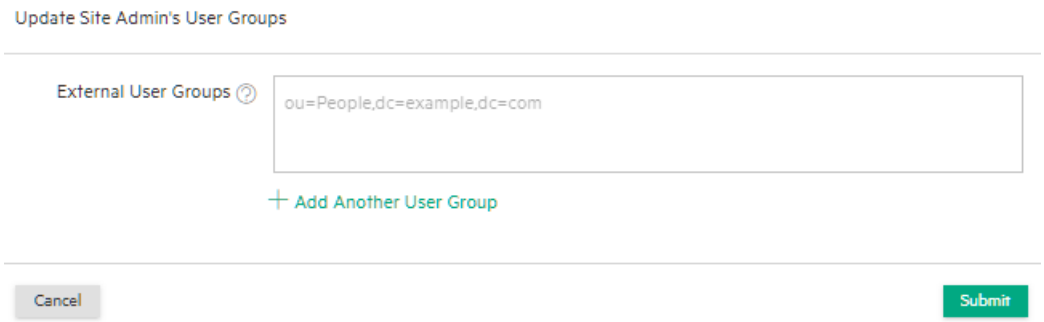


Updating Platform Administrator Groups

To configure the LDAP/AD groups that will be given Platform Administrator rights (assigned the `Site Admin` role):

1. In the **User Management** screen, select **Site Admin**.
2. Click **External User Groups**.

   The **Update Site Admin User's Groups** dialog appears.



3. Enter the first group to associate with the tenant in the field that appears, as shown in the example above.
4. To add another group, click the **Add Group** icon (plus sign) to the right of the field.
5. To remove a group, click the **Remove Group** icon (minus sign) to the right of the group you want to remove.

When you have finished making your desired changes, click the **Submit** button to close the dialog and return to the **User Management** screen.

The exact DN of the group in the LDAP or AD server will be confirmed, and that DN will be used to perform group membership checks on users.

last-updated: May 06, 2024

The User Management Screen

NOTE: See User Authentication on page 126, Assigning/Revoking User Roles (Local) on page 771, and Assigning/Revoking User Roles (LDAP/AD/SAML) on page 774 for additional information on user authentication.

If you are a Platform Administrator and are in the **Site Admin** tenant, then selecting **Users** in the main menu opens the **User Management** screen. The following tabs are available in this screen:

- **Users**: This tab displays all of the users in the deployment. See Users Tab on page 791.
- **Sessions**: This tab displays the users who are currently logged in. See Sessions Tab on page 792.
- **Site Admin**: This tab displays the users that have Platform Administrator rights (the Site Admin role). See Site Admin Tab on page 793.

NOTE: If you are a Tenant or Project Administrator, then selecting **Users** in the main menu will open the **Tenant Details** screen. See Viewing User Assignments on page 770.

Users Tab

The **Users** tab displays users and their current assignments and authentication types.

The top of this tab contains the following buttons:

- **Add User**: Clicking the **Add User** button opens the **Add New User** screen.
-
  - If the deployment is configured for local authentication (see The User Authentication Screen on page 766), then see Creating a New User (Local) on page 776.
  - If the deployment is configured for LDAP/AD (see The User Authentication Screen on page 766), then see Assigning/Revoking User Roles (LDAP/AD/SAML) on page 774).
- **Delete:** Deletes the selected users. See Deleting a User on page 777.

The table on this tab contains the following information/functions for each user:

- **Login Name:** Login name of the user.
- **Full Name:** Full name of the user.
- Assignment information, which will be either:
  - **Assigned Tenants:** Number of tenants/projects in which the user has a role, if the deployment is configured to use platform authentication.
  - **Assigned Tenant:** Name of the tenant/project to which the user is assigned, if the deployment is configured to use tenant independent authentication.

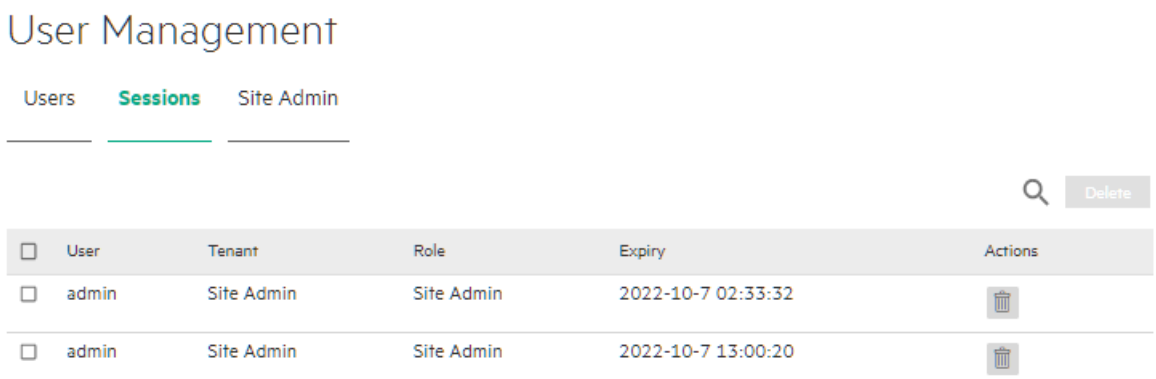
- **Authentication Type:** Type of authentication used when the user logs in. This will be either **Internal** (if you are using the internal user database to handle user authentication) or **External** (if the user is being authenticated using LDAP or Active Directory).
- **Actions:** The following actions are available for each user:
  - **Details:** Clicking the **Details** icon (bulleted list) opens the **User Details** screen for that user. See The User Details Screen on page 793.
  - **Delete:** Clicking the **Delete** icon (trash can) deletes the selected user. See Deleting a User on page 777.
  - **Reset Password:** Clicking the **Reset Password** icon (circular arrow) for a user opens the **Reset User Password** popup for the selected user. Enter and confirm the new password in the **New Password** and **Confirm Password** fields, and then click **Submit** to save your changes and close the popup.

CAUTION: You cannot undelete a user. Deleting a user removes all roles.

NOTE: Deleting a user only removes them from the user database. If you are using an external authentication server, then you will need to remove or disable the user's account on the authentication server.

Sessions Tab

The **Sessions** tab of the **User Management** screen displays all of the currently active user sessions (logins).



The top of this tab contains the following button:

- **Delete:** Selecting one more sessions in the table and then clicking this button deletes the selected sessions. The affected users will have to log back in with their username and password. Jobs, data, etc. are preserved.

The table on this tab contains the following information/functions for every active session:

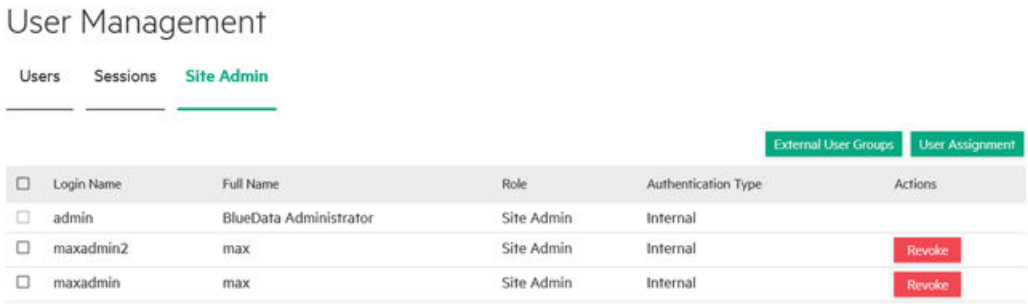
- **User:** Name of the user running the session.
- **Tenant:** Tenant or project the user is using for this session. A user with access to multiple tenants/projects may run more than one session.
- **Role:** Role of the user (**Site Admin**, **Admin**, or **Member**).
- **Expiry:** Date and time the current session will expire if the user takes no actions. By default, the deployment allows up to 24 hours of inactivity per session, to allow running jobs to complete.

- **Delete:** Clicking the **Delete** icon (trash can) in the **Actions** column deletes the selected session. A popup warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without deleting the sessions. The affected user will have to log back in with their username and password. Jobs, data, etc. are preserved. See Managing User Sessions on page 777.

Site Admin Tab

The Site Admin tab enables you to manage the Platform Administrator users.

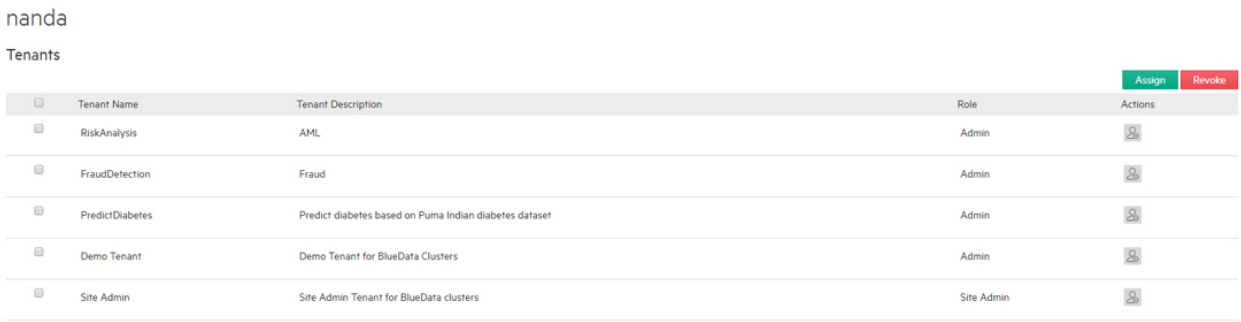
Platform Administrators have the role: `Site Admin`



The User Details Screen

If you are a Platform Administrator, then clicking the **Details** icon for a user name in the **Users** table on the **User Management** screen opens the **User Details** screen for the selected user.

The **User Details** screen appears as shown here when the platform is configured to authenticate users on the platform level:



This screen contains the following buttons:

- **Assign:** Clicking this button opens the **Assign Users** screen. See Assigning/Revoking User Roles (Local).
- **Revoke:** Clicking this button revokes the selected users access to the tenant or project. A popup warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without revoking the user's role for the tenant or project.

NOTE: If you revoke a user by mistake, you can reassign them to the tenant or project using the **Assign Users** screen. See Assigning/Revoking User Roles (Local).

The table on this screen contains the following information and functions:

- **Tenant Name:** Name of each tenant/project the user is currently assigned to. Each user may have one role per tenant or project.
- **Tenant Description:** Brief description of each tenant/project.

- **Role:** Role the user has within that tenant/project.
- **Revoke:** Clicking the **Revoke** icon (person) in the **Actions** column revokes the selected user's access to the tenant/project. A warning appears asking you to confirm or cancel the action. Click **OK** to proceed, or **Cancel** to exit without revoking the user's role for the tenant or project.

Authentication Groups

When configured for platform-wide LDAP/AD user authentication (see The User Management Screen on page 791 and Configuring User Authentication Settings on page 778) the addition of LDAP/AD groups (called *authentication groups*) to a tenant or project as Tenant/Project Members, Tenant/Project Administrators, or Platform Administrators via the **External Authentication** tab is supported, as described in Configuring User Authentication Settings on page 778.

This feature lets you avoid having to manually add individual users. Each authentication group may be associated with up to one role per tenant or project. A pop-up error dialog appears if you try to assign multiple roles to the same authentication group within a single tenant.

An LDAP/AD user who belongs to one of a tenant's or project's authentication groups, as declared by the `memberOf` or `isMemberOf` attribute in that user object, can log in and act within that tenant/project.

NOTE: The `isMemberOf` variant attribute is currently only supported for the purposes of UI/API login and tenant/project role assignment. The default authentication package used in the container login feature still requires `memberOf` as the group pointer. If `isMemberOf` needs to be recognized for container login purposes, then the authentication package will need to be modified..

Such a user is treated as follows:

- A user who is a member of at least one tenant authentication group can log into a tenant/project using their LDAP/AD credentials.
- A user who is authenticated because of group membership will have their role in a tenant (i.e. member or admin) determined by the role associated with that group.
- A user who is a member of multiple authentication groups for a tenant or project will have the Tenant Administrator role in that tenant if any of those groups are associated with the Tenant Administrator role.
- User privileges persist for the duration of a session. A session lasts until the user logs out, 24 hours pass, or until a Platform Administrator terminates the session as described in Managing User Sessions on page 777, whichever comes first.
- Changes to tenant authentication groups and role associations, or changes to group memberships on the LDAP/AD server, will apply to affected users the next time they log in and establish a new session.

NOTE: Nested group membership is not supported. For example, if Group_A is the only authentication group specified for a tenant/project and Group_B is a member of Group_A, then only users who are members of Group_A will be authenticated. Users who are members of Group_B but who are not direct members of Group_A will not be authenticated.

NOTE: When using an Active Directory server for authentication, an authentication group will not be able to grant access for AD users that have it as their Primary Group. Only the non-primary groups assigned to AD users can be employed as authentication groups. This issue is not a concern if you are using an LDAP server.

The user account for a group-authenticated user is created whenever that user logs in. This behavior has the following implications:

- Login-time account creation for a user will not occur if the Platform Administrator has manually added that user as an externally-authenticated LDAP/AD user. In that case, the user's manually assigned tenant/project roles will take precedence over the effects of any authentication group memberships.

- The Platform Administrator cannot modify the roles assigned to users who belong to an authentication group but who have not been manually added. These changes must happen at the LDAP/AD server level.
- Users who belong to an authentication group will not appear in the **User Management** screen until they log into HPE Ezmeral Runtime Enterprise or a specific tenant/project for the first time.
- Removing an authentication group user from the **User Management** screen does not override their group-based access permissions, because the affected user will simply be able to log back in and re-create their user account.

  Changing such a user's access privileges requires either removing them from the authentication group at the LDAP/AD server or changing the role associated with the entire authentication group (see Editing an Existing Kubernetes Tenant or Project on page 454).

The System Settings Screen

The topics in this section describe the System Settings screen and its tabs in HPE Ezmeral Runtime Enterprise.

Selecting **Settings** in the main menu opens the **System Settings** screen. The top of this screen contains a series of tabs:

- **Tenant Storage:** This tab allows the Platform Administrator to specify the root directories for automatically-created tenant DataTaps. See Tenant Storage Tab.
- **License:** This tab allows the Platform Administrator to manage licensing. See License Tab.
- **Air Gap:** This tab allows the Platform Administrator to manage air-gap settings for Kubernetes. See Air Gap Tab.
- **Updates**: This tab enables the Platform Administrator to manage available HPE Ezmeral Runtime Enterprise software updates and Kubernetes Bundle updates to the deployment. See Updates Tab on page 801.
- **Other:** This tab allows the Platform Administrator to manage various miscellaneous settings. See Other Tab.

Tenant Storage Tab

The Tenant Storage tab enables the Platform Administrator to designate a storage service (and, optionally, a path below the root directory) for use as tenant storage.

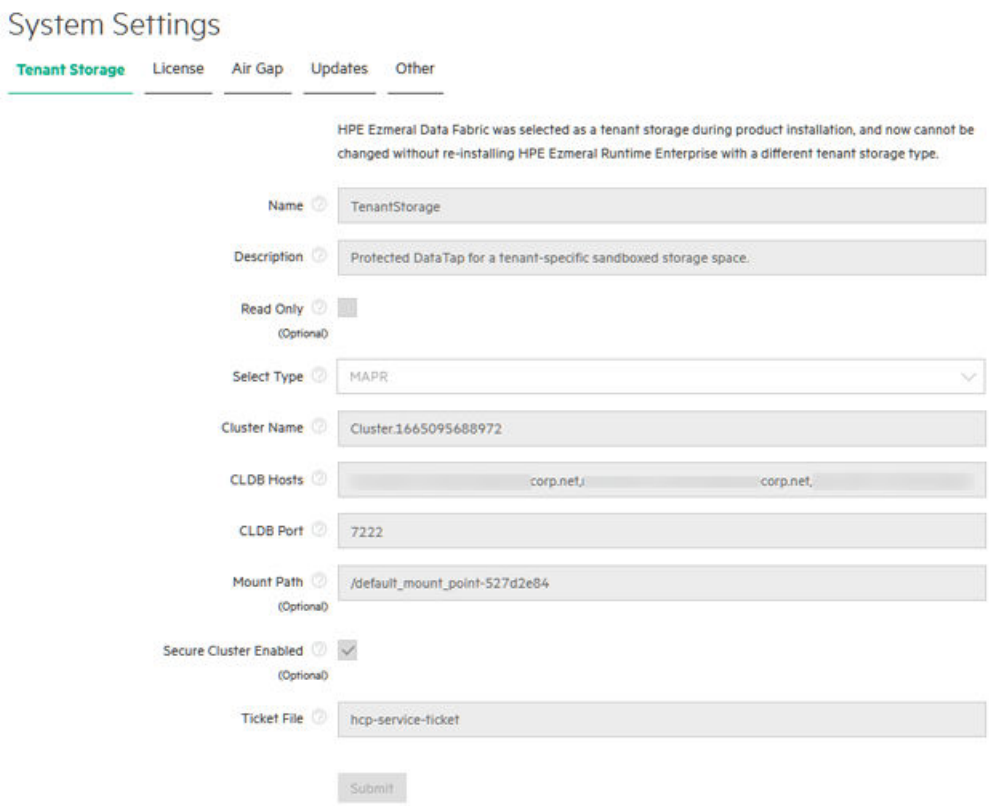
A unique subdirectory is created for each new tenant in the tenant storage when that tenant is created, as is a special DataTap pointing to that subdirectory. The properties of this DataTap cannot be edited, and the DataTap cannot be deleted until the tenant is deleted. The nodes in a tenant may not access the Tenant Storage service outside of this subdirectory.

If Tenant Storage is placed on local HDFS, then you may also assign a quota to the tenant to restrict how much data can be stored under this subdirectory.

Changing the tenant storage settings will affect tenants created after the change is made, but will not affect existing tenants. Once a tenant storage DataTap is created, it is never modified.

NOTE: If no tenant storage is added, then this field will remain unpopulated and no Tenant Storage volumes will be available for tenants until tenant storage is configured. Tenant storage helps organize information and is generally useful; the following page discusses the benefits of setting up tenant storage: Tenant/Project Storage on page 121.

The **Tenant Storage** tab of the **System Settings** screen (see The System Settings Screen on page 795) enables the Platform Administrator to designate a storage service (and, optionally, a path below the root directory) for use as tenant storage.



To change the tenant storage settings:

1. In the **Name** field, enter a name to be used when creating the special tenant storage DataTaps.
2. In the **Description** field, enter the description to display for those DataTaps.
3. If you want to be able to read but not write to the tenant storage, then check the **Read Only** check box. This only applies to access from within the virtual nodes in the tenant. You can still upload files using the DataTap browser or other external means.
4. Select the file system type to use (MAPR, HDFS, or NFS). If there are no existing tenants, then you may also select **None** to remove any existing tenant storage.

   HPE Ezmeral Runtime Enterprise Essentials supports NFS only.
5. Enter the cluster name, CLDB hosts, CLDB port, mount path, and ticket in the appropriate fields. You may also check the **HPE Ezmeral Data Fabric Secure** check box to enable HPE Ezmeral Data Fabric security.
6. If desired, enter the username that will be used to access the HDFS in the **Username** field.
7. Click **Submit** to make your changes.

MAPR Parameters

If you selected **MAPR** in Step 4, above, then enter the following parameters::

- **Cluster Name:** Name of the MapR cluster. See the MapR articles Creating the Cluster and Creating a Volume articles.

- **CLDB Hosts:** DNS name or address of the service providing access to the storage resource. For example, this could be the namenode of a MapR cluster. See the MapR article Viewing CLDB Information.
- **Port:** Port for the namenode server on the host used to access the MapR file system. See the MapR article Specifying Ports.
- **Mount Path:** Complete path to the directory containing the data within the specified MapR file system. You can leave this field blank if you intend the Data Source to point at the root of the specified share/volume/file system. See the MapR articles Viewing Volume Details and Creating a Volume.
- **MapR Secure:** Checking this check box enables the MapR Secure feature. MapR includes both the MapR Data Platform and MEP components, and is secure out-of-the-box on all new installations. All network connections require authentication, and all moving data is protected with wire-level encryption. MapR allows applying direct security protection for data as it comes into and out of the platform without requiring an external security manager server or a particular security plug-in for each ecosystem component. The security semantics are applied automatically on data being retrieved or stored by any ecosystem component, application, or users. See the MapR article Security.
- **Ticket:** Enter the complete path to the MapR ticket. MapR uses tickets for authentication. Tickets contain keys that are used to authenticate users and MapR servers. In addition, certificates are used to implement server authentication. Every user who wants to access a cluster must have a MapR user ticket (maprticket_<uid>), and every node in the cluster must have a MapR server ticket (maprserverticket). Tickets are encrypted to protect their contents. See the MapR articles Tickets and How Tickets Work.
- **Ticket Type:** Select the ticket type. This will be one of the following:
  - **User:** Grants access to individual users with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service**: Accesses services running on client nodes with no impersonation support. The ticket UID is used as the identity of the entity using this ticket.
  - **Service (with impersonation):** Accesses services running on client nodes to run jobs on behalf of any user. The ticket cannot be used to impersonate the `root` or `mapr` users.
  - **Tenant:** Allows tenant users to access tenant volumes in a multi-tenant environment. The ticket can impersonate any user.
- **Ticket User:** Username to be used by the ticket for authentication.
- **MapR Tenant Volume:** Volume to be accessed by the Data Source. See the MapR article Enabling and Restricting Access to Tenant Volume and Data.
- **Enable Impersonation:** Enable user impersonation.

Continue from Step 5, above, after entering the MAPR parameters.

HDFS Parameters

If you selected **HDFS** in Step 4, above, then enter the following parameters:

- **Host:** Enter either the hostname or IP address of the HDFS NameNode in the **Host** field.
- **Standby NameNode Host:** Enter the hostname or IP address of the HDFS standby NameNode, if any, in the **Standby NameNode Host** field.

- **Port:** Enter the NameNode port number in the **Port** field. Leave blank to use the default HDFS NameNode port.
- **Path:** Enter the HDFS directory under the share to use for the Data Source in the **Path** field. You may also click the **Browse** button to open an explorer window to navigate to the desired directory. You can leave this field blank if you intend the Data Source to point the root of the specified file system.
- **Username:** If needed, you can enter a valid username for accessing the HDFS.

Continue from Step 5, above, after entering the HDFS parameters.

NFS Parameters

If you selected **NFS** in Step 4, above, then enter the following parameters:

- **Host:** Enter either the hostname or IP address of the file system host in the **Host** field.
- **Share:** Enter the name of the share in the **Share** field.
- **Path:** This field specifies where the top of the Data Source's file system is rooted. For manually created Data Sources, this field must either be empty, or it must point to an existing subdirectory of the indicated storage system. For an automatically created tenant default Data Source, then HPE Ezmeral Runtime Enterprise will automatically create the indicated subdirectory if necessary, whenever any writes are done to that Data Source. Either enter the directory under the share to use for the Data Source in the **Path** field (click the **Browse** button to open an explorer window to navigate to the desired directory, if desired), or leave this field blank to point the Data Source to point the root of the specified share.

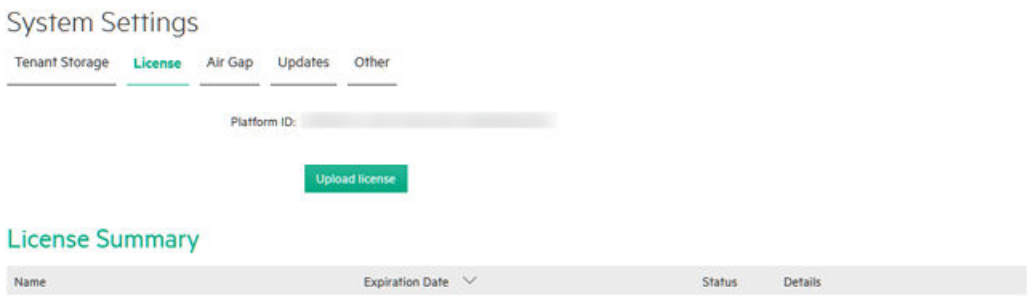
Also, be sure to configure the storage device to allow access from each host and each Controller and Worker that will using this Data Source.

Continue from Step 5, above, after entering the NFS parameters.

License Tab

The License tab enables the Platform Administrator to manage HPE Ezmeral Runtime Enterprise licenses.

The **License** tab of the **System Settings** screen (see The System Settings Screen on page 795) allows the Platform Administrator to view license information and upload a new license file (such as to change the number of allowed instances or to extend the license).



The following actions are available:

- To add a license, click **Upload license**. The **File Upload** dialog box opens. Navigate to and select a license file to upload.

  Licenses are cumulative.

  For example, if you have two licenses of the same type where one license allows 50 CPU cores and the other allows 30 CPU cores, then you will be able to use up to 80 CPU cores under that type of license.

- To delete an individual license, in the **License(s)** table, navigate to the license you want to delete and click the **Delete** icon (trash can) for that license.

- To delete multiple licenses, select the licenses you want to delete, and then click the **Delete** button above the **License(s)** table.

The **License Summary** table contains the following summarized license information:

- **Name:** Name of each available license type.
- **Expiration:** Expiration date for each license type:
  - **Latest Expiration:** Date on which the last license of this type will expire.
  - **Next Expiration:** Date on which the next license of this type will expire.
- **Status:** Status of each license type.
- **Details:** Total licensed and used CPU cores for each license type:
  - **Used Capacity:** How many CPU cores are being used by each license type.
  - **Total Capacity:** How many CPU cores have been licensed for each license type.

The **License(s)** table contains the following detailed information about each license file stored in this deployment of HPE Ezmeral Runtime Enterprise:

- **Name:** Name of the license file.
- **Expiration Date:** Date on which this license file expires.
- **License Key:** Unique key for the license file.
- **Details:** Contains the following information about the license:
  - **Start:** Date the license became valid.
  - **Capacity**: Number of CPU cores that can be used under this license.
  - **Feature:** Type of license. The entry in this column matches one of the types listed in the **Name** column of the **License Summary** table.
  - **Evaluation:** If true, indicates that this license is a temporary license granted for product evaluation.
  - **Device ID:** Unique device ID of the Controller host.
  - **Delete:** Clicking the **Delete** icon (trash can) for a license deletes that license file and, if the license has not expired, reduces the total number of licensed CPU cores for that license type.

Related concepts

Licensing on page 734

Air Gap Tab

The **Air Gap** tab of the **System Settings** screen enables the Platform Administrator to specify settings to be used when the Kubernetes hosts, clusters, and tenants do not have connectivity to the Internet.

If you will be using an air-gap configuration for Kubernetes objects, then you must configure air-gap settings before adding any Kubernetes hosts.

CAUTION:

Apply all air-gap settings with care. These settings do not propagate if updated after Kubernetes hosts have been installed, unless one of the following occurs:

- The Kubernetes host is rebooted.
- The version of Kubernetes running on a host is upgraded.

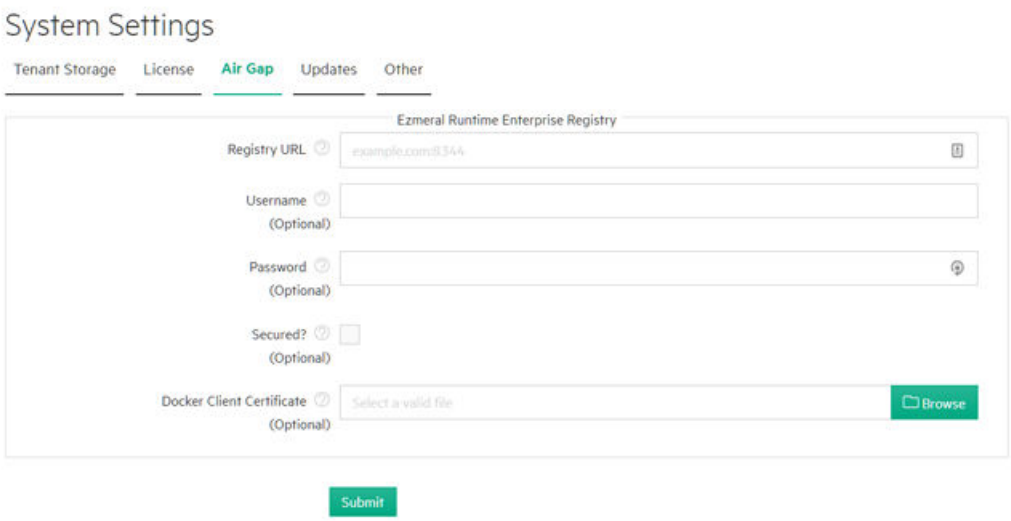
Any Kubernetes hosts in a ready state that are not part of a Kubernetes cluster must be restarted for the changes to be applied.

IMPORTANT: Changing an existing HPE Ezmeral Runtime Enterprise configuration from a non-airgapped environment to an air-gapped environment forces a reinstall of Kubernetes clusters.

If you are changing an existing HPE Ezmeral Runtime Enterprise configuration from a non-airgapped environment to an air-gapped environment, contact Hewlett Packard Enterprise support for assistance before you begin the transition. Several manual steps must be performed to transition to an air-gapped environment.

Air Gap Tab

The **Air Gap** tab of the **System Settings** screen appears as follows:



The **Ezmeral Runtime Enterprise Registry** section of this tab contains the following:

- **Registry URL:** URL to the container registry that contains the images needed for air-gap Kubernetes installations within HPE Ezmeral Runtime Enterprise.

  Ensure that you enter only hostname plus port name in **Registry URL**, For example, `test.registry.host.net:5000`. If you enter `http://` or `https://` in the URL, Kubenetes host setup fails.

  NOTE:

  HPE Ezmeral Runtime Enterprise does not support multiple container registry URLs.

- **Username:** Username to access the container registry, if needed.
- **Password:** Password to access the container registry, if needed.

- **Secured:** Checking the **Secured** check box indicates that SSL is enabled for the Kubernetes air-gap container registry. You must provide a certificate in the **Docker Client Certificate** field.
- **Docker Client Certificate:** Location of the Docker registry certificate, if you are using a secured connection.

  The certificate must use an RSA key length of 4096 bits.

  Clicking the **Browse** button opens a browser-standard **Open** dialog box that enables you to navigate to and locate the certificate to add.

Clicking **Submit** saves your changes.

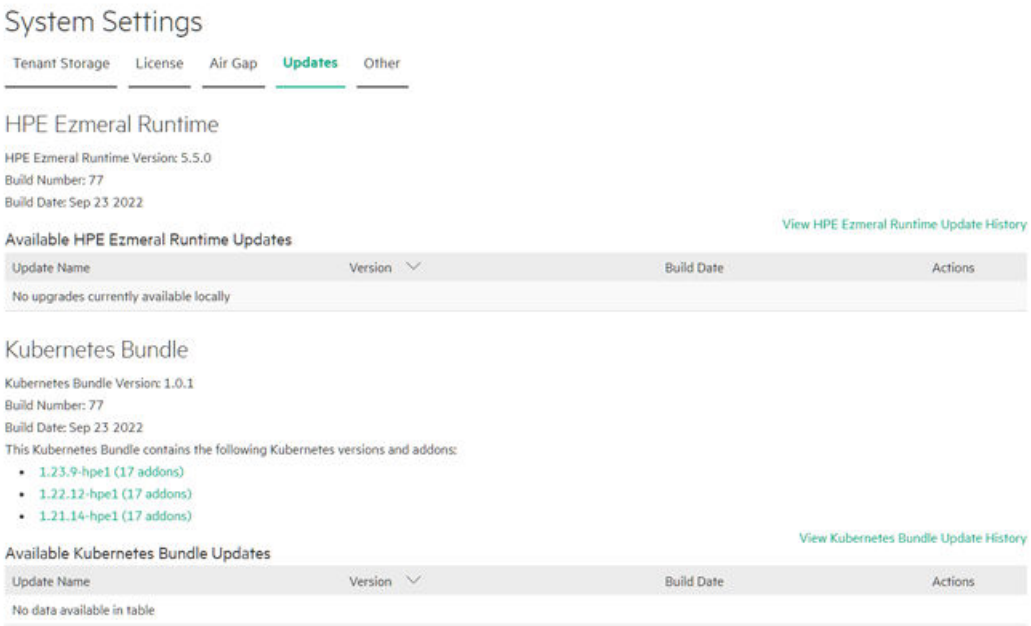
More information

Using the Air Gap Utility on page 869
Describes how to use the air gap utility to download files in an air-gapped HPE Ezmeral Runtime Enterprise environment.

Kubernetes Air-Gap Requirements on page 834

Updates Tab

The **Updates** tab enables the Platform Administrator to view the installed HPE Ezmeral Runtime Enterprise software and Kubernetes Bundles, view the history of update attempts, view the available updates, and to initiate and monitor updates.



The **Updates** tab enables the Platform Administrator to view the following details of the HPE Ezmeral Runtime Enterprise software, and the Kubernetes Bundles:

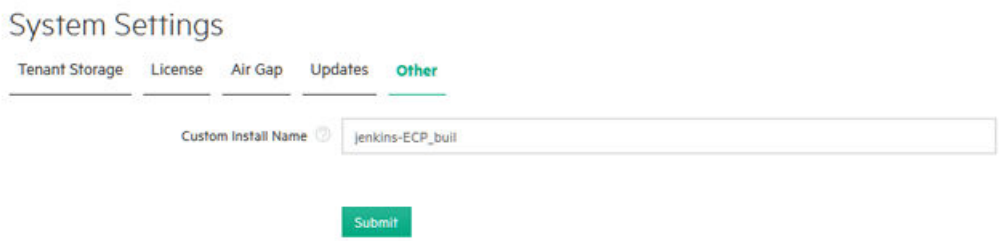
- Version number
- Build Number
- Build Date
- HPE Ezmeral Runtime Enterprise Update History
- Available HPE Ezmeral Runtime Enterprise Updates
- Kubernetes version and add-on details

- Kubernetes Bundle Date History
- Available Kubernetes Bundles Updates.

Other Tab
The **Other** tab of the **System Settings** screen enables the Platform Administrator to specify and view certain information about on-premesis deployments of HPE Ezmeral Runtime Enterprise.

NOTE: This tab only appears when HPE Ezmeral Runtime Enterprise is installed on your premises.

This tab appears as follows:



This tab displays the following information:

- The **Custom Install Name** field allows you to specify a custom name for the deployment that will appear in a green band on the left side of the **Toolbar**. This name may be up to 16 characters in length and may consist of letters (A-Z, a-z), digits (0-9), spaces, underscores (_), and/or dashes (-). The Custom Install Name is the only option that is present in HPE Ezmeral Runtime Enterprise Essentials.

To modify these settings, enter Lockdown mode (see Lockdown Mode on page 916), make your desired changes, and then click the **Submit** button to save your changes.

System Maintenance

This topic describes the preparation and post-procedure tasks to perform maintenance, such as OS patches and upgrades, on hosts that are part of an HPE Ezmeral Runtime Enterprise deployment.

Use the information in this topic when you perform maintenance, such as OS patches and upgrades, on hybrid or on-premises hosts that are part of an HPE Ezmeral Runtime Enterprise deployment.

To update the Kubernetes version on a host, see Upgrading Kubernetes on page 487.

HPE Ezmeral Runtime Enterprise automatically drains the nodes during platform software upgrades and when upgrading Kubernetes to a newer version. In most cases, a best practice is to drain a Kubernetes node before performing system maintenance tasks such as OS kernel updates or hardware repairs. If you choose to drain a Kubernetes node, be aware that the HPE Ezmeral Runtime Enterprise software uses DaemonSet, so you must use the `--ignore-daemonset` option of the kubectl `drain` command. If your deployment uses custom DaemonSets or PodDisruptionBudgets, there might be other considerations when draining nodes. If you need help to evaluate your deployment and perform a system maintenance task, contact Hewlett Packard Enterprise Technical Support.

Preventing Unintended Updates of Kubernetes Packages

The `yum update` command, by default, attempts to update packages from all enabled repositories, including the repository that manages the `kubeadm`, `kubelet`, and `kubecl` packages. However, updating that repository using `yum` is not the correct upgrade procedure, and it can result in the installation of a package version that is not compatible with the current HPE Ezmeral Runtime Enterprise deployment, which leads to failures that result in applications not running correctly. In addition, the installed package version no longer matches the package version listed in the HPE Ezmeral Runtime Enterprise UI.

To prevent yum from updating Kubernetes packages as part of operations such as rebooting a host, on each host, ensure that the `yum update` command is prevented from updating the Kubernetes repo:

1. Open the following file in an editor:

   ```
   /etc/yum.repos.d/bd-kubernetes.repo
   ```

2. The parameter `enabled=1` indicates that updates are enabled. To disable updates, change the parameter to `enabled=0`.

If you want to use yum to update other packages, run the `yum update` command without the `-y` option so that you can individually deny any Kubernetes packages updates that show as available.

Performing System Maintenance

1. If this host is a Kubernetes host, see Preventing Unintended Updates of Kubernetes Packages on page 802

2. Enter Lockdown mode, as described in Lockdown Mode.

3. Upgrade the Controller host, then reboot the host, and then wait for all services to come back up in the **Services** tab of the Platform Administrator **Dashboard** (see Dashboard - Platform Administrator on page 570).

4. If you have platform High Availability enabled, then repeat Step 3 for the Shadow Controller host.

5. If you have platform High Availability enabled, then repeat Step 3 for the Arbiter host.

6. Upgrade one Worker host, then reboot that Worker, and then wait for all services to come back up in the **Services** tab of the Platform Administrator **Dashboard**.

   If this host is a GPU host, NVIDIA GPU drivers must be reinstalled after OS Kernel updates because the NVIDIA kernel module has a kernel interface layer that must be compiled specifically for each kernel. To reinstall the NVDIA GPU drivers, see **Steps 8 to 11** in GPU Driver Installation.

7. Repeat Step 6 for each remaining Worker host, ensuring that you perform the entire process on one host at a time.

8. After all hosts have been fully rebooted, exit Lockdown mode and then perform the tests described in Validating the Installation to verify that the platform is functioning normally.

If you want to remove RPM packages after HPE Ezmeral Runtime Enterprise is installed, then be sure not to remove any required packages.

The Kubernetes RPMs file for air gap installations contains the required RPMs. See Configuring Air Gap Kubernetes Host Settings on page 868. If you need a separate list of RPMs for this version of HPE Ezmeral Runtime Enterprise, contact your Hewlett Packard Enterprise Support representative.

Planning the Deployment

A high-level overview of the items to consider when planning an HPE Ezmeral Runtime Enterprise deployment.

NOTICE: End of Life (EOL) for Elastic Private Instant Clusters (EPIC)

HPE Ezmeral Runtime Enterprise 5.4.1 is the last release that includes support for EPIC. Beginning with the next general availability release, deployments that use EPIC to manage virtual nodes/containers are not supported. No future enhancements to EPIC are planned; however, support (such as bug fixes) will continue to be provided until the EPIC functionality reaches End of Life (EOL).

Existing deployments that use EPIC can be transitioned to the newer Kubernetes-based solution on the latest HPE Ezmeral Runtime Enterprise release. Existing deployments that continue to use EPIC will be supported until EPIC reaches End of Life (EOL) on December 30, 2024.

This article provides a high-level overview of the items to consider when planning an HPE Ezmeral Runtime Enterprise deployment. These items include:

- **Storage:** The Storage on page 804 article describes the available storage schemas and the key advantages and considerations of each schema. The flowchart on this page helps you determine the best option.
- **Platform Resource Planning:** The flowchart in the Platform Resource Planning on page 806 article guides you through a number of additional configuration questions.

All of your planning considerations are subject to the system requirements. See System Requirements on page 808.

Storage

This article describes the various storage usages and how datasets are made available to the containerized clusters.

Container Local Data Storage

HDFS is provisioned within the containers that comprise a virtual Hadoop cluster when that cluster is created. The underlying storage for the HDFS data nodes in the containers resides on local disks in the physical servers hosting those containers. The deployment refers to the set of local disks as *node storage*. When using HDFS storage in a virtual cluster, the data does not persist beyond the life of the virtual cluster.

Ephemeral Storage

Ephemeral storage is built from the local storage in each host. It is used for the disk volumes that back the local storage for each virtual node. Installing a host reserves a subset of the local disks on that host for node storage. Physical Linux volumes are created on those disks and then used to create a Linux volume group. A Linux logical volume is then created from this Linux volume group. This Linux logical volume is assigned to the Linux container subsystem, which in turn uses portions of the logical volume to the containers running on that host for use as local storage within those containers.

Persistent Storage using HPE Ezmeral Data Fabric

A deployment of HPE Ezmeral Runtime Enterprise must use one HPE Ezmeral Data Fabric for persistent storage. You can choose which implementation of HPE Ezmeral Data Fabric that you use.

HPE Ezmeral Data Fabric on Bare Metal	**HPE Ezmeral Data Fabric on Bare Metal** is an implementation of HPE Ezmeral Data Fabric that is on physical or virtual machines that are not part of the HPE Ezmeral Runtime Enterprise deployment. You can connect from the HPE Ezmeral Runtime Enterprise deployment to a bare metal implementation of as external storage. Typically, you would choose this option if you have an existing deployment of HPE Ezmeral Data Fabric and

	you are adding a deployment of HPE Ezmeral Runtime Enterprise to your environment. To use this implementation as tenant/persistent storage in HPE Ezmeral Runtime Enterprise, you must do the following: • Do not specify any disks as Tenant/Persistent storage during the Platform Controller Setup portion of the installation procedure. • After you have installed and verified HPE Ezmeral Runtime Enterprise and configured Gateway hosts, you must register the implementation as tenant/persistent storage as described in HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579.
HPE Ezmeral Data Fabric on Kubernetes	**HPE Ezmeral Data Fabric on Kubernetes** is an implementation of HPE Ezmeral Data Fabric in a Kubernetes cluster instead of on physical or virtual servers. To use this implementation as tenant/persistent storage in HPE Ezmeral Runtime Enterprise, you must do the following: • Do not specify any disks as tenant/persistent storage during the Platform Controller Setup portion of the installation procedure. • After you have installed and verified HPE Ezmeral Runtime Enterprise and configured Gateway hosts, you must create a new Kubernetes Data Fabric cluster and register that cluster for tenant/peristent storage as described in Creating a New Data Fabric Cluster on page 611.
Embedded Data Fabric	Embedded Data Fabric is not supported on new deployments of HPE Ezmeral Runtime Enterprise and later. This option is available only if you are upgrading from a 5.3.x version of HPE Ezmeral Runtime Enterprise and that deployment has an existing Embedded Data Fabric. If your deployment has an existing Embedded Data Fabric, that implementation was registered as tenant/persistent storage during the Platform Controller Setup portion of the HPE Ezmeral Runtime Enterprise installation procedure.

For more information about the different implementations of HPE Ezmeral Data Fabric, and about host and other requirements when implementing HPE Ezmeral Data Fabric on Kubernetes, see HPE Ezmeral Data Fabric on Kubernetes Administration on page 590.

Compute and Storage Separation

Getting the maximum flexibility from a container-based solution requires being able to independently scale compute and storage resources. It is also essential to be able to support the persistence of Big Data datasets beyond the lifespan of a Big Data compute cluster. The DataTap and IOBoost technologies allow virtual clusters to access remote data regardless of location or format.

A DataTap creates a logical data lake overlay that allows access to shared data in the enterprise storage devices. This allows users to run Big Data and ML/DL jobs using the existing enterprise storage without needing to make time-consuming copies or transfers of data to local disks. IOBoost augments DataTap's

flexibility by adding an application-aware data caching and tiering server to ensure high-speed remote data delivery.

This persistent storage can also serve as filesystem mount storage (FS mounts). The filesystem mount feature allows automatically adding mounts to virtual nodes/containers, thereby allowing virtual nodes/containers to directly access POSIX data as if they were local directories. You can use this feature to provide common files across all of the virtual nodes/containers in a given tenant, such as a common configuration file that will be used by all of the virtual nodes/containers in the Marketing tenant. This eliminates the need to manually copy common files to individual virtual nodes/containers.

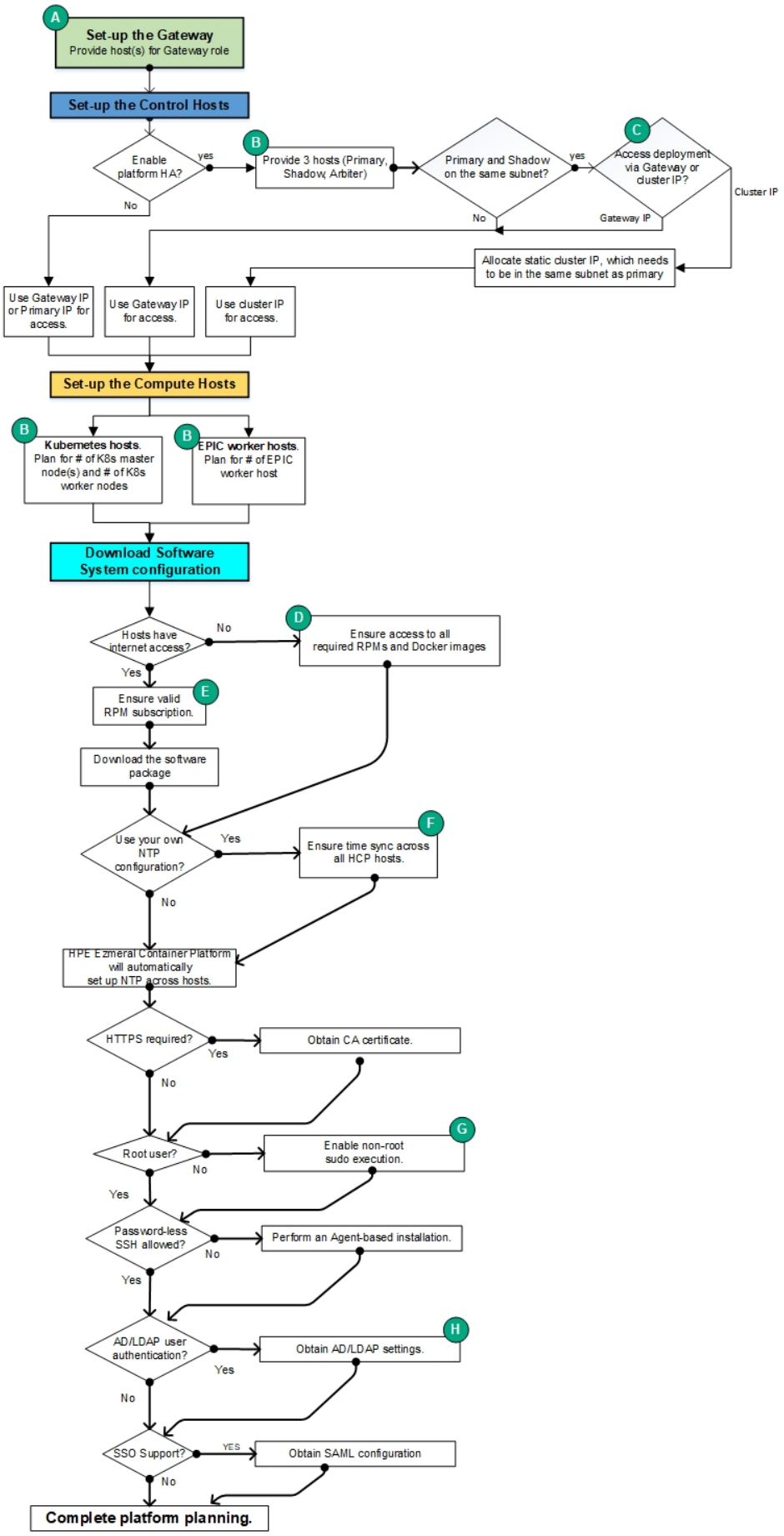
All applications running in containers can natively access data across the HPE Persistent Storage fabric via both DataTaps and FS mounts. Persistent volumes are seamlessly available across clusters from this persistent data fabric.

Operating System Storage

For all host types, the recommended storage for the operating system is two 960 GB SSD's in a RAID 1 configuration. See Host Requirements for detailed storage requirements and recommendations.

Platform Resource Planning

This diagram provides a workflow for making decisions about how to configure HPE Ezmeral Runtime Enterprise:



NOTES:

A One or more Gateway host(s) are required. Multiple Gateway hosts recommended for load balancing and HA. See Host Requirements.

B See "Host Requirements."

Please see the following for additional information:

- High Availability
- Tenant and Project Storage
- Node Storage
- Storage
- Gateway Hosts
- Host Requirements
- Operating System Requirements
- The Controllers & HA Screen on page 754
- Configuration Requirements
- Adding an SSL Certificate
- Restricted Sudo Privileges
- Installation Overview
- Installation, which will be either:
  - Standard
  - Using the Pre-Check Script, Sample Pre-Check Output, and Using the Pre-Check Config File
- Kubernetes Worker Installation Overview
- Gateway Installation Tab
- User Authentication (contains information on both SSO and non-SSO authentication)
- Configuring User Authentication Settings

Installing Root or Sudo User Password

The following information concerns the creation of root and sudo users and passwords in ECP.

When beginning ECP installation, specify a Linux install user (root or sudo). This user must be created before beginning the ECP installation process. That user cannot be deleted after installation, but the user's password can be changed. The user must remain across the entire ECP lifetime because all files and software installed by ECP on the platform hosts will be owned by that user.

Note that the user is only used for controller communication during initial install; afterwards, Erlang and RPC are used for controller communication.

When adding a new host to ECP, the controller will make an SSH connection to the new host. The user needs to either use an SSH key with no password enabled or the correct username and password. Passwords can be changed on the Linux host level; login to ECP admin is not necessary.

System Requirements

Your deployment must meet some or all of the following requirements, based on the types of operations you will be performing:

General

All deployments must meet all of the following requirements in order to install and run Big Data jobs, such as ActionScripts:

- Browser
- Port
- Host
- Operating System
- Web Proxy
- Network
- Configuration
- Restricted Sudo

Kubernetes

If you plan to run Kubernetes, then the deployment must meet the following requirements in addition to the general requirements:

- Controller
- Gateway
- Host
- Air gap
- Port

HPE Ezmeral ML Ops

If you plan to use EPIC AI/ML projects, then the deployment must meet the HPE Ezmeral ML Ops Requirements in addition to the general requirements.

General Requirements

The topics in this section describe the general requirements for deploying HPE Ezmeral Runtime Enterprise. Depending on the features and applications you include in your deployment, additional requirements might apply.

Browser Requirements

The web interface is accessible via plain HTML (HTTP) or secure HTML (HTTPS; see Adding an SSL Certificate) using the following browsers:

- Chrome: Version 68.0.3440.106 (Official Build) (64-bit)
- Firefox: 61.0.2 (64 bit)

Port Requirements

The ports listed in the following table must be available for use by the deployment. If you will be running Kubernetes, then the requirements listed in Kubernetes Port Requirements also apply.

If the `firewalld` service is installed and enabled on the Controller, and the `firewalld` service is installed and enabled on all hosts before they are added to the deployment, the installer for HPE Ezmeral Runtime Enterprise automatically configures firewall rules to open the required ports.

Port	Service	Protocol	Host	Direction	Agent*	Comments
22	SSH access	TCP	Controller Worker Gateway	both	Not required	This port is needed for password-less SSH installations. It is not applicable for agent-based installations such as cloud deployments. HPE recommends enabling SSH access for troubleshooting purposes.
53	DNS server TCP port	TCP	Controller Worker Gateway	From cloud to on-premises (Primary/ Shadow/ Cluster IP)		DNS port forwarding.
53	DNS server UDP port	UDP	Controller Worker Gateway	From cloud to on-premises (Primary/ Shadow/ Cluster IP)		DNS port forwarding.
80	Apache HTTP access	TCP	Controller Worker Gateway	From cloud to on-premises (Primary/ Shadow/ Cluster IP)		This is technically not needed for Worker hosts; however, since these hosts may become the Shadow Controller when High Availability is enabled, enabling HTTP access is recommended for these hosts.
88 464	Kerberos UDP	UDP	Controller Worker	Both directions		This is technically not needed for Worker hosts; however, since these hosts may become the Shadow Controller when High Availability is enabled, enabling HTTP access is recommended for these hosts.
88 464 749 754	Kerberos TCP	TCP	Controller Worker Gateway	Both directions		
111	RPC bind on TCP	TCP	Controller Worker Gateway	egress		
111	RPC bind on UDP	UDP	Controller Worker Gateway	egress		
123	NTP server port	TCP	Controller Worker	egress		
443	Apache HTTPS access	TCP	Controller Worker Gateway	N/A (outside access only)	Not required	

2224	PCS daemon	TCP	Controller Worker	N/A	This need only be open between the Primary and Shadow Controllers.	Required. Before the platform can be configured, the PCS daemon needs to be started and enabled to boot on startup on each host. This daemon works with the PCD CLI command to manage syncing the configuration across all the nodes in the platform.
2888, 5181, 3888	HPE Ezmeral Data Fabric-ZK	TCP	Controller, Worker	Both directions		All communication occurs between nodes; end users need not access these ports.
4369	Erlang EPMD	TCP	Controller Worker Gateway	Both directions		
4789	VxLAN		Controller Worker Gateway	Both directions	This port must be open on – Primary and Shadow controllers, Arbiter and Gateway hosts.	
5405	Cluster Manager	UDP	Controller Worker	N/A		
5610 9210 9211	Monitoring	TCP	Controller Worker Gateway			9210 from cloud to on-premises.
5660-5787	HPE Ezmeral Data Fabric-FS	TCP	Controller, Worker	Both directions		All communication occurs between nodes; end users need not access these ports.
5659	NRPE access	TCP	Controller Worker Gateway	On-premises to cloud, for Nagios to be able to access NRPE running on the cloud VMs.	Nagios Remote Plugin Executor (NRPE).	If this port is blocked, then the **Services** tab of the **Cluster Details** screen will not be able to report service statuses.
7220-7222	HPE Ezmeral Data Fabric-CLBD	TCP	Controller Worker	Both directions		All communication occurs between nodes; the end user need not access these ports.
7443	HPE Ezmeral Data Fabric-Login	TCP	Controller, Worker	Both directions		Please contact HPE Technical Support if a user needs to directly access the HPE Ezmeral Data Fabric Management Console.
8080	bd_mgmt REST API	TCP	Controller Worker Gateway	N/A (outside access only)	Not required	

8081	haproxy stats	TCP	Controller Worker Gateway	N/A (outside access only)	Not required	
8085	Apache HTTP access for Container Platform Nagios	TCP	Controller Worker Gateway	N/A (outside access only)	Not required	
8443	Apache HTTPS access for Container Platform Nagios	TCP	Controller Worker Gateway	N/A (outside access only)	Not required	
8443	HPE Ezmeral Data Fabric-REST	TCP	Controller, Worker	Both directions		All communication occurs between nodes; end users need not access these ports.
9000 9001	Erlang RPC	TCP	Controller Worker Gateway	Both directions		• 9000: set up through VM argument • bd_mgmt 9001: dataserver
9002	Erlang SSH-RPC	TCP	Controller Worker Gateway			
9500-9699	Kubernetes API endpoints for individual clusters	TCP	Gateway		N/A	These ports are used by Gateway hosts to communicate with Kubernetes hosts, specifically to connect the Kubernetes API server to the a specific Kubernetes cluster.
14000 14001	HTTPFS		Controller Worker Gateway		Not required for external access.	
10000 to 50000	Container Platform Gateway host service mapping		Controller Worker Gateway	N/A (outside access only)		Random port definitions from the pool are not used. By default, port usage will start from 10000 and proceed incrementally. As virtual clusters are deleted, those ports will become usable by the pool and will be used for new services.
7220:7223	MapR CLDB	TCP		To HPE Ezmeral Runtime Enterprise Controller nodes		By default, CLDB listens on ports 7222 and 7223. For performance reasons, additional ports may be opened. For more details, see Ports Used by HPE Ezmeral Data Fabric Software.
8660	MapR-MAST	TCP		Both		Data Fabric clients use this port to connect to the MAST Gateway.

7443	MapR-Login	TCP		To HPE Ezmeral Runtime Enterprise Controller nodes		When security is enabled for a cluster, the CLDB listens for connections on port 7443. If security is disabled, the maprlogin utility is unable to reach the CLDB.
8443	MapR REST	TCP		To HPE Ezmeral Runtime Enterprise Controller nodes		MapR REST API
5660, 5692, 5724, 5756	MapR FS	TCP		Both		Only required if it is set up with Embedded Data Fabric. The filesystem is a random, read-write, distributed filesystem that allows applications to read and write concurrently directly to disk. Clients use these ports to access the file-system server.
2888, 5181, 3888	MapR-ZK	TCP		To HPE Ezmeral Runtime Enterprise Controller nodes		MapR ZooKeeper

**=Determine whether or not the port is used for agent-based installations.*

More information

Kubernetes Port Requirements on page 836

Host Requirements

This topic lists the minimum host requirements for HPE Ezmeral Runtime Enterprise for production environments and for non-production environments, such as for development and testing.

The minimum and recommeded host requirements vary by the following:

- Deployment environment: Production or non-production
- Workload type, such as ML Ops or Big Data
- Host function, such as compute, gateway, or storage

Deployment Environments

Deployment environments include the following:

Production environments	Hewlett Packard Enterprise strongly recommends that you use the recommendations for production environments when deploying HPE Ezmeral Runtime Enterprise for a production workload. The minimum production requirements are not appropriate for all workloads. The appropriate sizing varies by the workload type and your performance and capacity requirements. Hewlett Packard Enterprise can help you determine the best configuration for your needs.
Non-production environments	Non-production environments include demonstration environments, development environments, testing environments, and so forth.

The minimum host requirements described in this topic for non-production environment are the minimums that are required to run HPE Ezmeral Runtime Enterprise. Minimum deployments do not meet high-availiability requirements and have performance and capacity limitations. The appropriate sizing for non-production environments varies by the workload type and your performance and capacity requirements.

For production workloads and non-production environments larger than 10 nodes, Hewlett Packard Enterprise strongly recommends that you collaborate with your Hewlett Packard Enterprise representative to design an architecture that meets your requirements based on your actual workload needs.

Workload Types

The type of workload influences the appropriate mix of CPU, memory, storage, and networking resources. Workloads such as ML Ops workloads have additional requirements beyond the platform minimum requirements described in this topic. Links to additional information for different workloads are provided as appropriate.

Host Types by Function

The term **host** and **node** are often used interchangeably. Nodes are hosts that are part of a cluster. The types of hosts, by function, are the following:

Controller hosts

The Controller host is the host where you initially install HPE Ezmeral Runtime Enterprise. This host controls the rest of the hosts in the deployment.

In high-availability (HA) deployments, there is also a Shadow Controller host and an Arbiter host, for a total of three (3) Controller hosts.

Controller hosts are part of the HPE Ezmeral Runtime Enterprise control plane.

Gateway LB hosts

Gateway load balancer (Gateway LB) hosts enable access to pods or container services from an external network.

In high-availability (HA) deployments, there are a minimum of two (2) Gateway LB hosts. For more information about Gateway host requirements, see Gateway Hosts on page 106.

Gateway LB hosts are part of the HPE Ezmeral Runtime Enterprise control plane.

Kubernetes control plane hosts

The Kubernetes control plane manages the worker hosts and pods in the cluster. For detailed information about what a Kuberentes control plane does, see Control Plane Components in the Kubernetes documentation (links opens an external website in a new browser window or tab).

In high-availability (HA) deployments that implement Kubernetes, there are a minimum of three (3) Kubernetes master hosts.

Worker hosts

Worker hosts run the pods or containers that process jobs in HPE Ezmeral Runtime Enterprise.

Data Fabric hosts

Data Fabric are the hosts are part of an implementation of **HPE Ezmeral Data Fabric on**

Kubernetes. **Data Fabric hosts are not included in the minimum requirements tables in this topic.**

Requirements for Data Fabric hosts are described in Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.

If the HPE Ezmeral Runtime Enterprise deployment has already implemented Embedded Data Fabric instead of **HPE Ezmeral Data Fabric on Kubernetes**, there is no dedicated Data Fabric; worker hosts are used instead. This configuration is applicable only for existing HPE Ezmeral Data Fabric deployments in which HPE Ezmeral Runtime Enterprise is upgraded from a release prior to 5.4.0.

General Recommendations for Hosts

The following recommendations apply regardless of host type or workload:

- Hewlett Packard Enterprise strongly recommends that you install HPE Ezmeral Runtime Enterprise on dedicated physical or virtual hosts. Do not use these resources for any other applications or services.

  CAUTION:

  HPE Ezmeral Runtime Enterprise performs numerous configuration changes to the controller and worker hosts during installation that are required in order for the HPE Ezmeral Runtime Enterprise deployment to function. These changes are not completely reversible and may impact other applications and processes that are currently running on the host. It is strongly recommended that you install HPE Ezmeral Runtime Enterprise on hosts that are not being used for any other purpose in order to avoid possible disruptions to your normal business processes.

- You must have at least three (3) Controller hosts in order to enable platform High Availability for HPE Ezmeral Runtime Enterprise, as described in High Availability Requirements, below.

- For best results, Hewlett Packard Enterprise recommends installing HPE Ezmeral Runtime Enterprise on hosts that share the same configuration (CPU, RAM, storage, GPU, OS, etc.).

- For production workloads and non-production environments larger than 10 nodes, Hewlett Packard Enterprise strongly recommends that you collaborate with your Hewlett Packard Enterprise representative to design an architecture that meets your requirements based on your actual workload needs.

- Hewlett Packard Enterprise strongly discourages the installation of other applications, including security software such as McAfee and Trend Deep Security, on HPE Ezmeral Runtime Enterprise hosts. Any software that has the ability to change system-wide file ownership and filesystem mounting options can prevent HPE Ezmeral Runtime Enterprise from working correctly.

Production Environments: Minimum Host Requirements

The following tables describe the minimum host requirements for HPE Ezmeral Runtime Enterprise in production deployments:

- The table entries incorporate the recommended number of Controller, Gateway, and Kuberentes Master (Kubernetes Control Plane) nodes needed for high-availability (HA) deployments. For information about other HA requirements, see High Availability Requirements on page 819.

- In HPE Ezmeral Runtime Enterprise deployments that implement Kubernetes, Kubernetes Master hosts are separate from and in addition to the HPE Ezmeral Runtime Enterprise Controller hosts.

last-updated: May 06, 2024

- The tables differ based on whether Embedded Data Fabric is included in the deployment (only for existing Embedded Data Fabric deployments in which HPE Ezmeral Runtime Enterprise is upgraded from a release prior to 5.4.0.). For new deployments of HPE Ezmeral Runtime Enterprise 5.4.0 or later, Embedded Data Fabric is not supported.
- The recommendations for Kubernetes hosts do not include the additional requirements for add-on applications. See Kubernetes Host/Node Requirements on page 833 and Online Sizing Tool for Compute Capacity on page 820.
- The table entries do not include host requirements for deployments that include a separate cluster for **HPE Ezmeral Data Fabric on Kubernetes**. For information about requirements for HPE Ezmeral Data Fabric on Kubernetes, see Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.
- In the tables, CPU cores are defined as the cores available to the kernel in the OS on which the HPE Ezmeral Runtime Enterprise software is directly installed. For the complete definition, see HEWLETT PACKARD ENTERPRISE SOFTWARE END USER SUBSCRIPTION AGREEMENT on page 87.

Table

Role	Controller	Gateway LB	Kubernetes Master	Kubernetes Worker
Quantity	3	2	3	1
CPU Cores	4	2	4	4
RAM (GB)	64	16	32	32
NIC (Gbps)	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps
Storage	See Storage Requirements.			
GPU[9]	N/A	N/A	N/A	Optional

GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.

Table

Role	Controller	Gateway LB	Kubernetes Master	Kubernetes Worker
Quantity	3	2	3	1
CPU Cores	8	2	4 (8 if this is an MFS node)	4 (8 if this is an MFS node)
RAM (GB)	64	16	32 (64 if this is an MFS node)	32 (64 if this is an MFS node)
NIC (Gbps)	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps
Storage	See Storage Requirements.			
GPU[10]	N/A	N/A	N/A	Optional

GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.

[9] GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.
[10] GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.

Non-Production Environments: Minimum Host Requirements

The following tables describe the minimum host requirements for HPE Ezmeral Runtime Enterprise in development, testing, or other non-production deployments:

- The table entries are for non-HA deployments. For information about high-availability (HA) deployments, see High Availability Requirements on page 819.
- In HPE Ezmeral Runtime Enterprise that implement Kubernetes, the Kubernetes master host is separate from and in addition to the HPE Ezmeral Runtime Enterprise Controller host.
- The tables differ based on whether Embedded Data Fabric is included in the deployment (only for existing Embedded Data Fabric deployments in which HPE Ezmeral Runtime Enterprise is upgraded from a release prior to 5.4.0). For new deployments of HPE Ezmeral Runtime Enterprise 5.4.x and later, Embedded Data Fabric is not supported.
- The recommendations for Kubernetes hosts do not include the additional requirements for add-on applications. See Kubernetes Host/Node Requirements on page 833.
- The table entries do not include host requirements for deployments that include HPE Ezmeral Data Fabric on Kubernetes. For information about requirements for HPE Ezmeral Data Fabric on Kubernetes, see Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.
- The installation pre-check scripts are designed to check that the minimum requirements for production environments are met. For non-production environments, when you want to create a deployment that is smaller than the requirements for production environments, specify the `--force` option when executing the pre-check script.

Table

Role	Controller	Gateway LB	Kubernetes Master	Kubernetes Worker
Quantity	1	1	1	1
CPU Cores	4	2	4	4
RAM (GB)	32	16	32	32
NIC (Gbps)	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps
Storage	See Storage Requirements.			
GPU[11]	N/A	N/A	N/A	Optional

GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.

Table

Role	Controller	Gateway LB	Kubernetes Master	Kubernetes Worker
Quantity	1	1	1	1
CPU Cores	8	2	4 (8 if this is an MFS node)	4 (8 if this is an MFS node)
RAM (GB)	64	16	32 (64 if this is an MFS node)	32 (64 if this is an MFS node)

[11] GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.

Table (Continued)

Role	Controller	Gateway LB	Kubernetes Master	Kubernetes Worker
NIC (Gbps)	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps	1 x 10Gbps
Storage	See Storage Requirements.			
GPU[12]	N/A	N/A	N/A	Optional

GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.

Storage Requirements

The following table lists the minimum storage requirements for deployments of HPE Ezmeral Runtime Enterprise.

The recommendations for Kubernetes hosts do not include the additional requirements for add-on applications. See Kubernetes Host/Node Requirements on page 833.

The table does not include storage requirements for deployments that include HPE Ezmeral Data Fabric on Kubernetes. See Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.

Table

Role	Controller	Gateway LB	Kubernetes Master	Kubernetes Worker
OS Disk (GB)	50 or more*	20	50 or more*	50 or more*
Ephemeral Storage (GB)[13]	150	N/A	150	150

All storage must be surfaced as a raw block device; it cannot have or be a part of a partition or have any file systems mounted to the volume.

* The minimum size of the OS disk corresponds to the root `/` mount point. The required size depends on the existence of other mount points. See Storage Partition Requirements on page 818.

Storage Partition Requirements

The supported mount points and their minimum sizes depends on the type of host. The host file system must have at least the root mount point: `/`

The total minimum required size is the combination of all the mount point sizes listed in the table for a given type of host. If you choose not to configure a listed mount point, that mount point's required size must be added to the root (`/`) mount point.

For example, if you choose not to configure `/opt` as a separate mount point on the Controller host, you must add the 100GB listed for `/opt` to the 50GB listed for the root mount point (/). That is, if `/opt` is not a separate mount point, the Controller host requires 150GB for the root (`/`) mount point.

The storage size for the Controller and Shadow Controller hosts must match.

Table

Mount Point	Minimum Size (GB)	Purpose
`/`	50	Root file system where the HPE Ezmeral Runtime Enterprise components are stored

[12] GPUs are optional; they are not required to install HPE Ezmeral Runtime Enterprise.
[13] All storage must be surfaced as a raw block device; it cannot have or be a part of a partition or have any file systems mounted to the volume.

Table (Continued)

Mount Point	Minimum Size (GB)	Purpose
`/var`, or `/var/lib`, or `/var/lib/docker`	150	Stores container metadata information
`/opt`	100	Stores all HPE Ezmeral Runtime Enterprise software
`/srv` or `/srv/bluedata`	20	`/srv/bluedata` stores all temporary runtime files, including any artifacts, such as scripts and .jar files, that have been uploaded for running jobs.

Table

Mount Point	Minimum Size (GB)	Purpose
`/`	70	Root file system where the HPE Ezmeral Runtime Enterprise components are stored.
`/var`, or `/var/lib`, or `/var/lib/containerd` , or `/var/lib/docker`	150	Stores container metadata information. `/var/lib/containerd` is used for hosts running the Hewlett Packard Enterprise distribution of Kubernetes. `/var/lib/docker` is used for the other hosts in the deployment.
`/opt`	50	Stores all HPE Ezmeral Runtime Enterprise software. `/opt/ezkube` (on Kubernetes hosts hosts only), `/opt/bluedata`, and `/opt/hpe` are used to install HPE Ezmeral Runtime Enterprise software.

The preceding tables list **minimum** sizes. The optimum sizes for your deployment vary.

High Availability Requirements

If you plan to provide High Availability for the Controller host (see High Availability), then each of the following additional requirements must be met:

- The deployment must consist of at least **three (3)** Controller hosts. You will install HPE Ezmeral Runtime Enterprise on the Controller host and then add the hosts that will become the Shadow Controller and Arbiter.
- If you are using a public (routable) virtual node network, then each of the three Controller hosts (Controller, Shadow Controller, and Arbiter) must have IP addresses that fall within the same subnet. Further, you must have an additional IP address available within the same subnet for use as the cluster IP address. This requirement does not apply if you are using a private (non- routable) virtual node network. See Network Requirements on page 825.
- If both the Controller host and Shadow Controller host are in the same subnet, and you want to connect to the Controller via a cluster IP address, then the external network switch must support a “gratuitous ARP” based IP-to-MAC discovery. See High Availability on page 132.

- Hewlett Packard Enterprise recommends not installing the Network Manager service because it conflicts with the High Availability monitoring service. If you install the Network Manager service while installing the base operating system on the nodes, then HPE Ezmeral Runtime Enterprise will display a warning in the **Config Checks** tab of the **Support/Troubleshooting** screen (see Config Checks Tab).

Online Sizing Tool for Compute Capacity

For additional information about sizing the compute portion of your solution, see the HPE Sizing Tool for Ezmeral Container Platform (link opens in a new browser tab/window). The online sizer for HPE Ezmeral Runtime Enterprise sizes the compute tier where the HPE Ezmeral Runtime Enterprise software is running. It includes predefined templates that can aid you in determining workload requirements based on the applications you want to run. Compute capacity can be scaled by adding more compute servers as new tenants and services are brought online, without incurring the cost of scaling storage capacity unnecessarily.

Kubernetes Host Node Requirements

See Kubernetes Host/Node Requirements on page 833.

Host Node Requirements for HPE Ezmeral Data Fabric on Kubernetes

See Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.

HPE Ezmeral ML Ops Requirements

See HPE Ezmeral ML Ops Requirements on page 836.

Related reference

Network Requirements on page 825

Operating System Requirements

Operating system requirements vary depending on the specific OS being used.

NOTE:

There are also configuration requirements, some of which can vary by OS. See Configuration Requirements on page 826.

SLES

HPE Ezmeral Runtime Enterprise runs on the following versions of the SLES Linux operating system:

- SLES Linux Enterprise Server v15 SP3
- SLES Linux Enterprise Server v15 SP2

For SLES installations:

- HPE strongly recommends using only dedicated hosts with clean OS installations on them. Installing HPE Ezmeral Runtime Enterprise on hosts with other running applications can cause unpredictable behavior. To ensure your OS has the latest packages, Hewlett Packard Enterprise recommends performing a `zypper update` before installation.
- Secure boot is not supported.
- No other version of this operatng system is supported.

See also Configuration Requirements on page 826.

RHEL/CentOS

HPE Ezmeral Runtime Enterprise runs on the versions of RHEL and CentOS described in OS Support on page 85.

Hewlett Packard Enterprise strongly recommends using only dedicated hosts with clean OS installations on them. Installing HPE Ezmeral Runtime Enterprise on hosts with other running applications can cause unpredictable behavior. To ensure your OS has the latest packages, Hewlett Packard Enterprise recommends performing a `yum update` before installation.

Use the standard OS kernel; modifications may cause HPE Ezmeral Runtime Enterprise to function unpredictably.

To minimize the need for troubleshooting, Hewlett Packard Enterprise recommends newer kernel versions.

HPE Ezmeral Runtime Enterprise does not support upgrades between major OS versions. For example, if you are migrating from OS version 6.x to 7.x, you must perform a new installation (not an upgrade), and then install HPE Ezmeral Runtime Enterprise.

RHEL systems must have active, valid subscriptions in order to access the RHEL RPM repositories. See Configuration Requirements on page 826.

The Kubernetes RPMs file for air gap installations contains the required RPMs. See Configuring Air Gap Kubernetes Host Settings on page 868.

View the full list of RPM repositories from the following links:

- RPM list (RHEL-7) (link opens an external website in a new browser tab/window)
- RPM list (RHEL-8) (link opens an external website in a new browser tab/window)

HPE Ezmeral Runtime Enterprise has a dependency on the Pacemaker RPM Package, which is used for high availability of the Control Plane.

RHEL 8.x support: In addition to the general requirements for RHEL, the following items apply to RHEL 8.x support:

- RHEL 8.x is supported on Kubernetes hosts. For fresh installations only, RHEL 8.x is also supported on HPE Ezmeral Runtime Enterprise control plane (Controller, Shadow, Arbiter, and Gateway) hosts.
- Both GPU and non-GPU Kubernetes hosts are supported with RHEL 8.x.
- A Kubernetes cluster of mixed RHEL 8.x and RHEL 7.x or CentOS 7.x nodes is not supported.
- Firewall is supported only in `iptables` mode for RHEL 8.x.

See also:

- Configuration Requirements on page 826
- If this is a Kubernetes host, see also Kubernetes Host Requirements

Web Proxy Requirements

HPE Ezmeral Runtime Enterprise hosts use the system web proxy configuration for all Internet access. The proxy configuration must be the same on each host in the deployment.

If your deployment is not an air-gapped deployment, HPE Ezmeral Runtime Enterprise use the Docker service to pull images from various public registries on the Internet.

If your environment requires a web proxy, you must configure the web proxy on **all** hosts, as follows:

- On the Controller, Shadow Controller, Arbiter, and Gateway hosts, configure the web proxy for the Docker service, even if all the other hosts in the deployment use the containerd runtime.

- On Kubernetes hosts that use the Docker runtime, configure the web proxy for the Docker service.

  For example, if a Kubernetes cluster was created in a deployment of an HPE Ezmeral Runtime Enterprise release prior to 5.5.0, the hosts in that cluster continue to use the Docker runtime, even after you upgrade HPE Ezmeral Runtime Enterprise. If you expand that cluster, the hosts you add to that cluster must also use the Docker runtime. The hosts continue to use the Docker runtime until you manually migrate the cluster to the HPE distribution of Kubernetes.

- On hosts that use the containerd runtime, configure the web proxy for the containerd service.

  For example, hosts on which you install the HPE Kubernetes distribution use the containerd runtime.

NO_PROXY Settings

When you configure web proxy settings, you can also configure NO_PROXY settings to specify what is **not** to be accessed through the web proxy. Hewlett Packard Enterprise recommends that you include the following items in the NO_PROXY settings:

- The IP address of the Controller and Shadow Controller.
- The Fully Qalified Domain Name (FQDN) of the Gateway or Gateway sets.
- The pod DNS domain name. For example: `cluster.local`
- The localhost name and IP address. For example: `localhost,127.0.0.1`
- The private IP address range in CIDR format. For example: 192.168.0.0/16

Configuring the Proxy for the containerd Service

Required access rights: Platform Administrator

The web proxy for the containerd service is configured using the same method for all the host OSs supported by HPE Ezmeral Runtime Enterprise.

1. If the following file does not exist, create the file: `/etc/sysconfig/proxy`
2. Open the `/etc/sysconfig/proxy` file for editing.
3. Set `PROXY_ENABLED="yes"`
4. Enter the HTTP_PROXY values. For example:

   ```
   # Some programs (e.g. lynx, arena and wget) support proxies, if set in
   # the environment.
   # Example: HTTP_PROXY="http://proxy.provider.de:3128/"
   HTTP_PROXY="http://web-proxy.corp.mycorp.net:8080"
   ```

5. Enter the HTTPS_PROXY values. For example:

   ```
   # This setting is for https connections
   HTTPS_PROXY="http://web-proxy.corp.mycorp.net:8080"
   ```

6. Enter the NO_PROXY values. For example:

```
# Example: NO_PROXY="www.me.de, .do.main, localhost"
#
NO_PROXY=".svc,localhost,10.96.0.0/12,192.168.0.0/16,.default.svc,.storag
e.mycorp.net,127.0.0.1,.cluster.local"
```

7. Save and close the file.

8. If containerd was already installed and running, then restart the service:

```
systemctl daemon-reload
 systemctl restart containerd
```

NOTE: Restarting containerd brings down all running containerd containers, which disrupts any running services.

Otherwise, when you install HPE Ezmeral Runtime Enterprise, containerd will be installed and will use the `/etc/sysconfig/proxy` file.

Example Proxy File for SLES/SUSE

The following is an example of a `/etc/sysconfig/proxy` file for SLES/SUSE:

```
## Path:    Network/Proxy
## Description:
## Type:    yesno
## Default:    no
## Config:      kde,profiles
#
# Enable a generation of the proxy settings to the profile.
# This setting allows to turn the proxy on and off while
# preserving the particular proxy setup.
#
PROXY_ENABLED="yes"

## Type:    string
## Default:    ""
#
# Some programs (e.g. lynx, arena and wget) support proxies, if set in
# the environment.
# Example: HTTP_PROXY="http://proxy.provider.de:3128/"
HTTP_PROXY="http://web-proxy.corp.mycorp.net:8080"

## Type:    string
## Default:    ""
#
# Some programs (e.g. lynx, arena and wget) support proxies, if set in
# the environment.
# This setting is for https connections
HTTPS_PROXY="http://web-proxy.corp.mycorp.net:8080"

## Type:    string
## Default:    ""
#
# Example: FTP_PROXY="http://proxy.provider.de:3128/"
#
FTP_PROXY="http://web-proxy.corp.mycorp.net:8080"
```

```
## Type:    string
## Default:    ""
#
# Example: GOPHER_PROXY="http://proxy.provider.de:3128/"
#
GOPHER_PROXY=""

## Type:    string
## Default:    ""
#
# Example: SOCKS_PROXY="socks://proxy.example.com:8080"
#
SOCKS_PROXY=""

## Type:    string
## Default:    ""
#
# Example: SOCKS5_SERVER="office-proxy.example.com:8881"
#
SOCKS5_SERVER=""

## Type:    string(localhost)
## Default:    localhost
#
# Example: NO_PROXY="www.me.de, .do.main, localhost"
#
NO_PROXY=".svc,localhost,10.96.0.0/12,192.168.0.0/16,.default.svc,.storage.m
ycorp.net,127.0.0.1,.cluster.local"
```

Configuring the Proxy for the Docker Service (RHEL/CentOS)

Required access rights: Platform Administrator

Create any needed directories as the `root` user.

You can complete this task either before or after you install Docker and HPE Ezmeral Runtime Enterprise.

1. Create the `/etc/systemd/system/docker.service.d/docker-proxy.conf` file that contains your HTTP_PROXY, HTTPS_PROXY, and NO_PROXY parameters:

   ```
   cat <<EOF >  /etc/systemd/system/docker.service.d/docker-proxy.conf
   > [Service]
   >Environment="HTTP_PROXY= @@@YOUR_HTTP_PROXY_PARAM@@@"
   >Environment="HTTPS_PROXY= @@@YOUR_HTTPS_PROXY_PARAM@@@"
   >Environment="NO_PROXY= @@@YOUR_NO_PROXY_PARAMS@@@"
   >EOF
   ```

2. If the Docker daemon was already installed and running, then restart it:

   ```
   systemctl daemon-reload
   systemctl restart docker
   ```

   NOTE: Restarting the Docker daemon brings down all running Docker containers, which disrupts any running services.

   Otherwise, when you install HPE Ezmeral Runtime Enterprise, Docker will be installed and will use the `docker-proxy.conf` file.

Configuring the Proxy for the HTTP/HTTPS and FTP services (RHEL/CentOS only)

Add the following lines to `/etc/profile.d/set_proxy.sh`, replacing items in placeholders, such as `<web_proxy_url>`, with your own values:

```
export http_proxy=<web_proxy_url>:<port>
export https_proxy=<web_proxy_url>:<port>
export ftp_proxy=<web_proxy_url>:<port>
export no_proxy="localhost,127.0.0.1, <controller_ip>, <gateway_ip>,
<worker1_ip>, …, <worker_ip>"
```

Configuring the Proxy for the YUM packaging service (RHEL/CentOS only)

Add the proxy setting to: `/etc/yum.conf` replacing items in placeholders, such as `<web_proxy_url>`, with your own values:

For example:

```
proxy = <web_proxy_url>:<port>
```

Network Requirements

General Network Requirements

- Each host in the deployment must include at least one 10Gb Ethernet card.
- The network must be configured with either:
  - A DHCP server that supports assigning static IP address by device MAC address.
  - A network that includes a block of static IP addresses reserved out of the DHCP pool.
- The host names assigned to the servers during installation must be available throughout the life of the deployment.
- IP addresses assigned to virtual nodes/containers must fall within the floating IP ranges that specified during or after installation.

The other network requirements vary depending on whether your virtual node/container network will use routable or non-routable floating IP address ranges:

- For networks with non-routable floating address ranges, see Private, Non-routable Virtual Node/Container Addresses. For non-routable networks, users must access containers via the Gateway hosts using HAProxy.

Private, Non-routable Virtual Node/Container Addresses

HPE Ezmeral Runtime Enterprise maintains a list of the virtual nodes and ports that users may need to connect to and makes those ports available through one or more Gateway hosts. Each Gateway host maps a range of ports to services running on the containers within the deployment. A user who needs to access a container uses the hostname of the Gateway host and a port number. The hostname can be either the name of a Gateway host or of a physical load balancer.

For example, assume that the deployment has a Gateway worker with the hostname `gateway-1.mycompany.com`. In this example, the port mappings could appear as follows:

- `virtualnode-1.bdlocal`
  - **Hive thrift server:** `gateway-1.mycompany.com:10020`
  - **MySQL server:** `gateway-1.mycompany.com:10018`

  - **Spark master:** `spark//gateway-1.mycompany.com:10019`
  - **SSH:** `gateway-1.mycompany.com:10017`
- `virtualnode-2.bdlocal`
  - **SSH:** `gateway-1.mycompany.com:10022`
- `virtualnode-3.bdlocal`
  - **SSH:** `gateway-1.mycompany.com:10024`

NOTE: Large deployments might need multiple Gateway nodes. If these nodes share a hostname, then round-robin load balancing will take place among the available Gateway nodes.

Requirements for Using Multiple Subnets

When configuring the container network to use multiple subnets, the following requirements apply:

- The hosts can be located on-premises, in a public cloud, or both. For example, hosts can reside on multiple racks and/or can be virtual machines residing on cloud-based services (such as AWS, Azure, or GCP).
- If the deployment includes cloud-based hosts, then the container network must be private and non-routable. The container network is private for Kubernetes deployments.
- If Platrform HA is configued with a cluster IP address, then the Controller and the Shadow Controller must be on the same subnet.
- If the Controller and Worker hosts are on different subnets, then the path MTU settings must be the same for both subnets.
- If the Controller and Worker hosts are on a single subnet and the Gateway hosts are on the different subnet, then:
  - The subnet with the Gateway hosts can use an MTU setting that is lower than or equal to the MTU setting on the other subnet with no further action needed.
  - If the MTU of the Gateway host is larger, then it must be at least 1,000 bytes larger than the MTU setting of the other subnet.

  The subnets used by Gateway hosts can have different path MTU settings, subject to the preceding requirements.
- If the Controller and Shadow Controller are on one subnet, the Worker hosts on a second subnet, and the Gateway hosts on a third subnet, then each of the hosts must have the same path MTU setting.

Configuration Requirements

CAUTION: You must complete all configuration tasks prior to installing HPE Ezmeral Runtime Enterprise on the hosts.

Some configuration requirements for HPE Ezmeral Runtime Enterprise vary by OS.

SLES Requirements

When HPE Ezmeral Runtime Enterprise is running on SLES, the following general configuration is required on all hosts:

- Sudo must be installed.

- On SLES15 SP3 HPE Ezmeral Runtime Enterprise supports `firewalld` using the `iptables` backend. Edit `/etc/firewalld/firewalld.conf` to change the value of `FirewallBackend` to `iptables`, then restart the `firewalld` service.
- SELinux, if enabled, must use the targeted policy, as described here (link opens an external website in a new browser tab or window).**
- AppArmor is not supported.**
- Systemd is supported in legacy mode.**
- IPv6 is not supported and must be disabled.**

The following SLES kernel command line configures the preceding items marked with asterisks (**):

```
systemd.unified_cgroup_hierarchy=0
systemd.legacy_systemd_cgroup_controller=1 apparmor=0 cgroup_enable=memory
swapaccount=1 ipv6.disable=1 security=selinux selinux=1
```

Also, the GRUB2 boot loader must be updated, as described here (link opens an external website in a new browser tab or window).

Common Host Packages (SLES)

All HPE Ezmeral Runtime Enterprise hosts that have SLES v15 SP3 must have the following modules enabled:

- `SUSEConnect -p PackageHub/15.3/x86_64`
- `SUSEConnect -p sle-module-legacy/15.3/x86_64`
- `SUSEConnect -p sle-module-python2/15.3/x86_64`
- `SUSEConnect -p sle-module-basesystem/15.3/x86_64`
- `SUSEConnect -p sle-module-public-cloud/15.3/x86_64`
- `SUSEConnect -p sle-module-desktop-applications/15.3/x86_64`

All HPE Ezmeral Runtime Enterprise hosts that have SLES v15 SP2 must have the following modules enabled:

- `SUSEConnect -p PackageHub/15.2/x86_64`
- `SUSEConnect -p sle-module-legacy/15.2/x86_64`
- `SUSEConnect -p sle-module-python2/15.2/x86_64`
- `SUSEConnect -p sle-module-basesystem/15.2/x86_64`
- `SUSEConnect -p sle-module-public-cloud/15.2/x86_64`
- `SUSEConnect -p sle-module-desktop-applications/15.2/x86_64`

Primary Controller and Shadow Controller Host Packages (SLES)

The Controller host (and Shadow Controller host, if platform HA is enabled) must have the following SLES module enabled in addition to the common packages listed in Common Host Packages (SLES) on page 827:

- SP3: `SUSEConnect -p sle-ha/15.3/x86_64`
- SP2: `SUSEConnect -p sle-ha/15.2/x86_64`

The Arbiter host does not require the preceding module to be enabled.

Additional Kubernetes Requirements (SLES)

The following additional requirement applies to Kubernetes within HPE Ezmeral Runtime Enterprise on SLES:

- If the deployment is using an air gap, then see Kubernetes Air-Gap Requirements.

RHEL and CentOS Requirements

When HPE Ezmeral Runtime Enterprise is running on RHEL or CentOS, the following general configuration is required on all hosts:

User Account	For information about the requirements for the user account that will be installing HPE Ezmeral Runtime Enterprise, see User Account. Console and SSH access to either the root account or a non-root user account with sudo privileges is required. See Restricted Sudo Requirements and Configuration Requirements on page 826.
SSHD	Controls how the Controller communicates with Workers. See SSHD.
ARP	Address Resolution Protocol. See ARP.
`umask` values	The supported `umask` values, are 022, 027, or 077.
SSL certificate	Install one if you will be accessing the web interface through HTTPS:// and not HTTP://. See SSL Certificate.
SELinux/IPtables	See SELinux/IPtables.
IPv6	IPv6 is not supported. However, the IPv6 module must be enabled but not used. Enabling the IPv6 module helps to avoid runtime errors and warnings.

Version-specific configuration requirements:

RHEL 7	• On all hosts, you need the following subscriptions enabled:
	  • rhel-7-server-rpms
	  • rhel-7-server-optional-rpms
	  • rhel-7-server-extras-rpms
	• Along with the preceding channels, on Controller and Shadow controller, you need the following subscription enabled:
	  • rhel-ha-for-rhel-7-server-rpms
	  The Arbiter host does not require the HA module to be enabled.

RHEL 8	• On Kubernetes hosts, you need the following subscriptions enabled:
	  • rhel-8-server-rpms
	  • rhel-8-server-optional-rpms
	• Along with the preceding channels, on Controller and Shadow controller, you need the following subscription enabled:
	  • rhel-8-for-x86_64-highavailability-rpms
	  The Arbiter host does not require the HA module to be enabled.

Centos 7.x	`Base` and `extras` repositories are required on all hosts.

OS Locale

To install HPE Ezmeral Runtime Enterprise on a host, the system locale setting must be set to United States English with UTF-8 encoding. For example:

```
LANG=en_US.UTF-8
```

User Account

Console and SSH access to either the root account, or a non-root user account with sudo privileges is required.

The user account requirements are the following:

- The user account that is employed for the initial installation must also be available on all hosts that will be added as Worker hosts. Credentials for that account (either password or SSH key) must be available for all hosts.
- If you are installing as the root user, then SSHD must be configured to allow root login on all hosts, as described in SSHD.
- If you are installing as a non-root user, then that user (for example, the service user account) must have sudo permissions to execute the specific binaries listed in Restricted Sudo Requirements without restrictions on all hosts in the deployment.
  - By default, the `sudoers` file is configured to include files located in the `/etc/sudoers.d` directory. Do not change this default configuration.
  - The non-root service account user must be part of these groups: `docker`, `nagios`, and `apache`. If the user is not part of these groups, you must add the user to the groups manually.

SSHD

The SSHD service allows the Controller host to communicate directly with Worker hosts through passwordless SSH when adding the Worker hosts. If enabled, all hosts must have the OpenSSH server and client service running on port 22 on each host with a `umask` of either 022, 027, or 077.

The following configuration only affects how the Controller communicates with Workers. It does not affect user access to containers through SSH.

The following parameters must be set in `/etc/ssh/sshd_config` on the Controller host and on each Worker host:

- `PubkeyAuthentication=true`
- `AuthorizedKeysFile=.ssh/authorized_keys`
- `PermitRootLogin=yes` (if the Controller will be accessing the Workers as the root user. If the Controller will be acting as a non-root user, then this parameter is not required.)

Thus:

- If `PermitRootLogin=yes` and `PubkeyAuthentication=true`, then install as the root user.
- If `PermitRootLogin=no` but `PubkeyAuthentication=true`, then install as a non-root user.
- If `PubkeyAuthentication=false`, then use the `--worker-agent-install` option when installing the Controller host and the CLI agent when installing Worker hosts, as described in Standard Installation and Using the Pre-Check Script, respectively.

After you make changes to SSHD parameters, execute the `service sshd restart` command.

NOTE: If your environment does not permit passwordless SSH access for the installing user on all hosts in the deployment, then you must use the agent as described in Using the Pre-Check Script, Standard Installation, and Agent-Based Kubernetes Host Installation.

ARP

The ARP settings in the `/etc/sysctl.conf` configuration file for `arp_announce` and `arp_ignore` should be set to 0.

```
net.ipv4.conf.eth0.arp_ignore=0
net.ipv4.conf.eth0.arp_announce=0
```

SSL Certificate

If you want to access the web interface using a secure (https://) connection instead of a standard, non-secured connection (http://), then you must have both an SSL certificate and private key available when you install HPE Ezmeral Runtime Enterprise. You can use either a self-generated certificate or can obtain a certificate from a trusted Certificate Authority (CA). See Transport Layer Certificate (link opens an external website in a new browser/tab) for more information about SSL and certificates.

SELinux/IP Tables

For the Controller and any Worker hosts, you may choose to configure your deployment with or without these services. This decision cannot be changed after HPE Ezmeral Runtime Enterprise is installed.

SELinux is supported on HPE Ezmeral Runtime Enterprise 5.2 and later in Enforcing, Permissive, and Disabled mode as follows:

- To enable Enforcing mode on nodes that are part of HPE Ezmeral Data Fabric on Kubernetes, contact Hewlett Packard Enterprise Support.
- The mode cannot be changed after installing HPE Ezmeral Runtime Enterprise.
- For SLES 15 SP2 and SLES 15 SP3, supported with HPE-installed policies only.

IPv6

HPE Ezmeral Runtime Enterprise does not support IPv6. For configuration requirements, see RHEL and CentOS Requirements on page 828 and SLES Requirements on page 826.

(Optional) Container Security (Falco) Support

HPE Ezmeral Runtime Enterprise supports the Container Security (Falco) service. For more information, see Falco Container Runtime Security on page 499.

Air Gap RPMs

This article contains the lists of files necessary for air-gapped installation of HPE Ezmeral Runtime Enterprise.

The following are the lists of files for air-gapped installation.

HPE Ezmeral Runtime Enterprise 5.6.0 release:

- RPM list (North America download site)

Related reference

HPE Ezmeral Runtime Enterprise Air Gap Utility Release Notes on page 53
Change history and version compatibility information for the HPE Ezmeral Runtime Enterprise Air Gap Utility, `hpe-airgap-util`, on HPE Ezmeral Runtime Enterprise.

More information

Configuring Air Gap Kubernetes Host Settings on page 868
Download image and RPM files and configure settings for Kubernetes hosts in an air-gapped environment.

Using the Air Gap Utility on page 869
Describes how to use the air gap utility to download files in an air-gapped HPE Ezmeral Runtime Enterprise environment.

Kubernetes Air-Gap Requirements on page 834

Restricted Sudo Privileges

The term *sudo* stands for *super user do*. This technology allows one user to execute a command as another user. If HPE Ezmeral Runtime Enterprise is installed as a non-root/superuser user, that user must have `sudo` permissions to execute some commands as the superuser. A number of different tools are available for implementing `sudo` functionality. The most common such tool on the Linux operating system is called `sudo`. The `sudo` packages must be installed on each host in the HPE Ezmeral Runtime Enterprise deployment.

Security policies at your organization may require you to control access to the `sudo` commands run by HPE Ezmeral Runtime Enterprise. You can implement this access control by creating an allowed list of `sudo` commands that HPE Ezmeral Runtime Enterprise runs.

The lists of `sudo` commands provided in this topic are formatted for ease of copying and pasting.

Set the `NOPASSWD` tag to ensure all bin files execute successfully.

Installing and Upgrading HPE Ezmeral Runtime Enterprise 5.6.x

The following `sudo` privileges are required for installing and upgrading HPE Ezmeral Runtime Enterprise 5.6.x:

```
/bin/base64 /bin/bdconfig /bin/cat /bin/chcon /bin/chgrp /bin/chmod /bin/
chown /bin/container-storage-setup /bin/cp /bin/dd /bin/echo /bin/find /bin/
getent /bin/grep /bin/hostnamectl /bin/id /bin/killall /bin/ln /bin/ls /bin/
mkdir /bin/mount /bin/ovs-ofctl /bin/ovs-vsctl /bin/mv /bin/pkill /bin/
python3 /bin/rm /bin/rpm /bin/sed /bin/sg /bin/
systemctl /bin/tar /bin/tee /bin/test /bin/touch /bin/umount /bin/which /bin/
xargs /bin/yum /opt/bluedata/common-install/scripts/
generate_django_secret.py /opt/bluedata/common-install/scripts/monitoring/
services_config/tls/generate-certs.sh /sbin/alternatives /sbin/blkid /sbin/
blockdev /sbin/chpasswd /sbin/corosync-cmapctl /sbin/dmidecode /sbin/dmsetup /
```

```
sbin/groupadd /sbin/groupdel /sbin/ip /sbin/iptables /sbin/lvcreate /
sbin/lvs /sbin/mkfs /sbin/parted /sbin/pcs /sbin/pvcreate /sbin/pvremove /
sbin/restorecon /sbin/semodule /sbin/semanage /sbin/service /sbin/setsebool /
sbin/ss /sbin/subscription-manager /sbin/sysctl /sbin/useradd /sbin/userdel /
sbin/usermod /sbin/vgcreate /sbin/vgdisplay /sbin/vgremove /usr/bin/
firewall-cmd /usr/sbin/dmidecode /usr/sbin/pcs /usr/sbin/haproxy /bin/ls /
sbin/sysctl /sbin/vgscan /sbin/lvscan /sbin/pvscan /bin/egrep /bin/
nerdctl /usr/bin/ezconfig /bin/ctr
```

Running HPE Ezmeral Runtime Enterprise 5.6.x

The following sudo privileges are required for running HPE Ezmeral Runtime Enterprise 5.6.x:

```
/bin/systemctl , /bin/sed , /bin/cat , /bin/rm , /bin/mkdir , /bin/
chgrp , /bin/chmod , /bin/chown , /bin/cp , /sbin/ip , /bin/ovs-ofctl , /bin/
killall , /usr/sbin/dnsmasq , /usr/sbin/haproxy , /bin/echo , /
sbin/ip , /bin/stat , /bin/umount , /bin/mount , /usr/sbin/crm_mon , /usr/
sbin/pcs , /bin/ovs-vsctl , /usr/sbin/haproxy , /sbin/vgdisplay , /
sbin/dmidecode , /bin/sed , /bin/umount , /bin/stat , /bin/mount , /bin/
mkdir , /bin/chgrp , /bin/chmod , /usr/sbin/pcs , /usr/sbin/crm_mon /sbin/
iptables /bin/nerdctl /bin/find /bin/ls /bin/xargs /bin/tar /bin/test /sbin/
modprobe /bin/mv /sbin/restorecon /sbin/sysctl /bin/yum /bin/tee /bin/chcon /
sbin/semanage /bin/ezctl /usr/bin/ezctl bin/pkill /bin/
timeout /bin/ctr /usr/bin/ezconfig /bin/containerd /sbin/lvs /sbin/lvremove /
sbin/vgreduce /sbin/pvremov /sbin/parted /sbin/blockdev /sbin/vgremove /sbin/
vgscan /sbin/lvscan /sbin/pvscan
```

SETENV Sudo Tag

You also need to set the SETENV sudo tag for the following commands:

```
/bin/cat
/usr/sbin/haproxy
```

Kubernetes Requirements

The topics in this section describe the requirements for deploying HPE Ezmeral Runtime Enterprise on Kubernetes. Depending on the features and applications you include in your deployment, additional requirements might apply.

Kubernetes Version Requirements

The supported versions of Kubernetes vary by host operating system, and are different depending on whether the cluster was created using HPE Ezmeral Runtime Enterprise or was imported into HPE Ezmeral Runtime Enterprise. See Support Matrixes on page 54.

For additional HPE Ezmeral Data Fabric on Kubernetes requirements, see Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.

Kubernetes Controller Requirements

Kubernetes clusters and tenants use the same Controller host (and Shadow Controller/Arbiter, if platform High Availability has been enabled; see High Availability.

A Controller host must meet the requirements described in Host Requirements, Operating System Requirements, and Configuration Requirements.

The Controller host:

- Serves as the control plane.

- Manages the Kubernetes hosts
- Manages the Embedded Data Fabric, if present.
- Manages DataTaps, FS Mounts, etc.

See Kubernetes Physical Architecture for more information.

Kubernetes Gateway Requirements

Hosts that will be used as Gateways (see Gateway Hosts) for Kubernetes must conform to the requirements listed in Host Requirements, Operating System Requirements, and Configuration Requirements. For Kubernetes clusters, Gateway hosts expose the KubeAPI server and NodePort services within each Kubernetes cluster.

Kubernetes Host/Node Requirements

Worker Hosts

Hosts that will be used for Kubernetes can only be used for Kubernetes clusters.

All Kubernetes hosts must conform to the requirements listed in the following:

- Host Requirements
- Operating System Requirements
- Configuration Requirements
- If you are deploying **HPE Ezmeral Data Fabric on Kubernetes**, see Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.

Furthermore, if you will be running any add-ons, then the Kubernetes hosts must also comply with the cumulative requirements for all of the add-ons you will be running. See Add-ons Overview for a list of current add-ons and links to additional details. For example:

- If you will not be running any add-ons, then the Kubernetes hosts need only comply with the base requirements.
- If you will be running `Add-on_1`, then the Kubernetes hosts must comply with the base requirements plus the requirements for `Add-on_1`.
- If you will be running `Add-on_2`, then the Kubernetes hosts must comply with the base requirements plus the requirements for `Add-on_2`.
- If you will be running `Add-on_1` and `Add-on_3`, then the Kubernetes hosts must comply with the base requirements plus the requirements for `Add-on_1` plus the requirements for `Add-on_3`.
- If you will be running `Add-on_1` and `Add-on_2` and `Add-on_3`, then the Kubernetes hosts must comply with the base requirements plus the requirements for `Add-on_1` plus the requirements for `Add-on_2` plus the requirements for `Add-on_3`.

If you experience issues with traffic being routed incorrectly because `iptables` are being bypassed, then ensure that `net.bridge.bridge-nf-call-iptables` is set to `1` in your `sysctl config`. For example:

```
cat <<EOF > /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
```

```
EOF
sysctl --system
```

Make sure that the `br_netfilter` module is loaded before this step. You can do this by executing the command `lsmod | grep br_netfilter`. To load it explicitly, call `modprobe br_netfilter`.

All Kubernetes host clocks must be synchronized with the HPE Ezmeral Runtime Enterprise clocks.

The HPE online sizer for HPE Ezmeral Runtime Enterprise can provide additional guidance for sizing your deployment. See: HPE Sizing Tool for Ezmeral Container Platform (link opens in a new browser tab/window).

Data Fabric Nodes

Each Data Fabric node must meet the minimum requirements described in Requirements for HPE Ezmeral Data Fabric on Kubernetes (for non-production environments only) on page 595.

Disk and Network Requirements on Hosts Running etcd Service

Etcd is an integral part of the Kubernetes control plane, and requires low network latency and a sustained high performance disk to run.

Hewlett Packard Enterprise recommends running an etcd performance benchmark on your host to make sure it meets minimum network and disk I/O requirements: etcd benchmark tools (link opens an external site in a new browser tab or window).

Kubernetes Docker Hub Requirements

Hewlett Packard Enterprise recommends using a Docker Hub account. To create a Docker Hub account, see Docker Hub accounts.

Kubernetes clusters running on any version of Hewlett Packard Enterprise can occasionally encounter problems caused by the pull rate limit that Docker Hub applies to all free and anonymous accounts. These limits can cause cluster creation and application deployment to fail. If Kubernetes pods in a non-air-gap environment are failing to come into Ready state and are showing ImagePullBackoff or related errors, this is the most likely cause.

If you are using public repository and not Docker Hub account, you may face the pull rate limit issue. For possible workarounds see EZESC-232 in Issues and Workarounds on page 15.

Kubernetes Air-Gap Requirements

Kubernetes containers created within HPE Ezmeral Runtime Enterprise will automatically conform to these requirements. Kubernetes containers that were not created within HPE Ezmeral Runtime Enterprise must be brought into compliance with these requirements before you can use them within the deployment.

If you will be using an air-gap configuration for Kubernetes objects, then you must configure air-gap settings before adding any Kubernetes hosts.

CAUTION:

Apply all air-gap settings with care. These settings do not propagate if updated after Kubernetes hosts have been installed, unless one of the following occurs:

- The Kubernetes host is rebooted.
- The version of Kubernetes running on a host is upgraded.

Any Kubernetes hosts in a ready state that are not part of a Kubernetes cluster must be restarted for the changes to be applied.

Python

Python version 2.7 or 3.6 and above is required to run the air gap utility script. This script is used to query, filter, and download the container images necessary for your air gap environment. For information on using this script, see Using the Air Gap Utility on page 869.

RHEL and CentOS

If you use a Docker client certificate to secure the container registry, you must use a 4096-bit RSA key.

If the certificate uses a different RSA key length, such as 2048 bits, when you attempt to add a Kubernetes host, the following occurs:

- A storage error is returned.
- An error message similar to the following is added to the `bds-mgmt.log`:

```
dictionary update sequence element #3 has length 4; 2 is required
```

Viewing Docker Images By License

To find the Docker images required to run applications provided with HPE Ezmeral Runtime Enterprise in an air-gapped environment, use the Air Gap Utility. See Using the Air Gap Utility on page 869.

Within the Air Gap Utility, use the following commands to view Docker images by license:

- **Standard (non-ML Ops)**:
  - HPE Ezmeral Runtime Enterprise:

    ```
    hpe-airgap-util --release <release-number> --license enterprise
    ```

  - HPE Ezmeral Runtime Enterprise Essentials:

    ```
    hpe-airgap-util --release <release-number> --license essential
    ```

- **ML Ops**:

  ```
  hpe-airgap-util --release <release-number> --license mlops
  ```

- **Analytics for Apache Spark**:

  ```
  hpe-airgap-util --release <release-number> --license analytics
  ```

Related reference

HPE Ezmeral Runtime Enterprise Air Gap Utility Release Notes on page 53
Change history and version compatibility information for the HPE Ezmeral Runtime Enterprise Air Gap Utility, `hpe-airgap-util`, on HPE Ezmeral Runtime Enterprise.

Air Gap Tab on page 799
The **Air Gap** tab of the **System Settings** screen enables the Platform Administrator to specify settings to be used when the Kubernetes hosts, clusters, and tenants do not have connectivity to the Internet.

More information

Using the Air Gap Utility on page 869
Describes how to use the air gap utility to download files in an air-gapped HPE Ezmeral Runtime Enterprise environment.

Kubernetes Port Requirements

The following ports must be available when running Kubernetes inside HPE Ezmeral Runtime Enterprise:

Inbound

- **22 (TCP):** Remote access over SSH.
- **80 (TCP):** Load balancer/proxy that does external SSL termination, and HTTP ingress.
- **443 (TCP):** Virtual nodes and sources that require the HPE Container Platform interface or API, and HTTPS ingress.
- **2379 (TCP):** etcd client requests.
- **2380 (TCP):** etcd peer communication.
- **6443 (TCP):** Kubernetes API Server.
- **8472 (UDP):** Canal/Flannel VXLAN overlay networking.
- **9099 (TCP):** Canal/Flannel livenessProbe/readinessProbe.
- **10250 (TCP):** Kubelet.
- **30000-32767 (TCP/UDP):** NodePort port range.

Outbound

- **22 (TCP):** SSH node provisioning
- **443 (TCP):** Catalogs and agent.
- **2376 (TCP):** Docker daemon TLS port.
- **2379 (TCP):** etcd client requests.
- **2380 (TCP):** etcd peer communication.
- **6443 (TCP):** Kubernetes API server.
- **8472 (UDP):** Canal/Flannel VXLAN overlay networking.
- **9099 (TCP):** Canal/Flannel livenessProbe/readinessProbe
- **10250 (TCP):** Kubelet.
- **10254 (TCP):** Ingress controller livenessProbe/readinessProbe.

More information

Port Requirements on page 809

HPE Ezmeral ML Ops Requirements

HPE Ezmeral ML Ops has the same requirements as HPE Ezmeral Runtime Enterprise for the following:

- Browser
- Host
- Operating system

- Configuration
- Network
- Web proxy
- Restricted sudo

In addition, the following requirements must also be met:

- LDAP must be configured in order to run HPE Ezmeral ML Ops in a Kubernetes cluster.

  All AI/ML project users (Project Members and Project Administrators) must be LDAP/AD users. They cannot be authenticated using local authentication.

- Tenant storage must be configured and registered on the HPE Ezmeral Runtime Enterprise deployment.

  HPE Ezmeral Runtime Enterprise supports implementations of HPE Ezmeral Data Fabric as tenant storage. See HPE Ezmeral Data Fabric as Tenant/Persistent Storage on page 579.

The HPE online sizer for HPE Ezmeral Runtime Enterprise can provide additional guidance for sizing deployments for ML Ops workloads. See: HPE Sizing Tool for Ezmeral Container Platform (link opens a different HPE site in a new browser tab/window).

Deploying the Platform

The topics in this section describe deploying HPE Ezmeral Runtime Enterprise. Deployment is divided into phases.

Installation Overview

The general installation process is as follows:

1. Plan your installation, as described in Planning Overview and subsequent articles.
2. Verify that the resources being used meet all of the following requirements and recommendations:
   - Browser Requirements
   - Host Requirements (be sure to see the Sizing Considerations, which are recommendations to ensure that you are allocating the appropriate resources to the deployment).
   - Operating System Requirements
   - Network Requirements
   - Configuration Requirements
   - Restricted Sudo Requirements
   - If you will be using an air-gap configuration, in which Kubernetes hosts, clusters, and tenants do not have connections to the Internet, see Kubernetes Air-Gap Requirements on page 834.
3. Install the operating system on the host(s) you will be using.
4. If any of the host(s) include GPUs, then install the GPU drivers as described in GPU Driver Installation.
5. Determine which installer bundle you need, as described in Bundles.
6. Obtain the installer bundle, as directed by Hewlett Packard Enterprise.

7. If needed, add an SSL certificate to permit access to the web interface via https:// instead of unsecured http://, as described in Adding an SSL Certificate.

8. Use the pre-check script to verify that the machine that will become the Controller host is ready to receive the installation. See Using the Pre-Check Script.

9. If you are upgrading to HPE Ezmeral Runtime Enterprise, follow the instructions for upgrading the platform.

10. Determine how you will install HPE Ezmeral Runtime Enterprise on the Controller host from the command line interface. Your available options are:
    - Standard Installation
    - Using the Pre-Check Config File

11. After the installation bundle completes, access the web interface to continue setup, as described in Platform Controller Setup.

12. Validate the installation as described in Validating the Installation. This will probably involve installing a Worker host (see Kubernetes Worker Installation Overview) and may involve installing a Gateway host (see Gateway Installation Tab).

13. The deployment automatically installs the HPE Ezmeral Instant-On evaluation license for an unlimited number of CPU cores. The HPE Ezmeral Instant-On is valid for 30 days from the installation date. Before the license expires, add a purchased license as described in Licensing.

    CAUTION:

    If the HPE Ezmeral Instant-On and all other evaluation licenses expire before before a purchased license has been applied, then the deployment will go into Lockdown mode (see Lockdown Mode on page 916). The Platform Administrator will not be able to exit Lockdown mode until a purchased license is applied.

You are now ready to begin using HPE Ezmeral Runtime Enterprise!

GPU Driver Installation

Download NVIDIA GPU drivers from NVIDIA, install them on hosts, and test the installation after the hosts have been added to HPE Ezmeral Runtime Enterprise.

IMPORTANT:

The host OS NVIDIA driver must be compatible with the NVIDIA driver included in your application image (such as, for example, TensorFlow in the HPE Ezmeral ML Ops Training image).

For MIG support on GPUs, the driver must also support MIG.

See GPU and MIG Support on page 721.

If possible, install the NVIDIA GPU drivers before adding the host to HPE Ezmeral Runtime Enterprise.

If you update the OS Kernel on a host, you must reinstall the NVIDIA GPU drivers on that host (see Steps 8-11).

If you want add GPUs or update GPU drivers after a host has been added to HPE Ezmeral Runtime Enterprise, do the following:

- If the host is a Kubernetes host, remove the Worker from the Kubernetes cluster and then remove the host from HPE Ezmeral Runtime Enterprise.

Installing or Updating the GPU Driver on RHEL and CentOS Hosts

You must perform the following procedure on each RHEL or CentOS host that will supply GPU devices to the deployment.

1. Install the GPU devices in the host.

2. Locate the appropriate GPU device driver and libraries package on the NVIDIA Downloads Index (link opens an external website in a new browser tab or window), and then download it to each GPU-providing host. You will use the downloaded file in both Steps 3 and 7 of this procedure.

   IMPORTANT:

   Select **Linux 64-bit** (not **Linux 64-bit RHEL6** or similar) to obtain a runfile. Selecting a specific Linux distribution will download an RPM, which will not work with your HPE Ezmeral Runtime Enterprise deployment.

3. If you are performing the initial driver installation, execute the following commands as the root user (you should not need to perform this step when upgrading existing drivers):

   ```
   yum update -y
   yum install -y kernel-devel kernel-headers gcc-c++ perl pciutils
   yum install -y kernel-devel-$(uname -r) kernel-headers-$(uname -r)
   ```

4. If any packages get updated through yum update (in previous step), ensure that GPU driver is still working (You can execute the command `nvidia-smi` to ensure the expected results). If the GPU driver is not working, reinstall the driver (see Steps 8-11).

5. **NOTE:**

   This step reboots the host. If HPE Ezmeral Runtime Enterprise is already installed on this host, and if virtual nodes/containers are assigned to this host, then this step will briefly interrupt those nodes/containers.

   If you are performing the initial driver installation, execute the following commands as the root user. (You should not need to perform this step when upgrading existing drivers.)

   ```
   cat > /etc/modprobe.d/blacklist-nouveau.conf <<EOF
   blacklist nouveau
   options nouveau modeset=0
   EOF
   rmmod nouveau
   dracut --force
   reboot
   ```

6. After reboot, verify that the nouveau module is not loaded by executing the command `lsmod | grep nouveau`.

   If nouveau is still loaded, then repeat Steps 5 and 6.

7. Install or upgrade the host GPU driver by executing the following commands as a root user:

   ```
   cd /nvidia
   chmod +x ./NVIDIA-Linux-*.run
   ./NVIDIA-Linux-*.run -s
   ```

8. Execute the command `nvidia-smi` to query the available GPU devices on the host, to verify successful installation.

9. Execute the command `nvidia-modprobe -u -c=0`.

   This command is needed to probe the `nvidia-uvm` kernel module, which is necessary in order for HPE Ezmeral Runtime Enterprise to recognize the host as having GPUs.

10. Reboot the Worker host.

Installing GPU Drivers on SLES Hosts

You must perform the following procedure on each SLES host that will supply GPU devices to the deployment.

1. Install the GPU devices in the host.

2. Determine which version of the NVIDIA GPU drivers to install.

   The host OS NVIDIA driver must be compatible with the NVIDIA driver included in the application image. See GPU and MIG Support on page 721.

3. Locate the appropriate GPU device driver and libraries package on the NVIDIA Downloads Index (link opens an external website in a new browser tab or window), and then download it to each GPU-providing host.

4. Install the drivers and CUDA packages as appropriate for this operating system. See NVIDIA Driver Installation Quickstart Guide (link opens an external website in a new browser tab or window).

5. To verify successful installation, execute the command to query the available GPU devices on the host:

   ```
   nvidia-smi
   ```

6. Set permissions to enable read and write access to the GPU devices by tenant users:

   a. In the `/etc/modprobe.d/50-nvidia-default.conf` file, change the entry `NVreg_DeviceFileMode=0660` to the following:

      ```
      NVreg_DeviceFileMode=0666
      ```

   b. Reboot the host.

   c. Verify that the permissions are read and write for all users of the GPU devices:

      ```
      ls -al /dev/nvidia*
      crw-rw-rw-. 1 root video 195,   0 Jun 21 11:21 /dev/nvidia0
      crw-rw-rw-. 1 root video 195, 255 Jun 21 11:21 /dev/nvidiactl
      ...
      ```

Testing the Installation (Kubernetes Pods)

To test GPU installation in Kubernetes, see Using GPUs in Kubernetes Pods.

Deploying MIG Support

This topic describes how to configure and deploy a supported MIG-enabled GPU on HPE Ezmeral Runtime Enterprise.

You must configure MIG before adding the host to HPE Ezmeral Runtime Enterprise. If the host has not yet been added to HPE Ezmeral Runtime Enterprise, see Host Has Not Been Added to HPE Ezmeral Runtime Enterprise on page 841.

If the host has already been added to HPE Ezmeral Runtime Enterprise, see Host Already Added to HPE Ezmeral Runtime Enterprise on page 841.

Required access rights: Platform Administrator

Host Has Not Been Added to HPE Ezmeral Runtime Enterprise

If the host has not yet been added to HPE Ezmeral Runtime Enterprise, you install the driver, enable and configure MIG. Then you can add the host to HPE Ezmeral Runtime Enterprise and to the Kubernetes cluster.

1. Install NVIDIA driver version 470.57.02 or later. To install the driver, see GPU Driver Installation on page 838.
2. Use the `nvidia-smi` tool to configure and enable MIG. See MIG Configuration Using nvidia-smi on page 841.
3. Add the host to HPE Ezmeral Runtime Enterprise and to the Kubernetes cluster as a Kubernetes Worker. See Kubernetes Worker Installation Overview on page 528.

Host Already Added to HPE Ezmeral Runtime Enterprise

Use the following procedure if you are adding GPU or MIG GPU support to a host and you are not performing this task as part of HPE Ezmeral Runtime Enterprise.

If you are upgrading from an earlier version of HPE Ezmeral Runtime Enterprise, you remove the host from the Kubernetes cluster and from HPE Ezmeral Runtime Enterprise as part of the upgrade procedure.

1. Remove the host from the Kubernetes cluster. See Expanding or Shrinking a Kubernetes Cluster on page 483.
2. Delete the host from HPE Ezmeral Runtime Enterprise. See Decommissioning/Deleting a Kubernetes Host on page 555.
3. Ensure that the NVIDIA driver is driver version 470.57.02 or later. Update the driver on the host as needed. See GPU Driver Installation on page 838.
4. Use the `nvidia-smi` tool to configure and enable MIG. See MIG Configuration Using nvidia-smi on page 841.
5. Add the host to HPE Ezmeral Runtime Enterprise a Kubernetes Worker. See Kubernetes Worker Installation Overview on page 528.

MIG Configuration Using `nvidia-smi`

You use the NVIDIA `nvidia-smi` command-line interface to configure, enable, and manage MIG.

See the following NVIDIA documentation (links open an external website in a new browser tab or window):

- MIG Support on Kubernetes
- NVIDIA Multi-Instance GPU User Guide

IMPORTANT:

As stated in the NVIDIA documentation, to run CUDA workloads on the GPU, you must create both MIG GPU instances and their corresponding compute instances. However, the created MIG devices are not persistent across system reboots or if the GPU is reset.

The HPE Ezmeral Runtime Enterprise `bds-nvidia-mig-config` service preserves the MIG device configurations across system reboots, so no additional configuration or mitigation is required.

HPE Ezmeral Runtime Enterprise supports the following Kubernetes strategies for MIG deployment:

Single	The `single` strategy enables you to interact with MIG instances in the same way you interact with physical GPUs. All MIG devices on a node have the same MIG configuration, such as `MIG 1g.5gb`. All MIG and physical GPU devices are enumerated using the same resource type: `nvidia.com/gpu`. For example: `--limits=nvidia.com/gpu=1`
Mixed	MIG devices on a node can have different configurations. Each MIG configuration in the cluster is identified by a resource type, in the form `<slice_count>g.<memory_size>gb`. You specify and enumerate MIG devices by their fully qualified name in the form: `nvidia.com/mig-<slice_count>g.<memory_size>gb` For example: • `--limits=nvidia.com/mig-1g.5gb=1` • `--limits=nvidia.com/mig-3g.20gb=2` The mixed strategy supports nodes that include GPUs that do not support MIG. GPU devices that do not support MIG are enumerated using the resource type: `nvidia.com/gpu`.

Related concepts

Troubleshooting MIG on HPE Ezmeral Runtime Enterprise on page 731
Troubleshooting tips for verifying MIG installation and configuration in Kubernetes deployments of HPE Ezmeral Runtime Enterprise.

Related reference

GPU and MIG Support on page 721
This topic provides information about support for NVIDIA GPU and MIG devices on HPE Ezmeral Runtime Enterprise.

Phase 1

The topics in this section describe Phase 1 of deploying HPE Ezmeral Runtime Enterprise.

Bundles

Software bundles are available for the supported operating systems. These bundles will have a name such as `hpe-cp-<os>-<version>-<build>.bin`, where:

- `<os>` is the operating system (for example `rhel`)
- `<version>` is the version of HPE Ezmeral Runtime Enterprise, such as `5.6`.
- `<build>` is the specific build number, such as `3100`.

Phase 2

The topics in this section describe Phase 2 of deploying HPE Ezmeral Runtime Enterprise.

Adding an SSL Certificate

If you want to access the HPE Ezmeral Runtime Enterprise web interface and the Kubernetes web terminal using Secure Sockets Layer (https://), then you will need to add an SSL certificate on the machine that will become the Controller host before running the pre-check script. If you do this, then be sure to record the location and name of the certificate. From there, be sure to use the `--ssl-cert` and `ssl-priv-key` options when running the pre-check script. See Using the Pre-Check Script.

If you do not perform this step during installation, then you can enable SSL connections post-installation by following the instructions in Enabling SSL Connections.

Using the Pre-Check Script

The pre-check script performs a series of checks on the Controller host to determine whether it is ready to accept the installation. To use the script:

1. Download the `hpe-cp-prechecks-<version>.bin` script, where `<version>` is the version number, such as `5.6`.
   - **For RHEL/CentOS (5.6.0):**
     - NA download link: 5.6.0 RHEL/CentOS Pre-check script
     - AP download link: 5.6.0 RHEL/CentOS Pre-check script
   - **For SLES (5.6.0):**
     - NA download link: 5.6.0 SLES Pre-check script
     - AP download link: 5.6.0 SLES Pre-check script
2. If needed, copy the .bin file to a directory on the machine that will become the Controller host.
3. Make the .bin file executable by executing the command:

   ```
   chmod a+x hpe-cp-prechecks-<version>.bin
   ```

4. Run the executable binary using the format:

   ```
   hpe-cp-prechecks-<version>.bin <options>
   ```

   where `<options>` denotes the options and parameters you need to pass to the script. See Script Options on page 844 for a complete list of options and when to use them, and Examples on page 846 for examples of usage scenarios.

5. After running the script, see Sample Pre-Check Output on page 847 for a complete example of a successful pre-check as well as links to common errors and how to resolve them. In addition to the displayed output, the pre-check script generates several files that are described in Pre-Check Generated Files on page 852. If needed, remediate any issues and then re-run the pre-check script until all tests pass or until you have accounted for any warnings.

   CAUTION:

   Do not use the config file to install the controller in a production environment if the pre-check output lists one or more errors.

   For non-production environments, when you want to create a deployment that is smaller than the requirements for production environments, specify the `--force` option when executing the script.

6. After you are satisfied with the results of the pre-check, you may proceed to install the Controller host using either of the following methods:

   - Configuration file generated by the pre-check script, as described in Using the Pre-Check Config File on page 853. The name of the configuration file is the following:

     ```
     /tmp/bd_prechecks.conf
     ```

   - Manually, as described in Standard Installation on page 854.

NOTE: Hewlett Packard Enterprise recommends using the config file generated by the pre-check script when installing HPE Ezmeral Runtime Enterprise on the Controller host as this method will most likely be faster and easier than manual installation.

Script Options

This table lists all of the Controller-specific options that can be used with the pre-check script. Not all of these options will apply in all situations. See Examples on page 846 for exmaples of how this script can be used in a variety of real-world scenarios.

Option	Short Option	Description	Notes	Example
`--controller`	`-c`	Verify host meets Controller requirements.	Use this option on the Controller host only.	
`--int-gatewa y-ip`		Internal gateway IP address for virtual nodes/containers. This address cannot be used by another resource on the corporate network. See Network Requirements.	*Optional*. If not specified, default value is 172.16.13.1	`--int-gatewa y-ip 172.1.1.1`

`--proxy`	`-p`	Proxy URL in the format
		`[protocol://]`
		`[username:password@]`
		`proxy_address[:port]`
			Skip this option if no web proxy is being used for EPIC. See Web Proxy Requirements.	`--proxy https://admin:admin123@172.1.1.2:8080`

`--controller-automount-root`		Auto-mount root on the Controller.		`--controller-automount-root <path>`
`--config-file-path`		Path where the configuration file (see Sample Pre-Check Output) will be written.		`--config-file-path <path>`
`SUDO_PREFIX`		Specify the actual sudo prefix to use when running privileged commands as a non-root user.	Defaults to `sudo -n` if no other option provided.	`export SUDO_PREFIX="usr/sbin/dzdo/dzdo"`
`--ssl-cert`		SSL certificate to use for secure (https://) access to the web interface.	Do not use this option if you will be using non-secured (http://) access to the web interface.	`--ssl-cert /root/bds-https.cert`
`--ssl-ca-data`		Provide the CA authentication chain data required for having an SSL client authenticate the server certificate.	This must be an absolute file path that will be readable by the httpd process. The “CA data” file is used for an `openssl verify -CAfile` command to ensure that an SSL client (such as those used in our k8s support) can validate the certificate.	`--ssl-ca-data <path>`
`--gateway-ssl-cert`		Provide the public-key SSL certificate for SSL termination in the gateway. If this is not provided, but the server certificate and private key are, these values will now default from those server certificate values (`--ssl-cert` and `--ssl-priv-key`).	This must be an absolute path.	`--gateway-ssl-cert <path>`

`--gateway-priv-key`		Provide the private key corresponding to the above.	This also must be an absolute path if given.	`--gateway-priv-key <path>`
`--ssl-priv-key`		SSL private key to use for secure (https://) access to the web interface.	Do not use this option if you will be using non-secured (http://) access to the web interface.	`--ssl-priv-key /root/bds-https.pem`
`--worker-agent-install`		Worker hosts must be initialized manually before installing them via the web interface.	*ONLY USE THIS OPTIONS WHEN YOUR ENVIRONMENT DOES NOT ALLOW KEY-BASED SSH. SEE Configuration Requirements.*	
`--force`	`-f`	Force pre-check validation to succeed regardless of any errors.	*THIS OPTION IS FOR ADVANCED USERS ONLY AND MAY RESULT IN AN UNUSABLE DEPLOYMENT.*	
`--dnsmasq-user`		Optional. Specifies the user that `dnsmasq` service will run under.	If not specified, then the `dnsmasq` service will run under user `nobody:nobody`.	`--dnsmasq-user dnsmasquser`
`--dnsmasq-grp`		Optional. Specifies the group of `dnsmasq` users.	=	`--dnsmasq-grp dnsmasqgroup`

Examples

This section presents some examples of using the pre-check script.

- **Root/Agent:** This example pre-checks HPE Ezmeral Runtime Enterprise as the root user and includes the Worker agent because password-less SSH is not available in the environment.

```
# root@localhost> /root/hpe-cp-prechecks-5.1.bin --worker-agent-install
```

- **Root/non-Agent:** This example pre-checks HPE Ezmeral Runtime Enterprise as the root user. The environment does allow password-less SSH, and thus the `--worker-agent-install` option is *not* used.

```
# root@localhost> /root/hpe-cp-prechecks-5.1.bin
```

- **Root/Agent/SSL:** This example pre-checks HPE Ezmeral Runtime Enterprise as the root user and includes the Worker agent because password-less SSH is not available in the environment. It also provides SSL information to enable secure (https://) access to the HPE Ezmeral Runtime Enterprise interface.

```
# root@localhost> /
root/hpe-cp-prechecks-5.1.bin --worker-agent-install --ssl-cert /root/
bdhost.cert --ssl-priv-key /root/bdhost.pem
```

- **Root/Non-Agent/SSL:** This example pre-checks HPE Ezmeral Runtime Enterprise as the root user. The environment does allow password-less SSH, and thus the `--worker-agent-install` option is *not* used. This example also provides SSL information to enable secure (https://) access to the HPE Ezmeral Runtime Enterprise interface.

```
# root@localhost> /home/epic/hpe-cp-prechecks-5.1.bin --ssl-cert /root/
bdhost.cert --ssl-priv-key /root/bdhost.pem
```

- **Non-root/Agent:** This example pre-checks HPE Ezmeral Runtime Enterprise as a non-root user and includes the Worker agent because password-less SSH is not available in the environment.

```
# epic@localhost> /home/epic/
hpe-cp-prechecks-5.1.bin --worker-agent-install
```

- **Non-root/non-Agent:** This example pre-checks HPE Ezmeral Runtime Enterprise as a non-root user. The environment does allow password-less SSH, and thus the `--worker-agent-install` option is *not* used

```
# epic@localhost> /home/epic/hpe-cp-prechecks-5.1.bin
```

- **Non-root/Agent/SSL:** This example pre-checks HPE Ezmeral Runtime Enterprise as a non-root user and includes the Worker agent because password-less SSH is not available in the environment. It also provides SSL information to enable secure (https://) access to the web interface.

```
# epic@localhost> /home/epic/
hpe-cp-prechecks-5.1.bin --worker-agent-install --ssl-cert /home/epic/
bdhost.cert --ssl-priv-key /home/epic/bdhost.pem
```

- **Non-root/non-Agent/SSL:** This example pre-checks HPE Ezmeral Runtime Enterprise as a non-root user. The environment does allow password-less SSH, and thus the --worker-agent-install option is *not* used. This example also provides SSL information to enable secure (https://) access to the web interface.

```
# epic@localhost> /home/epic/hpe-cp-prechecks-5.1.bin --ssl-cert /home/
epic/bdhost.cert --ssl-priv-key /home/epic/bdhost.pem
```

NOTE: Be sure to run both the precheck script and installer as the user who will be installing HPE Ezmeral Runtime Enterprise.

Sample Pre-Check Output

The pre-check script will return output that is similar to that shown in the following table. The **Error Resolution** column of the table lists the most common errors encountered by each check, along with diagnosis/remediation instructions. The pre-check script will also generate the files described in Pre-Check Generated Files on page 852.

NOTE: This table displays all of the tests that could be run assuming that all of the options described in Using the Pre-Check Script on page 843 are used. Your output will probably not contain all of the information presented below; this is normal behavior.

NOTE: The totals displayed on this page will vary by operating system and configuration.

Option	Expected Result	Error Resolution
`Hardware properties`		
`Checking CPU count:`	`PASSED`	The host must have at least eight (8) physical CPU cores. There is no other way to remediate this issue.
`Checking memory capacity:`	`PASSED`	The host must have least 64GB of RAM. There is no other way to remediate this issue.
`Checking number of disks available:`	`PASSED`	Raw disks with no filesystem, logical volumes, or partitions are required. Use the `lsblk` command to verify the state of the disks before proceeding. If needed, use the `--force` option to install on non-raw disks.
`Total: 3 -- Failed: 0 -- Warning: 0 -- Forced(success): 0`		
`Network configuration`		
`Network port availability`	`PASSED`	The script checks a number of ports. See Port Requirements and Kubernetes Port Requirements for the ports that must be available for the deployment.
`Checking primary IP address: Using eth<number>:<ip_address> as primary interface`	`PASSED`	You may see errors if there is more than one network interface (NIC) with an assigned IP address. To resolve this, ensure that there is only one NIC with an assigned IP address. If this is not possible, add the option `--controller-public-if` to specify the NIC to use.
`Checking FQDN of the host:`	`PASSED`	The hostname must be a Fully Qualified Domain Name (FQDN) that includes at least one "dot" (.) To fix this problem, specify a valid FQDN using the `hostnamectl set-hostname` command. There is no need to reboot the Controller host; however, you will need to open a new SSH session by logging out and back in for the change take to effect.
`Checking default gateway settings:`	`PASSED`	Use the command `route -n` to verify that there is only one internal gateway, which is indicated with the notation **`UG`** in the **Flags** column of the table returned by this command.
`Checking Proxy settings`	`PASSED`	Checks for environment and shell variables.

Option	Expected Result	Error Resolution
`Checking internet access:`	`PASSED`	To fix internet access problems, ensure that the proxy URL in the `--proxy` option is properly formatted. If needed, use the `--force` option to run the installer without specifying a proxy.
`Total: 5 -- Failed: 0 -- Warning: 0 -- Forced(success): 0`		
`Operating system configuration`		
`Checking OS type:`	`PASSED`	See Operating System Requirements.
`Checking OS Language:`	`PASSED`	The system locale must be set to US English with UTF-8 encoding. See Configuration Requirements on page 826.
`Checking OS Family:`	`PASSED`	See Operating System Requirements.
`Checking RHEL subscription:`	`PASSED`	*This test is not performed on CentOS or SLES systems, and the **Total** at the bottom of this section will read 11 instead of 12.* If the host is running RHEL, then a valid subscription must exist to the server RPM, the server optional RPM and the HA for server RPM channels. For example for RHEL 7, you must subscribe to the following: `rhel-7-server-rpms`, `rhel-7-server-optional-rpms`, and `rhel-ha-for-rhel-7-server-rpms`. If needed, use the `--force` option if you have either a satellite server or access to RHEL OS rpms by any other means.
`Checking running kernel version:`	`PASSED`	The kernel version must greater or equal to the minimum version listed in OS Support on page 85. If needed, upgrade the kernel and then reboot the host.
`Checking CONFIG_SECCOMP enabled in kernel`	PASSED	rhel
`Checking compatible DOCKER version`	PASSED	rhel

Option	Expected Result	Error Resolution
`Checking SELinux enforcement:`	`PASSED`	**SELinux is disabled** does not affect HPE Ezmeral Runtime Enterprise functionality. However, if installation occurs with SELinux disabled, then you must leave SELinux disabled in order for the deployment to continue to function. Do not alter the SELinux configuration after installation. See Configuration Requirements.
`Checking SELINUX policy:`	`targeted`	SLES only.
`Checking IPtables/ Firewalld configuration:`	`PASSED`	**IPtables is disabled** is a warning only. Do not change this setting after installing HPE Ezmeral Runtime Enterprise.
`Checking automount configuration:`	`PASSED`	Ensure that `/etc/auto.master` has only one `-hosts` line. If this line exists, it be the same on the Controller and all Worker hosts.
`Checking SSHD configuration:`	`PASSED`	*This test is not performed on RHEL systems.* • **SSHD not found:** Install both `openssh-server` and `openssh-client`. • **SSHD must be configured at boot:** Ensure that SSHD starts at bootup using chkconfig. • **SSHD must allow root login:** If installing as the root user, ensure that `PermitRootLogin=Yes` in `/etc/ssh/sshd_config`.
`Checking rsyslog setting:`	`PASSED`	Ensure that `/etc/rsyslog.conf` parses all `/etc/rsyslog.d/*.config` files, and that `imuxsock` is loaded in `/etc/rsyslog.conf`.
`Checking user and group specified:`	`PASSED`	You must specify an already available user and group when using the `--user` and `--group` options.
`Checking for krb5.keytab:`	`PASSED`	This is a warning only; however, do not kerberize the tenant storage while installing the Controller host using the web interface as described in Platform Controller Setup. Doing so will destroy any configuration you may have already done.
`Checking cgconfig kernel params:`	`PASSED`	
`Checking compatible python version:`	`PASSED`	
`Checking for presence of erlang cookie`	`PASSED`	

last-updated: May 06, 2024

Option	Expected Result	Error Resolution
`Checking SSL server certificate and private key:`	`PASSED`	The paths specified for both the certificate and private key must be absolute paths.
`Total: 16 -- Failed: 0 -- Warning: 0 -- Forced(success): 0`		
`Filesystem free space checks`		
`Checking freespace on /`	`PASSED`	If you are using a logical volume for the root filesystem, expand the volume and execute the `resizefs` command. If needed, use the `--force` option to ignore this failure. **CAUTION:** *Forcing the installation to continue might cause HPE Ezmeral Runtime Enterpriseto be unusable.*
`Checking freespace on /srv:`	`PASSED`	If you are using a logical volume, expand the volume and execute the `resizefs` command. Alternatively, you may provision another disk and use it to create a mountpoint at either `/srv` or `/srv/HPE`.
`Checking freespace on /var:`	`PASSED`	If you are using a logical volume, expand the volume and execute the `resizefs` command. Alternatively, you may provision another disk and use it to create a mountpoint at either `/var`, `/var/lib` or `/var/lib/docker`.
`Checking freespace on /opt:`	`PASSED`	If you are using a logical volume, expand the volume and execute the `resizefs` command. Alternatively, you may provision another disk and use it to create a mountpoint at either `/opt` or `/opt/HPE`.
`Checking configured swap size`	`PASSED`	This is a warning only. HPE recommends a swap size that is at least 20% of Host RAM.
`Checking docker storage`	`PASSED`	Checks node storage.
`Total: 6 -- Failed: 0 -- Warning: 0 -- Forced(success): 0`		

Option	Expected Result	Error Resolution

```
*************************************************************************
     Aggregate tests summary:
        Total:           56
        Failed:           0
        Warning:          0
        Forced(success):  0
Additional information for debugging is written to
/tmp/bd_prechecks.14962.log
*************************************************************************
```

Once you are satisfied that the pre-check has completed correctly, you may proceed to install the Controller using either of the following methods:

- **Using the pre-check configuration file:** This is the preferred Controller installation method. See Using the Pre-Check Config File on page 853.
- **Manually:** This option requires you to specify all installation options. See Standard Installation on page 854.

See Pre-Check Generated Files on page 852 to view the files that are generated by the pre-check script.

Pre-Check Generated Files

The pre-check script (see Using the Pre-Check Script) generates the following files:

- **Precheck Log:** This log is more detailed than the basic output described in Sample Pre-Check Output, but contains high-level information based on the options supplied when you ran the script and the findings generated by the script. This file will be stored in `/tmp/bd_prechecks.<pid>.log`, where `<pid>` is the process ID number of the script run, such as `11893`. HPE may request this file if you contact us for technical support.
- **Xtrace:** This file is a verbose, line-by-line description of the exact commands used by the script to both get data and determine the outcome of each test. This file will be stored in `/tmp/bd_prechecks.<pid>.log.xtrace`, where `<pid>` is the process ID number of the script run, such as `11893`. It is intended for use by HPE for support purposes, and HPE may request this file if you contact us for technical support.

NOTE: Neither the .log nor .xtrace files will pass any data or other sensitive information to HPE.

- **Configuration file:** The pre-check script will also generate a configuration file that you can use to speed up installing the Controller host. This file passes relevant configuration to the installer and bypasses the checking performed by that script. This file will be named `/tmp/bd_prechecks.conf`. See Using the Pre-Check Config File.

CAUTION: DO NOT MODIFY THE CONTENTS OF THE CONFIG FILE BEFORE PASSING IT TO THE INSTALLER AS THIS COULD CAUSE THE CONTROLLER HOST INSTALLATION PROCESS TO FAIL.

CAUTION: THE PRE-CHECK SCRIPT WILL GENERATE A CONFIGURATION FILE REGARDLESS OF ANY WARNINGS OR ERRORS ENCOUNTERED. DO NOT USE THE CONFIGURATION FILE TO INSTALL HPE EZMERAL CONTAINER PLATFORM IF ANY TESTS HAVE FAILED.

Phase 3

The topics in this section describe Phase 3 of deploying HPE Ezmeral Runtime Enterprise.

NOTE: Refer to the Updating External Service Passwords on page 141 page for steps on changing the default password for external services such as Nagios, HAProxy and HACluster.

Step 1: CLI
The topics in this section describe the CLI tasks that are the first step in this phase of deploying HPE Ezmeral Runtime Enterprise.
Using the Pre-Check Config File

NOTE: This article describes using the precheck scripts as part of installing HPE Ezmeral Runtime Enterprise. For information about using the scripts during an upgrade, see the instructions for upgrading the platform.

Using the config file generated by the pre-check script (see Using the Pre-Check Script and Pre-Check Generated Files to install HPE Ezmeral Runtime Enterprise on the Controller host can be a useful option. This option bypasses the pre-check script or overrides pre-check values provided that all of the following conditions are met:

- You are aware of and have accounted for any warnings contained in the pre-check output.
- The pre-check output did not contain any errors.
- Nothing has changed about the Controller host, user, network, infrastructure, operating system, or configuration since the pre-check was successfully run.

CAUTION: DO NOT USE THE CONFIG FILE TO INSTALL THE CONTROLLER IF THE PRE-CHECK OUTPUT LISTS ONE OR MORE ERROR(S).

NOTE: Be sure to rerun the pre-check script with the appropriate option(s) if anything has changed since the last time it was successfully run.

To use the config file for the installation:

1. Log into the host that you will be using as the Controller host using either the root account and password or your assigned username and password.
2. If needed, copy the binary (.bin) to the host that you will use as the Controller host.
3. Make the .bin file executable by executing the command `chmod a+x <hpe_ezmeral>.bin`

   Where:

   - `<hpe_ezmeral>` is the full name of the .bin file.
4. Run the executable binary from the Linux console as the assigned user by typing `./<hpe_ezmeral>.bin --prechecks-config-file <path> <additional_options>`, where:
   - `<hpe_ezmeral>` is the full name of the .bin file.
   - `<path>` is the complete path to the config file generated by the pre-check script.
   - `<additional_options>` are any additional options you need to add, such as --default-password. See Standard Installation for a list of options.
5. The installer checks the integrity of the bundle and then extracts the bundle contents.
6. The End User License Agreement (EULA) appears. Read through the EULA, pressing [SPACE] to page through the content. Once you have viewed the entire EULA, press [y] to accept it and continue the installation.

7. HPE Ezmeral Runtime Enterprise installs on the Controller host. A series of messages appear during the installation. The following message appears once the installation is complete:

```
                          Successfully installed HPE CP.
                          Please visit https://10.50.1.1 to configure the
server.
                          [root@hostname-1 ~] .
```

This concludes the first phase of the installation. Note the URL provided, as you will use this to continue configuration. Please proceed to Platform Controller Setup to continue the installation using the web interface.

NOTE: If you encounter any errors during the installation process, see Step 1 Troubleshooting for information on diagnosing and fixing those errors. If problems persist, contact HPE for support.

Standard Installation

CAUTION:

If you are unable to install after attempting to remediate errors and warnings, contact Hewlett Packard Enterprise for support.

To perform a manual installation on the Controller host:

1. Log into the host that you will be using as the Controller host using either the `root` account and password or your assigned username and password.

2. If needed, copy the HPE Ezmeral Runtime Enterprise Enterprise binary (.bin) to the host that you will use as the Controller host.

3. Make the .bin file executable by executing the command `chmod a+x hpe-cp-<os>-<version>-<build>.bin`, where:

   - `<os>` is the operating system supported by this .bin file. This can be either `sles` (for SLES) or `rhel` (for Red Hat Enterprise Linux/CentOS).

   - `<version>` is the .bin version.

   - `<build>` is the specific .bin build number.

4. Run the executable binary from the Linux console as the assigned user by typing `./<hpe_ezmeral>.bin <options>`, where `<options>` is a list of one or more configuration option(s). See Installer Options and Examples, below.

   CAUTION: See the top row of the Installer Options on page 855 table for the `--default-password` OPTION.

   CAUTION: If you want to install HPE Ezmeral Runtime Enterprise Essentials, see the second row of the Installer Options on page 855 table for the `--default-password <password> --tier essentials` OPTION.

   CAUTION: If you want to install HPE Ezmeral Runtime Enterprise, see the third row of the Installer Options on page 855 table for the `--default-password <password> --tier enterprise` OPTION.

   CAUTION: If you do not specify `--tier` option with `--default-password`, HPE Ezmeral Runtime Enterprise will be installed by default.

5. The installer checks the integrity of the bundle and then extracts the bundle contents.

6. The End User License Agreement (EULA) appears, if you have not used the `-s` or `--skipeula` option. Read through the EULA, pressing [SPACE] to page through the content. Once you have viewed the entire EULA, press [y] to accept it and continue the installation.

7. HPE Ezmeral Runtime Enterprise installs on the Controller host. A series of messages appear during the installation. The following message appears once the installation is complete:

```
Successfully installed HPE CP.
Please visit https://10.50.1.1 to configure the server.
[root@hostname-1 ~] .
```

This concludes the first phase of the installation. Note the URL provided, as you will use this to continue the configuration. Proceed to Platform Controller Setup to continue via the web interface.

Installer Options

This table lists all of the Controller-specific options that can be used with the installation script. Not all of these options will apply in all situations.

Option	Shortcut	Description	Notes	Example
`--default-password`		Specify a default password.	Specifies the default password for the `admin` and `demo.user` users that are created during the installation process. Failure to include this parameter will cause the default password for these users to be set to `admin123`.	
`--default-password <password> --tier essentials`		Install `--tier` must be specified with `essentials` to install HPE Ezmeral Runtime Enterprise Essentials		
`--default-password <password> --tier enterprise`		Install `--tier` must be specified with `enterprise` to install HPE Ezmeral Runtime Enterprise		

`--proxy`	`-p`	Proxy URL in the format
		`[protocol://]`
		`[username:password@]`
		`proxy_address[:port]`
			Skip this option if no web proxy is being used for EPIC applications. See Web Proxy Requirements.	`--proxy https://admin:admin123@172.1.1.2:8080`

Option	Shortcut	Description	Notes	Example
`--no-proxy-ips`		Comma-separated list of IP address that have been assigned for use by HPE Ezmeral Runtime Enterprise.	See Web Proxy Requirements.	`--no-proxy-ips 127.0.0.1,172.16.1.10`
`--no-ntp-config`		*Optional*. Specify this option to prevent the installer from modifying the NTP configuration.		
`--worker-agent-install`		Worker hosts must be initialized manually before installing them via the web interface.	*ONLY USE THIS OPTION WHEN YOUR ENVIRONMENT DOES NOT ALLOW KEY-BASED SSH. SEE Configuration Requirements.*	
`--storagepolicy`		*Optional*. Storage policy to use for tenant storage.	Click here for a list of supported storage policies (link opens a new browser tab/window). The `--reportstoragetype` option is REQUIRED when using this option.	`--storagepolicy ONE_SSD`
`--reportstoragetype`		*Optional**. Allow the data node to report the storage type (either `DISK` or `SSD`.	*This option is REQUIRED when using the `--storagepolicy` option, above.	`--storagepolicy ONE_SSD`
`--int-gateway-ip`		Internal gateway IP address for virtual nodes. This address cannot be used by another resource on the corporate network. See Network Requirements.	*Optional*. If not specified, default value is 172.16.13.1	`--int-gateway-ip 172.1.1.1`
`SUDO_PREFIX`		Specify the actual sudo prefix to use when running privileged commands as a non-root user.	Defaults to `sudo -n` if no other option provided.	`export SUDO_PREFIX="usr/sbin/dzdo/dzdo"`

Option	Shortcut	Description	Notes	Example
`--ssl-cert`		SSL certificate to use for secure (https://) access to the web interface.	Do not use this option if you will be using non-secured (http://) access to the web interface. If you expect to be creating and using Kubernetes clusters, then you must provide a previously-created and signed TLS Certificate and Key with Subject Alternate Names that indicate all IP Addresses (as IP Addresses), hostnames (as Domains), URLs (as Domains) and aliases (as Domains) that users expect to be able to use to access the installation. (A Kubernetes cluster can be created without specifying an SSL certification for your controller/gateway.) This encompasses all Controller hosts (Primary, Shadow, Arbiter), all Gateway hosts, and all Gateway sets, except in cases where gateway hosts and controllers have different certificates.	`--ssl-cert /root/bds-https.cert`
`--ssl-priv-key`		SSL private key to use for secure (https://) access to the web interface.	Do not use this option if you will be using non-secured (http://) access to the web interface.	`--ssl-priv-key /root/bds-https.pem`
`--ssl-ca-data`		Provide the CA authentication chain data required for having an SSL client authenticate the server certificate.	This must be an absolute file path that will be readable by the httpd process. The “CA data” file is used for an `openssl verify -CAfile` command to ensure that an SSL client (such as those used in our k8s support) can validate the certificate.	`--ssl-ca-data <path>`

Option	Shortcut	Description	Notes	Example
`--gateway-ssl-cert`		Provide the public-key SSL certificate for SSL termination in the gateway. If this is not provided, but the server certificate and private key are, these values will now default from those server certificate values (`--ssl-cert` and `--ssl-priv-key`).	This must be an absolute path.	`--gateway-ssl-cert <path>`
`--gateway-priv-key`		Provide the private key corresponding to the above.	This also must be an absolute path if given.	`--gateway-priv-key <path>`
`--force`	`-f`	Force the installer to proceed despite any error(s) that may be encountered.	*USING THIS OPTION MAY RENDER HPE EZMERAL CONTAINER PLATFORM UNSTABLE AND/OR UNUSABLE, EVEN IF THE INSTALLER SEEMS TO COMPLETE SUCCESSFULLY. THIS IS AN ADVANCED OPTION THAT MUST ONLY BE USED WHEN THE CAUSE OF THE ERROR IS KNOWN AND PROCEEDING WITH THE INSTALLATION WILL NOT CAUSE ANY PROBLEMS. CONTACT HPE FOR SUPPORT BEFORE USING THIS OPTION.*	
`--skipeula`	`-s`	*Optional*. Skips displaying the End User License Agreement (EULA).		
`--controller-public-if`		Interface name to use for the Controller host.	Use this option for a Controller installation if there are multiple interfaces present on the Controller with the same IP addresses assigned.	`--controller-public-if ens32`

Examples

This section presents some examples of using the installer.

- **Root/Agent:** This example installs HPE Ezmeral Runtime Enterprise as the root user and includes the Worker agent because password-less SSH is not available in the environment.

```
# root@localhost> /root/
hpe-cp-rhel-release-5.1-3010.bin --worker-agent-install
```

- **Root/non-Agent:** This example installs HPE Ezmeral Runtime Enterprise as the root user. The environment does allow password-less SSH, and thus the `--worker-agent-install` option is *not* used.

```
# root@localhost> /root/hpe-cp-rhel-release-5.1-3010.bin
```

- **Root/Agent/SSL:** This example installs HPE Ezmeral Runtime Enterprise as the root user and includes the Worker agent because password-less SSH is not available in the environment. It also provides SSL information to enable secure (https://) access to the web interface.

```
# root@localhost> /root/
hpe-cp-rhel-release-5.1-3010.bin --worker-agent-install --ssl-cert /root/
bdhost.cert --ssl-priv-key /root/bdhost.pem
```

- **Root/Non-Agent/SSL:** This example installs HPE Ezmeral Runtime Enterprise as the root user. The environment does allow password-less SSH, and thus the `--worker-agent-install` option is *not* used. This example also provides SSL information to enable secure (https://) access to the web interface.

```
# root@localhost> /root/hpe-cp-rhel-release-5.1-3010.bin --ssl-cert /root/
bdhost.cert --ssl-priv-key /root/bdhost.pem
```

- **Non-root/Agent:** This example installs HPE Ezmeral Runtime Enterprise as a non-root user and includes the Worker agent because password-less SSH is not available in the environment. To perform this action, login as the *specific non-root user*, for example "epic" in this case.

```
# # epic@localhost> /home/epic/
hpe-cp-rhel-release-5.1-3010.bin --worker-agent-install
```

- **Non-root/non-Agent:** This example installs HPE Ezmeral Runtime Enterprise as a non-root user. The environment does allow password-less SSH, and thus the `--worker-agent-install` option is *not* used. To perform this action, login as the *specific non-root user*, for example "epic" in this case.

```
# epic@localhost> /home/epic/hpe-cp-rhel-release-5.1-3010.bin
```

- **Non-root/Agent/SSL:** This example installs HPE Ezmeral Runtime Enterprise as a non-root user and includes the Worker agent because password-less SSH is not available in the environment. It also provides SSL information to enable secure (https://) access to the web interface. To perform this action, login as the *specific non-root user*, for example "epic" in this case.

```
# epic@localhost> /home/epic/
hpe-cp-rhel-release-5.1-3010.bin --worker-agent-install --ssl-cert /home/
epic/bdhost.cert --ssl-priv-key /home/epic/bdhost.pem
```

- **Non-root/non-Agent/SSL:** This example installs HPE Ezmeral Runtime Enterprise as a non-root user. The environment does allow password-less SSH, and thus the --worker-agent-install option is *not* used. This example also provides SSL information to enable secure (https://) access to the web interface. To perform this action, login as the *specific non-root user*, for example "epic" in this case.

```
# epic@localhost> /home/epic/hpe-cp-rhel-release-5.1-3010.bin --ssl-cert /
home/epic/bdhost.cert --ssl-priv-key /home/epic/bdhost.pem
```

Step 1 Troubleshooting

This article contains instructions that may help you if you run into problems during command line installation. To troubleshoot errors:

1. Open the installer log file (see Installer Logs) and begin reading it from top to bottom.
2. Stop at the first ERROR you find. The first error can often cause further problems downstream, and taking a start-to-finish approach (instead of working your way back from the tail end of the log file) may help you solve one error that in turn resolves a series of cascading errors.
3. If the problem is obvious (such as a typo while setting options), then correct the problem and re-run the installer.
4. If the error is listed in Common Errors, then attempt the remediation step(s) outlined for that error.
5. If you are unable to resolve the problem(s) on your own, then contact HPE for support. You may be asked to provide the installer log and xtrace files.

Installer Logs

The installer generates the following files:

- **Installer Log:** This log contains high-level information based on the options supplied when you ran the installer and the error(s) encountered during installation. This file will be stored in `/tmp/bds_<timestamp>.log`, where `<timestamp>` is the time the installer was run in `yearmonthdayhourminutesecond` format (such as `bds_20170401223718.log`. HPE may request this file if you contact us for support.
- **Xtrace:** This file is a verbose, line-by-line description of the exact commands used by the script to both get data and determine the outcome of each test. This file will be stored in `/tmp/bds_<timestamp>.log.xtrace`, where `<timestamp>` is the time the installer was run in `yearmonthdayhourminutesecond` format (such as `bds_20170401223718.log.xtrace`. HPE may request this file if you contact us for support.

NOTE: Neither the .log nor .xtrace files will pass any data or other sensitive information to HPE.

Common Errors

The most common installation errors include:

- Yum MultiLib Version Errors
- RPM Version Error 1

YUM MultiLib Version Errors

There are many reasons why YUM may throw a multilib version error. Please refer to the Red Hat documentation (Solution #57783) for help resolving these problems. Use the **yum-config-manager** to set the required options (and work around the issues displayed in the previous image). You must have a valid RedHat subscription in order to access this page.

RPM Version Error 1

You may receive an error saying that a newer version of an RPM or a dependency is already installed. This error may look similar to the following:

```
                         ---> Package netpbm.x86_64 0:10.47.05-11.el6 will be
installed
                         ---> Package perl-hivex.x86_64 0:1.3.3.-4.2.el6 will
be installed
                         --> Finished Dependency Resolution
                         Error: Package:
cyrus-sals-md5-2.1.23-13.el6_3.1.x86_64 (centos6)
                                    Requires: cyrus-sasl-lib =
2.1.23-13.el6_3.1
                                    Installed:
cyrus-sasl-lib-2.1.23-15.el6.x86_64 (@base)
                                        cyrus-sasl-lib = 2.1.23-15.el6
                                    Available:
cyrus-sasl-lib-2.1.23-13.el6_3.1.x86_64 (centos6)
                                        cyrus-sasl-lib = 2.1.23-13.el6_3.1
                         You could try using --skip-broken to work around the
problem
                         You could try running: rpm -Va --nofiles -- nodigest
```

Resolution:

- One way to recover is to erase the newer (already installed) version and try again. When erasing through YUM, it may end up treating almost all the packages on the system as dependencies and trying to erase everything. This often happens when the RPM that you are trying to delete installs some shared libraries. Avoid using the `-y` options when you are in this situation.

- If the above problem prevents you from performing a YUM erase, you can try to use `rpm -e` directly. First, find the version of the RPM required in the error log and copy it to a known location. Execute rpm -e on the existing installed version of the rpm and immediately install the required version using `rpm -ivh`. The preceding solutions may fail with some of the core RPMs, leaving the system in an unusable state. Attempts to execute any binary will return an error saying that some library is missing. The only recourse in this case is to perform a fresh OS reinstallation.

Step 2: GUI

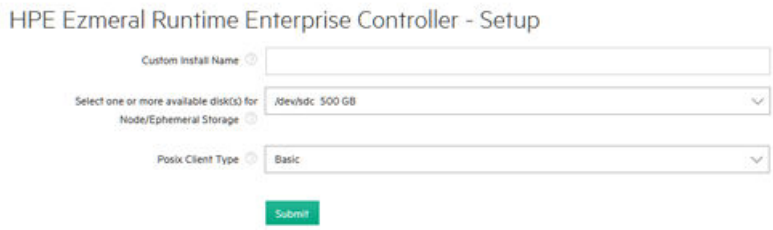
The topics in this section describe the tasks that are the second step in this phase of deploying HPE Ezmeral Runtime Enterprise.

Platform Controller Setup

The next step of the installation process uses a Web browser to access the web interface. To do this:

1. Open a Web browser and navigate to the URL provided at the end of the command line installation process (see Using the Pre-Check Config File or Standard Installation, as appropriate).

   The **HPE Ezmeral Runtime Enterprise Controller - Setup** screen appears.



2. If desired, enter a custom installation name in the **Custom Install Name** field. This name will appear in the **Toolbar** to help you identify this deployment, which can be useful if you are administering multiple deployments. You may add, edit, or remove this name at any time using the **Other** tab of the **Settings** screen. See Other Tab.

3. Select one or more disks to use for ephemeral storage from the **Select one or more available disk(s) for Node/Ephemeral Storage** menu.

   Ephemeral storage does not persist after the pods cease to exist.

   Press either [CONTROL] (Windows/Linux) or [COMMAND] (MacOS) while clicking to select multiple disks. If you make a mistake, then either [CONTROL]-click or [COMMAND]-click the selected disks that you want to remove.

4. Use the **Posix Client Type** pull-down menu to select the type of Posix client to use (**Basic** or **Platinum**). Click here for more information (link opens in a new browser tab/window).

5. Click **Submit** to finish the installation on the Controller host.

   A popup appears indicating that the installation process has started successfully.

   This popup is subsequently replaced by a status summary as the installation completes.

6. If you like, you may click the **Details** button to open a popup that displays additional information about the installation. Please allow about 20 minutes for this process to complete (actual time will vary depending on various factors).

7. The **Setup completed successfully** dialog appears when the installation process completes. Click the **Close!** button to exit to the web interface **Login** screen.

Proceed to installing a Gateway host and, if desired, enable platform high availability and Using the Air Gap Utility on page 869 (if applicable) you validate the installation or add Kubernetes hosts to the deployment.

Step 2 Troubleshooting

This article contains instructions that may help you if you run into problems during installation via the command line. To troubleshoot errors:

1. Open the installer log file (see Installer Logs) and begin reading it from top to bottom. You may also view the summary and details that appear during installation (see Platform Controller Setup).

2. Stop at the first ERROR you find. The first error can often cause further problems downstream, and taking a start-to-finish approach (instead of working your way back from the tail end of the log file) may help you solve one error that in turn resolves a series of cascading errors.

3. If the problem is obvious (such as a typo while setting options), then correct the problem and re-run the installer.

4. If you are unable to resolve the problem(s) on your own, then contact Hewlett Packard Enterprise for support. You may be asked to provide the installer log and xtrace files.

Installer Logs

The installer generates the following files:

- **Installer Log:** This log contains high-level information based on the options supplied when you ran the installer and the error(s) encountered during installation. This file will be stored in `/var/log/bluedata/install/install.out_<timestamp>.log`, where `<timestamp>` is the time the installer was run in `year-month-day-hour-minute-second` format (such as `install.out_2017-04-02-01-33-21.log`. HPE may request this file if you contact us for support.

- **Xtrace:** This file is a verbose, line-by-line description of the exact commands used by the script to both get data and determine the outcome of each test. This file will be stored in `/var/log/bluedata/install/install.out_<timestamp>.log.xtrace`, where `<timestamp>` is the time the installer was run in `year-month-day-hour-minute-second` format, such as `bds_2017-04-02-01-33-21.log.xtrace`. Hewlett Packard Enterprise may request this file if you contact us for support.

NOTE: Neither the .log nor .xtrace files will pass any data or other sensitive information to Hewlett Packard Enterprise.

Phase 4

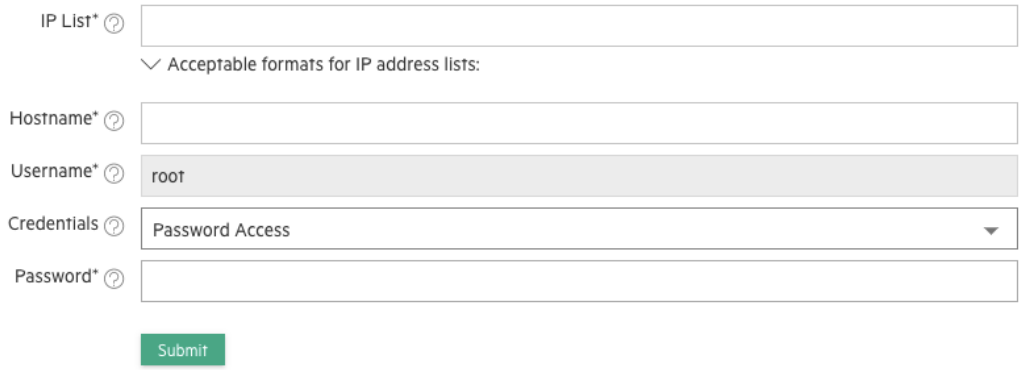
The topics in this section describe Phase 4 of deploying HPE Ezmeral Runtime Enterprise.

Installing a Gateway Host

To add one or more Gateway hosts, you will use the top portion of the **Gateway/Load Balancer** screen (see The Gateway/Load Balancer Screen).

Before adding one or more Gateway hosts, ensure that the hosts conform to the requirements described in Host Requirements on page 813.

If the `firewalld` service is installed and enabled on the Controller, and the `firewalld` service is installed and enabled on all hosts before they are added to the deployment, the installer for HPE Ezmeral Runtime Enterprise automatically configures firewall rules to open the required ports.



To select the hosts:

1. If you do not see the **User name** and **Password** fields, then follow the instructions found in Agent-Based Gateway Installation; otherwise, proceed to Step 2.

2. Enter the IP addresses of the Gateway hosts that you are adding in the **IP List** field. You may select one or more hosts as follows:

   - **Single IP address:** Enter a properly formatted IP address, such as `10.10.1.1`. This will add a single host.

   - **Multiple IP addresses:** Enter the first three octets of the IP addresses, and then separate each digit of the fourth octet with a comma, such as `10.10.1.1,2,5,8`. In this example, four Gateway hosts with IP addresses of 10.10.1.1, 10.10.1.2, 10.10.1.5, and 10.10.1.8 will be added.

- **Multiple IP addresses:** Enter multiple IP addresses separated by commas, such as `10.10.1.1`, `10.10.1.2`, `10.10.1.5`, `10.10.1.8`. In this example, four Gateway hosts with the same IP addresses as the previous example will be added.
- **IP address range:** Enter an IP address range, such as `10.10.1.1-8`. In this example, eight Gateway hosts with IP addresses from 10.10.1.1 to 10.10.1.8 will be added.
- **Combination:** Use a combination of the above methods, such as `10.10.1.1`, `10.10.1.2,5,8,10.10.1.9-12`.

NOTE: You may only perform one set of Gateway host additions to one or more hosts at once. To save time, consider adding all of the Gateway hosts at once by entering multiple IP addresses as described above.

3. Select how to access the Gateway hosts. Your available options are:
   - **Password access:** Check the **Password Access** radio button and then enter the password for the Gateway hosts you are adding in the **Password** fields. The password must be valid for the username in the **User name** field.
   - **SSH Key:** If the Gateway hosts already have a public key installed to allow password-free access, then you may check the **SSH Key based Access** radio button. Upload the private key by clicking the **Browse** button to open a standard **File Upload** dialog that allows you to browse for and select the key file. If the key requires a pass phrase, enter that phrase in the **Passphrase** field. The uploaded private key will only be used for initial host access and will not be permanently stored.

   NOTE: If Gateway installation fails because of a security error, then check the local times on the Controller and Gateway Hosts. If these times are significantly different, then set the local time on the Gateway host to match the local time on the Controller host, and then begin the installation process again.

4. Click the **Add Gateway** button to install the selected Gateway hosts.

The selected Gateway hosts are installed. The **Gateway(s) Status** table displays the following information for each host you are adding:

- **Host:** IP address and hostname of the Gateway host.
- **Details:** Information about the Gateway host (RAM, CPU cores, etc.).2
- **Status:** Current status of the Compute host, which updates as the installation progresses. This will appear as one of the following:
  - **Connecting:** HPE Ezmeral Runtime Enterprise is attempting to connect to the listed Gateway hosts.
  - **Running bundle:** HPE Ezmeral Runtime Enterprise has successfully connected to the listed Gateway hosts and is preparing the hosts.
  - **Bundle completed:** HPE Ezmeral Runtime Enterprise has completed preparing the listed Gateway hosts, which are ready to be added to the deployment. If you added the hosts by mistake, you may remove them by clicking the **Delete** icon (trash can).
  - **Installed:** The Gateway host is available for use.
- **Actions:** Once the Gateway hosts are reviewed, a **Delete** icon (trash can) will appear next to that Gateway. See Deleting a Gateway Host.

Troubleshooting

If you experience issues when installing a Gateway host, then access the following logs:

- **Controller host:**
  - **Gateway Installer log:**`/var/log/bluedata/install/addworker.out_.log`.
  - **Xtrace file:** This file is a verbose, line-by-line description of the exact commands used by the script to both get data and determine the outcome of each test. This file will be stored in `/var/log/bluedata/addworker/install.out_.log.xtrace`.
- **Gateway host**
  - **Gateway setup log:** `/var/log/bluedata/install/worker_setup_<timestamp>`
  - **Gateway Xtrace set-up file:** `/var/log/bluedata/install/worker_setup_<timestamp>.xtrace`

Begin reading these logs from top to bottom.

Stop at the first ERROR you find. This first error can often cause further problems downstream, and taking a start-to-finish approach (instead of working your way back from the tail end of the log file) may help you solve one error that in turn resolves a series of cascading errors. If the problem is obvious, then correct the problem and re-run the installer.

If you are unable to resolve the problems on your own, then contact Hewlett Packard Enterprise for support. You may be asked to provide the these installer logs and xtrace files.

Enabling Platform High Availability

Platform High Availability (HA) protects your HPE Ezmeral Runtime Enterprise a failure of the Controller host. Hewlett Packard Enterprise recommends that you enable HA for the HPE Ezmeral Runtime Enterprise Controller before you create Kubernetes clusters.

Prerequisites

Required access rights: Platform Administrator

New Deployments: In a new deployment of HPE Ezmeral Runtime Enterprise, the prerequisites to enabling platform HA are the following:

- You have completed installing HPE Ezmeral Runtime Enterprise and completed Platform Controller Setup on page 861 on the Controller host.
- You have two hosts that conform to the requirements for controller hosts and to the high-availability requirements listed in Host Requirements. These two hosts will become the Shadow Controller and the Arbiter.
- Hewlett Packard Enterprise recommends enabling platform High Availability shortly after initial installation, before adding a large number of Kubernetes hosts.

  Hewlett Packard Enterprise recommends enabling platform High Availability before creating any Kubernetes clusters, including an HPE Ezmeral Data Fabric on Kubernetes cluster. Kubernetes clusters that were created before enabling platform HA might not send data to the correct host after an HA failover. If you want to enable platform HA without deleting existing Kubernetes clusters, contact Hewlett Packard Enterprise Support for assistance.
- If the Controller and the Shadow Controller hosts are to be on the same subnet, in order for the cluster IP address to function correctly, the external switch connecting the hosts to the network must support gratuitous ARP.

- If a cluster IP address is not provided, the Controller and the Shadow Controller are not required to be on the same subnet.

Changing HA Hosts: If you want to change the hosts used for Shadow Controller and Arbiter roles after platform HA has been enabled, you must disable HA protection and then re-enable HA protection using the updated IP addresses and hostnames.

Re-enabling Platform HA: If you are re-enabling platform HA after disabling platform HA in an existing deployment, the prerequisites are the following:

- If platform HA was disabled while the Shadow Controller host was offline, when the faulty hardware is replaced and HA protection is re-enabled, both of the following are required:
  - The original Arbiter host must be redesignated as the Arbiter.
  - The new Shadow Controller host must use the same IP address as the previous Shadow Controller host.
- If platform HA was disabled while the Arbiter host was offline, when the faulty hardware is replaced and HA protection is re-enabled, both of the following are required:
  - The original Shadow Controller host must be redesignated as the Shadow Controller.
  - The new Arbiter host must use the same IP address as the previous Arbiter host.

About this task

When enabling platform High Availability on page 132 for a new HPE Ezmeral Runtime Enterprise deployment, you will add two hosts. The hosts you add become the Shadow Controller and Arbiter hosts. The hosts can not be used for any other purpose.

Procedure

1. If you have not already done so, add the hosts that will become the Shadow Controller and Arbiter hosts to the deployment.

   See Adding the Shadow Controller and Arbiter Hosts on page 742.

2. Enter Lockdown mode as described in Lockdown Mode on page 916.

3. On the **Controllers & HA** screen, select **Enable HA**.

4. Enter values in **Cluster IP**, **Cluster Name**, or both, as appropriate:
   - If the Controller and Shadow Controller hosts are in different subnets, then you must leave the **Cluster IP** field blank. By leaving both the **Cluster IP** and **Cluster Name** fields blank, you can access the web interface by navigating to `http://<gateway_ip>` or `https://<gateway_ip>`, as appropriate, where `<gateway_ip>` is the IP address of a Gateway host. See Gateway Hosts on page 106.
   - If the Controller and Shadow Controller hosts are on the same subnet, then you can enter an available IP address to use as the cluster IP address in the **Cluster IP** field.

     The cluster IP address must be in the same subnet as the Controller host and cannot be in use by any other resource.

     If you do not supply a cluster IP address, if you have defined a cluster name in the **Cluster Name** field, then you can access the web interface by navigating to `http://<cluster-name>` or `https://<cluster-name>`, as appropriate. This cluster name must be mapped to the cluster IP address in a user-accessible DNS server. You can also access the web interface using a Gateway IP address.

5. Use the **Shadow Controller** and **Arbiter Node** menus to select a host for each role.

   If the deployment has three hosts, then after you select a host as Shadow Controller or Arbiter, the remaining host is automatically assigned to the other role.

   If there are more than three hosts, no automatic assignment occurs. You can select the Shadow Controller and Arbiter hosts in any order.

   You cannot remove or modify the Shadow Controller or Arbiter host while platform HA is enabled.

6. Click **Submit**.

   The **Controllers** tab displays the message **HA Setup in progress**. This process may take up to 30 minutes to complete, depending on a number of factors. During HA setup, this page reloads and you are signed out. To see updated status, sign in and view this page.

   If you want more detailed information about the setup process, click the **Details** button to open the **HA Setup Details**.

   After the setup process completes, a message appears informing you that HPE Ezmeral Runtime Enterprise is running in High Availability mode, and reminding you to begin using the cluster IP address or cluster name to sign in to the web interface.

   If you installed the Network Manager service while installing the base OS on the hosts, then this service will stop because it conflicts with the High Availability monitoring services.

7. Click **Click here to migrate to Cluster Name** link in the message.

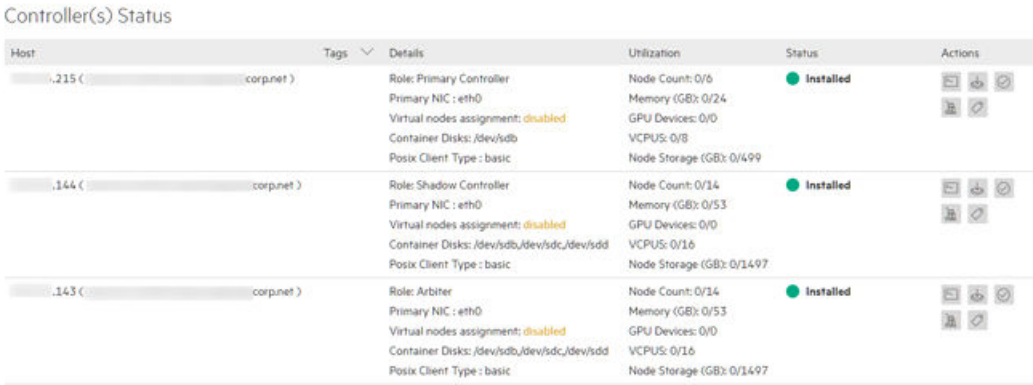
   Clicking the **Click here to migrate to Cluster Name** link in this message logs you out of the web interface and returns you to the sign-in screen using the cluster IP address.

8. Sign in to HPE Ezmeral Runtime Enterprise.

9. Exit Lockdown mode as described in Lockdown Mode on page 916.

Results

The newly added Shadow Controller and Arbiter will appear in the **Controllers & HA** screen.



After enabling HA, Hewlett Packard Enterprise recommends that you use either the cluster IP address or cluster name to sign into the web interface. Doing so will automatically connect you to the Controller host (during normal operation) or the Shadow Controller host (when a Controller host failure triggers HA protection). If the Controller host fails, then you will not be able to access the web interface using the IP address of that host.

If enabling High Availability fails, then the fields in the **HA Setting** section of the **Controllers & HA** screen reappear, and the deployment continues to run with a single Controller host. Contact Hewlett Packard Enterprise Support for assistance.

Related reference

High Availability on page 132
High availability (HA) in deployments of HPE Ezmeral Runtime Enterprise is divided into platform controller HA, gateway HA, and cluster HA.

Configuring Air Gap Kubernetes Host Settings

Download image and RPM files and configure settings for Kubernetes hosts in an air-gapped environment.

CAUTION:

If you will be using an air-gap configuration for Kubernetes objects, then you must configure air-gap settings before adding any Kubernetes hosts.

Apply all air-gap settings with care. These settings do not propagate if updated after Kubernetes hosts have been installed unless one of the following occurs:

- The Kubernetes host is rebooted.
- The version of Kubernetes running on a host is upgraded.

Any Kubernetes hosts in a ready state that are not part of a Kubernetes cluster must be restarted for the changes to be applied.

IMPORTANT: Changing an existing HPE Ezmeral Runtime Enterprise configuration from a non-airgapped environment to an air-gapped environment forces a reinstall of Kubernetes clusters.

If you are changing an existing HPE Ezmeral Runtime Enterprise configuration from a non-airgapped environment to an air-gapped environment, contact Hewlett Packard Enterprise support for assistance before you begin the transition. Several manual steps must be performed to transition to an air-gapped environment.

For information about the requirements for air-gapped installation, see Kubernetes Air-Gap Requirements on page 834.

Air Gap File Download Locations

You can download files for air-gapped installation from the following locations:

Kubernetes container images (HPE Ezmeral Runtime Enterprise 5.4.0 and above)	For HPE Ezmeral Runtime Enterprise 5.4.0 and above, the complete set of air gap container image files can be downloaded with the air gap utility script. See Using the Air Gap Utility on page 869.

Installing Air-Gapped Kubernetes Hosts

To install Kubernetes hosts without internet access (air gap environment) do the following:

1. Use the air gap utility script (Using the Air Gap Utility on page 869) to download container images and import them to a local filesystem or remote container registry.

   For information on using container registries, see either Existing Container Registry on page 869 or New Container Registry on page 868, as appropriate.

2. Configure the air-gap parameters. See Air Gap Tab on page 799.

3. Proceed with adding the Kubernetes hosts.

New Container Registry

To use a new container registry:

1. Create an open-source Docker registry. Refer to these instructions (link opens an external website in a new browser tab or window).

   You are not required to set up a Docker Trusted Registry.

2. After the registry is deployed, follow the Existing Container Registry on page 869 instructions to import the images into the registry.

Existing Container Registry

To use an existing container registry:

1. Obtain the URL and credentials for your container registry.

2. Proceed with Using the Air Gap Utility on page 869.

Related tasks

Kubernetes Worker Installation Overview on page 528
Describes how to add a host to HPE Ezmeral Runtime Enterprise as a Kubernetes worker for compute workloads.

Related reference

Air Gap Tab on page 799
The **Air Gap** tab of the **System Settings** screen enables the Platform Administrator to specify settings to be used when the Kubernetes hosts, clusters, and tenants do not have connectivity to the Internet.

More information

Kubernetes Air-Gap Requirements on page 834
Using the Air Gap Utility on page 869
Describes how to use the air gap utility to download files in an air-gapped HPE Ezmeral Runtime Enterprise environment.

Using the Air Gap Utility

Describes how to use the air gap utility to download files in an air-gapped HPE Ezmeral Runtime Enterprise environment.

Requirements:

- **Python:**
  - 2.7
  - 3.6 and above
- **Operating system:**

  At minimum:
  - RHEL 8
  - SLES 15
  - CentOS 7x
- **Skopeo:**

  At minimum:
  - For RHEL or CentOS:
    - Skopeo 0.1.40
  - For SLES:

- Skopeo 0.1.41

CAUTION:

If you will be using an air-gap configuration for Kubernetes objects, then you must configure air-gap settings before adding any Kubernetes hosts.

Apply all air-gap settings with care. These settings do not propagate if updated after Kubernetes hosts have been installed unless one of the following occurs:

- The Kubernetes host is rebooted.
- The version of Kubernetes running on a host is upgraded.

Any Kubernetes hosts in a ready state that are not part of a Kubernetes cluster must be restarted for the changes to be applied.

About the Air Gap Utility

HPE Ezmeral Runtime Enterprise provides a utility you can use to query, filter, and download all air gap container images necessary for your environment to a local filesystem or remote registry.

Installing the Air Gap Utility Package

Before downloading files for your air gap environment, you must first install the air gap script package. You can install the package on any non-platform host, even outside the platform installation. Python 2.7 or Python 3.6 and greater is required for install.

To install the air gap utility package:

1. Download the air gap utility package from the following links:
   - HPE air gap utility (North America download site)
   - HPE air gap utility (Asia Pacific download site)
2. Install Skopeo. In the CLI, enter the following:
   - If you are using RHEL:

     ```
     dnf install -y skopeo
     ```

   - If you are using SLES:

     ```
     zypper install -y skopeo
     ```

3. Install the `hpeairgaputil` package:
   - PIP2:

     ```
     pip install hpeairgaputil-1.3-py2.py3-none-any.whl
     ```

   - PIP3:

     ```
     pip3 install hpeairgaputil-1.3-py2.py3-none-any.whl
     ```

NOTE: To uninstall `hpeairgaputil`, use:

- PIP2:

```
pip uninstall hpeairgaputil-1.3-py2.py3-none-any.whl
```

- PIP3:

```
pip3 uninstall hpeairgaputil-1.3-py2.py3-none-any.whl
```

Using Air Gap Utility Filters

After Installing the Air Gap Utility Package on page 870, you can filter the available apps for a given HPE Ezmeral Runtime Enterprise version in a project.

You must provide one of the following mandatory arguments in each of your commands:

- `--list_releases`
- `--release`

NOTE:

To display a list of options available in the `hpe-airgap-util`, use the following command:

```
hpe-airgap-util --help
```

You can use filters to display the following information:

NOTE:

The system output in the following examples are for illustration only, and might not represent the software available for your release of HPE Ezmeral Runtime Enterprise.

- **Release:** List all releases with the following command:

```
hpe-airgap-util --list_releases
```

For example:

```
hpe-airgap-util --list_releases
----------------------------
ERE          Kubernetes    Release
----------------------------

5.4.1        NA            GA
5.5.0-102    1.0.1          RC

----------------------------
```

- **Images:** List all images for a particular release:

```
hpe-airgap-util --release <release-number>
```

For example:

```
hpe-airgap-util --release 5.4
INFO: Found 263 repositories to process.
+------------------------------------------------------------------------------------

|
repository
|                                 digest
| size(mb) |       component       | requirement |  license   |
+------------------------------------------------------------------------------------

|
postgres:9.5                                      |
sha256:78bdf72abdd619368cd22fd6372553f4ece87b2ca0f8f9fa4a2ab0e3a3932c36
|  72.78   |        airflow        |  optional   | enterprise |
|                       pbweb/
airflow-prometheus-exporter:latest                        |
sha256:8e61f0b7980eb672b1f9a250153ce1f55135b9af5c3f594aa3e4c0a847889766
|   6.39   |        airflow        |  optional   | enterprise |
|                               k8s.gcr.io/
volume-nfs:0.8                               |
sha256:3899ca782a272608fb4139eca436e87592eb779ae76adad6f8e0080365d57de0
|  88.99   |        airflow        |  optional   | enterprise |
|                          k8s.gcr.io/git-sync/
git-sync:v3.3.4                          |
sha256:866599ca98bcde1404b56152d8601888a5d3dae7fc21665155577d607652aa09
|  56.47   |        airflow        |  optional   | enterprise |
...
...
...
|                            bluedata/
kd-livy:050-5.4.0-1.1                             |
sha256:23ddfe633e18ba431d4794097e919760d4c7c6eec89a4032750c436f521fdb0e
|  331.72  |         spark         |  optional   | analytics  |
|                        bluedata/
kd-hivemetastore:238-5.4.0-1.1                        |
sha256:08d22f25191902f54c37d05bca42e03ceab03dff9cdc45e88f36e080fcc72ec1
|  331.72  |         spark         |  optional   | analytics  |
+------------------------------------------------------------------------------------
```

- List available images without headers:

  ```
  hpe-airgap-util --release <release-number> --noheaders
  ```

- List all required images:

  ```
  hpe-airgap-util --release <release-number> --required
  ```

- List all optional images:

  ```
  hpe-airgap-util --release <release-number> --optional
  ```

last-updated: May 06, 2024

- **List components:** List all the components that are available for a particular release:

```
hpe-airgap-util --list_components --release <release-number>
```

For example:

```
hpe-airgap-util --list_components --release 5.4
INFO: Found 285 repositories to process.
+-----------------------+
| components            |
+-----------------------+
| airflow               |
| argocd                |
| datafabric            |
| falco                 |
| hpecp-agent           |
| hpecp-monitoring      |
| hpecp-nvidiagpubeat   |
| hpecp-serviceaccounts |
| istio                 |
| kube-state-metrics    |
| kubedirector          |
| kubeflow              |
| kubernetes            |
| kubernetes-dashboard  |
| metrics-server        |
| mlops                 |
| nvidia-plugin         |
| opa-gatekeeper        |
| spark                 |
| spark-operator        |
+-----------------------+
```

- **Component:** List all images for a particular component:

```
hpe-airgap-util --release <release-number> --component <component>
```

For example:

```
hpe-airgap-util --release 5.4 --component hpecp-agent
INFO: Found 263 repositories to process.
+-----------------------------------------------+-------------------------------------

|                  repository
|                                  digest
| size(mb) |  component  | requirement |  license  |
+-----------------------------------------------+-------------------------------------

|          bluedata/hpecp-dtap:1.8.0           |
sha256:3d61f2a5c56da15e4002e720e6ab83103673ef9c4ae39834ca2e692f8fe334a1
|  250.42  | hpecp-agent |  required   | essential |
|         bluedata/hpecp-fsmount:1.1.1         |
sha256:42ba5577fccc7340dd4c58a748ef5ad3465362b77b0e38325e7d720bb421aa6b
|  88.04   | hpecp-agent |  required   | essential |
|          bluedata/hpecp-agent:1.2.1          |
sha256:a7d25b4a777f0f2db842a4a6dbef88cf46a73d576d7facc12a7ed83796236f24
|  177.31  | hpecp-agent |  required   | essential |
|      bluedata/hpecp-bootstrap-tools:0.4      |
sha256:501ffa0dfda7a277717158c81441c54fa401ae4aa3ae08c90201bb99e232b998
|  221.02  | hpecp-agent |  required   | essential |
| bluedata/hpecp-bootstrap-hpecp-agent:1.2.1-3 |
sha256:c9fc9084b3904c79f3566c4ea440efc8229bcf0187569144232072d8ce7ed14b
|  128.54  | hpecp-agent |  required   | essential |
+-----------------------------------------------+-------------------------------------
```

- **Size:** Valid values include `b`, `kb`, `mb`, and `tb`.

- Display images less than a certain size:

```
hpe-airgap-util --release <release-number> --lessthan 1mb
```

For example:

```
hpe-airgap-util --release 5.4 --lessthan 1mb
INFO: Found 263 repositories to process.
+-----------------------------------+-------------------------------------------

|            repository             |
digest                                   | size(mb) | component  |
requirement |  license   |
+-----------------------------------+-------------------------------------------

| gcr.io/mapr-252711/busybox:latest |
sha256:31a54a0cf86d7354788a8265f60ae6acb4b348a67efbcf7c1007dd3cf7af05ab
|   0.77   | datafabric |  optional   | enterprise |
|          busybox:latest           |
sha256:b69959407d21e8a062e0416bf13405bb2b71ed7a84dde4158ebafacfa06f5578
|   0.77   |  kubeflow  |  optional   |   mlops    |
|      k8s.gcr.io/pause:3.4.1       |
sha256:9ec1e780f5c0196af7b28f135ffc0533eddcb0a54a0ba8b32943303ce76fe70d
|   0.30   | kubernetes |  required   | essential  |
|       k8s.gcr.io/pause:3.2        |
sha256:4a1c4b21597c1b4415bdbecb28a3296c6b5e23ca4f9feeb599860a1dac6a0108
|   0.30   | kubernetes |  required   | essential  |
+-----------------------------------+-------------------------------------------
```

- Display images greater than a certain size:

```
hpe-airgap-util --release <release-number> --greaterthan 100mb
```

For example:

```
hpe-airgap-util --release 5.4 --greaterthan 10gb
INFO: Found 263 repositories to process.
+----------------------------------------+----------------------------------------

|               repository
|
digest                                   | size(mb) | component |
requirement | license |
+----------------------------------------+----------------------------------------

| bluedata/kd-deployment-api-serving:1.0 |
sha256:1b5062ca915f0a846d640e58a30b3231761fa4a03a6ed1b1fce5317149fe22c8
| 11885.33 |   mlops   |  optional   |  mlops  |
|  bluedata/kd-training-api-serving:1.0  |
sha256:efd5e86e270d9dc775fd04b17aff5465b48c0c3693feb313a6bd9f17780caab8
| 11876.42 |   mlops   |  optional   |  mlops  |
|        bluedata/kd-notebook:3.1        |
sha256:07234b781ce21518da396ccfec9708edfd2dea463ada0f02098ddbee0a0cc04c
| 14777.13 |   mlops   |  optional   |  mlops  |
+----------------------------------------+----------------------------------------
```

- Display images between two sizes:

```
hpe-airgap-util --release <release-number> --lessthan 1gb --greaterthan
500mb
```

- **License:** view all images for a particular license.

  Valid values include `essential`, `enterprise`, `analytics`, and `mlops`.

  For example:

```
hpe-airgap-util --release <release-number> --license enterprise
```

- You can **combine filters** to provide a more customized query.

  For example:

```
hpe-airgap-util --release <release-number> --component falco
```

```
hpe-airgap-util --release 5.4 --component falco
repo_name
digest
      size(mb)                addon      required|optional
license
----------------------------------------
 ---------------------------------------------------------------------------
       ----------               -------    -------------------
 ----------
falcosecurity/falcosidekick-ui:v1.1.0
sha256:1ad2a5ecf3eae9b760f0f2467a1add7e0f014760f0f105f25ba7465b4981ca2d
           9.79                 falco       required
enterprise
bluedata/hpecp-bootstrap-tools:0.4:0.5
sha256:72d5b10e34076f542e1dcc9ae749001c9dc01ae5c6962fd66249704918b63eab
           156.99               falco       required
enterprise
falcosecurity/falco-no-driver:0.29.1
sha256:d51f4a74c3c0b10139cf490e32b685d733e878720c5f925f60552058114fd077
           6.46                 falco       required
enterprise
falcosecurity/falcosidekick:2.23.1
sha256:e3b7473f25a7e631fe82b15ac6cbb88f7c203a8064db5baca6ff1fa634b392cc
           27.02                falco       required
enterprise
bluedata/hpecp-bootstrap-falco:0.29.1-2
sha256:b2de01af4c77dcb90fcc2d9ab5a16b1ae4c15a03a64bd5545790e3807e17cb26
           52.05                falco       required
enterprise
```

- To filter for a **specific name or string**, you can use the options `--noheaders | grep <String>`.

  For example:

```
hpe-airgap-util --release <release-number> --noheaders | grep bootstrap
```

```
hpe-airgap-util --release 5.4 --noheaders | grep bootstrap
bluedata/hpecp-bootstrap-hpecp-agent:1.2.1-0
sha256:d95558301b629c6cc286e045b269324d4197596b1ae0da55782d41b50fb14992
61.63  hpecp-agent       required  enterprise
bluedata/hpecp-bootstrap-hpecp-monitoring:6.6.5-8.0
sha256:151a57f2c4b3b7ca5319099e2de9cbab06d799ee5595fba31c8d49a49e419bdb
51.35  hpecp-monitoring  required  enterprise
bluedata/hpecp-bootstrap-velero:1.6.3-2
sha256:ee57acaf232ceb944030f722eeecb70df92572c10cb2a7069b70e8b3cf5ce200
78.55  velero            optional  enterprise
bluedata/hpecp-bootstrap-argocd:2.1.2-2
sha256:e09637ede315a2e0b0faf3012dfd1ada80fdac71291f998273093ff683435d6e
347.66  argocd            required  enterprise
bluedata/hpecp-bootstrap-tools:0.4
sha256:501ffa0dfda7a277717158c81441c54fa401ae4aa3ae08c90201bb99e232b998
221.02  hpecp-agent       required  enterprise
bluedata/hpecp-bootstrap-tools:0.4:0.5
sha256:72d5b10e34076f542e1dcc9ae749001c9dc01ae5c6962fd66249704918b63eab
156.99  falco             required  enterprise
bluedata/hpecp-bootstrap-falco:0.29.1-2
sha256:b2de01af4c77dcb90fcc2d9ab5a16b1ae4c15a03a64bd5545790e3807e17cb26
52.05  falco             required  enterprise
```

Downloading Air Gap Files

After Using Air Gap Utility Filters on page 871 to find the necessary files for your deployment, download the files as follows:

1. Use a single command to filter and copy air gap files to a local filesystem or remote registry. Include all filters you want to apply to your download.

   Include `--dest_compress` to compress the files and download in a .tgz file. Otherwise, the files download in a .tar file. For example:

```
hpe-airgap-util --release <release-number> --lessthan
1mb --copy --dest_path images/ --dest_compress
```

   Use `--force` to delete the .tgz or .tar file of the image if it already exists. For example:

```
hpe-airgap-util --release <release-number> --lessthan
1mb --copy --dest_path images/ --force
```

```
hpe-airgap-util --release <release-number> --lessthan
1mb --copy --dest_path images/ --dest_compress --force
```

- To copy **multiple images** to a local filesystem, execute the following command. Provide the destination path where you want to store your files.

```
hpe-airgap-util --release <release-number>
<add-on_filters> --copy --dest_path <destination-path>
```

For example:

```
hpe-airgap-util --release 5.4 --component
hpecp-agent --copy --dest_path /home/user/image
INFO: Found 263 repositories to process.
+-----------------------------------------------+-----------------------------------

|                  repository
|
digest                                    | size(mb) |  component  |
requirement |  license  |
+-----------------------------------------------+-----------------------------------

|          bluedata/hpecp-dtap:1.8.0           |
sha256:3d61f2a5c56da15e4002e720e6ab83103673ef9c4ae39834ca2e692f8fe334a1
 |  250.42  | hpecp-agent |  required   | essential |
|         bluedata/hpecp-fsmount:1.1.1         |
sha256:42ba5577fccc7340dd4c58a748ef5ad3465362b77b0e38325e7d720bb421aa6b
 |  88.04   | hpecp-agent |  required   | essential |
|          bluedata/hpecp-agent:1.2.1          |
sha256:a7d25b4a777f0f2db842a4a6dbef88cf46a73d576d7facc12a7ed83796236f24
 |  177.31  | hpecp-agent |  required   | essential |
|      bluedata/hpecp-bootstrap-tools:0.4      |
sha256:501ffa0dfda7a277717158c81441c54fa401ae4aa3ae08c90201bb99e232b998
 |  221.02  | hpecp-agent |  required   | essential |
| bluedata/hpecp-bootstrap-hpecp-agent:1.2.1-3 |
sha256:c9fc9084b3904c79f3566c4ea440efc8229bcf0187569144232072d8ce7ed14b
 |  128.54  | hpecp-agent |  required   | essential |
+-----------------------------------------------+-----------------------------------

INFO: Processing artifact bluedata/hpecp-dtap:1.8.0 ...
INFO: Copying artifact to /home/user/image/
bluedata_hpecp-dtap_1.8.0.tar
Getting image source signatures
Copying blob 8ba884070f61 done
Copying blob 71d2b71667ea done
Copying blob 10bb33b8b168 done
Copying blob 1204286a37b7 done
...
...
Storing signatures
INFO: Processing artifact bluedata/hpecp-fsmount:1.1.1 ...
INFO: Copying artifact to /home/user/image/
bluedata_hpecp-fsmount_1.1.1.tar
Getting image source signatures
Copying blob 29291e31a76a done
Copying blob 615a2023df20 done
...
...
Storing signatures
INFO: Processing artifact bluedata/hpecp-agent:1.2.1 ...
INFO: Copying artifact to /home/user/image/
bluedata_hpecp-agent_1.2.1.tar
Getting image source signatures
Copying blob ed5dc850ecaf done
Copying blob 79f8ae5118d4 done
```

```
...
...
Storing signatures
INFO: Processing artifact bluedata/hpecp-bootstrap-tools:0.4 ...
INFO: Copying artifact to /home/user/image/
bluedata_hpecp-bootstrap-tools_0.4.tar
Getting image source signatures
Copying blob 3aa8b87b7f88 done
Copying blob 57584b59d88b done
Copying blob 135070274eb1 done
...
...
Storing signatures
INFO: Processing artifact bluedata/
hpecp-bootstrap-hpecp-agent:1.2.1-3 ...
INFO: Copying artifact to /home/user/image/
bluedata_hpecp-bootstrap-hpecp-agent_1.2.1-3.tar
Getting image source signatures
Copying blob ed5dc850ecaf done
Copying blob 79f8ae5118d4 done
Copying blob e17b1b3fe6d3 done
...
...
Copying config 0168331b65 done
Writing manifest to image destination
Storing signatures
```

- To copy a **single image** to a local filesystem, execute the following command. Provide the destination path where you want to store your files.

```
hpe-airgap-util --release <release-number> --image
<image-name> --copy --dest_path <destination-path>
```

- To copy **multiple images** to a remote container registry, select one of the following options. Provide the destination URL and credentials for your container registry.
  - Use the `--dest_creds <username:password>` command line option:

    ```
    hpe-airgap-util --release <release-number>
    <add-on-filters> --copy --dest_url <destination-url> --dest_creds
    <username:password>
    ```

  - Alternatively, set environment variable `AIRGAP_UTIL_CREDS`. You can set environmental variables using the `export` command:

    ```
    export AIRGAP_UTIL_CREDS=<username>:<password>
    ```

  For example:

```
hpe-airgap-util --release 5.4  --component
hpecp-agent --copy --dest_url my.local.registry.com/airgap-ecp54
INFO: Found 263 repositories to process.
+-----------------------------------------------+-----------------------------------

|                  repository
|
digest                                     | size(mb) |  component  |
requirement |  license  |
+-----------------------------------------------+-----------------------------------
```

```
|          bluedata/hpecp-dtap:1.8.0           |
sha256:3d61f2a5c56da15e4002e720e6ab83103673ef9c4ae39834ca2e692f8fe334a1
 |  250.42  | hpecp-agent |  required   | essential |
|         bluedata/hpecp-fsmount:1.1.1         |
sha256:42ba5577fccc7340dd4c58a748ef5ad3465362b77b0e38325e7d720bb421aa6b
 |  88.04   | hpecp-agent |  required   | essential |
|          bluedata/hpecp-agent:1.2.1          |
sha256:a7d25b4a777f0f2db842a4a6dbef88cf46a73d576d7facc12a7ed83796236f24
 |  177.31  | hpecp-agent |  required   | essential |
|      bluedata/hpecp-bootstrap-tools:0.4      |
sha256:501ffa0dfda7a277717158c81441c54fa401ae4aa3ae08c90201bb99e232b998
 |  221.02  | hpecp-agent |  required   | essential |
| bluedata/hpecp-bootstrap-hpecp-agent:1.2.1-3 |
sha256:c9fc9084b3904c79f3566c4ea440efc8229bcf0187569144232072d8ce7ed14b
 |  128.54  | hpecp-agent |  required   | essential |
+----------------------------------------------+-----------------------------------

INFO: Processing artifact bluedata/hpecp-dtap:1.8.0 ...
INFO: Copying artifact to my.local.registry.com/airgap-ecp54/bluedata/
hpecp-dtap:1.8.0
Getting image source signatures
Copying blob 8ba884070f61 done
Copying blob 71d2b71667ea done
Copying blob 10bb33b8b168 done
Copying blob 1204286a37b7 done
...
...
Storing signatures
INFO: Processing artifact bluedata/hpecp-fsmount:1.1.1 ...
INFO: Copying artifact to my.local.registry.com/airgap-ecp54/bluedata/
hpecp-fsmount:1.1.1
Getting image source signatures
Copying blob 29291e31a76a done
Copying blob 615a2023df20 done
...
...
Storing signatures
INFO: Processing artifact bluedata/hpecp-agent:1.2.1 ...
INFO: Copying artifact to my.local.registry.com/airgap-ecp54/bluedata/
hpecp-agent:1.2.1
Getting image source signatures
Copying blob ed5dc850ecaf done
Copying blob 79f8ae5118d4 done
...
...
Storing signatures
INFO: Processing artifact bluedata/hpecp-bootstrap-tools:0.4 ...
INFO: Copying artifact to my.local.registry.com/airgap-ecp54/bluedata/
hpecp-bootstrap-tools:0.4
Getting image source signatures
Copying blob 3aa8b87b7f88 done
Copying blob 57584b59d88b done
Copying blob 135070274eb1 done
...
...
Storing signatures
INFO: Processing artifact bluedata/
hpecp-bootstrap-hpecp-agent:1.2.1-3 ...
INFO: Copying artifact to my.local.registry.com/airgap-ecp54/bluedata/
hpecp-bootstrap-hpecp-agent:1.2.1-3
Getting image source signatures
Copying blob ed5dc850ecaf done
Copying blob 79f8ae5118d4 done
```

```
Copying blob e17b1b3fe6d3 done
...
...
Copying config 0168331b65 done
Writing manifest to image destination
Storing signatures
```

- To copy a **single image** to a remote container registry, execute the following command. Provide the destination URL and credentials for your container registry.

```
hpe-airgap-util --release <release-number> --image
<image-name> --copy --dest_url <destination-url> --dest_creds
<username:password>
```

Air Gap Utility Logging

By default, the Air Gap Utility creates a `logs/` directory in the present working directory from which you invoked the Air Gap Utility command line.

You can change the log directory location as follows:

- If you pass the `--logdir` argument in the Air Gap Utility command line, then the Air Gap Utility creates a `logs/` directory in the path provided in the `--logdir` arguement.
- If you set the `AIRGAP_UTIL_LOGDIR` environment variable, but do not pass the `--logdir` argument in the Air Gap Utility command line, then the Air Gap utility creates a `logs/` directory in the path set in the `AIRGAP_UTIL_LOGDIR` environment variable.

NOTE: The Air Gap Utility does not create log files when commands are run in `TTY` mode. For example:

```
hpe-airgap-util --release 5.4 | grep -i argocd
```

Related tasks

Kubernetes Worker Installation Overview on page 528
Describes how to add a host to HPE Ezmeral Runtime Enterprise as a Kubernetes worker for compute workloads.

Related reference

HPE Ezmeral Runtime Enterprise Air Gap Utility Release Notes on page 53
Change history and version compatibility information for the HPE Ezmeral Runtime Enterprise Air Gap Utility, `hpe-airgap-util`, on HPE Ezmeral Runtime Enterprise.

Air Gap Tab on page 799
The **Air Gap** tab of the **System Settings** screen enables the Platform Administrator to specify settings to be used when the Kubernetes hosts, clusters, and tenants do not have connectivity to the Internet.

More information

Kubernetes Air-Gap Requirements on page 834
Configuring Air Gap Kubernetes Host Settings on page 868
Download image and RPM files and configure settings for Kubernetes hosts in an air-gapped environment.

Validating the Installation

The first post-installation step is to perform some basic tests to validate the installation. If these tests pass, then the deployment can be considered ready for use. Proceed to the following topics as appropriate for your installation:

- **Big Data (Kubernetes):** See Big Data (Kubernetes).
- **AI/ML (Kubernetes):** See AI/ML (Kubernetes).

If your deployment includes more than one of the above options, then follow the instructions in all applicable sections.

Big Data (Kubernetes)

To validate Big Data functionality with Kubernetes, create and test a Kubernetes cluster and Big Data tenant, as described in Getting Started with General Kubernetes Functionality. Once you have completed this process, you will be ready to begin running Big Data jobs using the Kubernetes functionality within HPE Ezmeral Runtime Enterprise!

AI/ML (Kubernetes)

To validate AI/ML functionality with Kubernetes, create and test a Kubernetes cluster and AI/ML project, as described in HPE Ezmeral ML Ops on page 148. Once you have completed this process, you will be ready to begin running AI/ML jobs using the Kubernetes functionality within HPE Ezmeral Runtime Enterprise!

Using the Built-In Config Checks

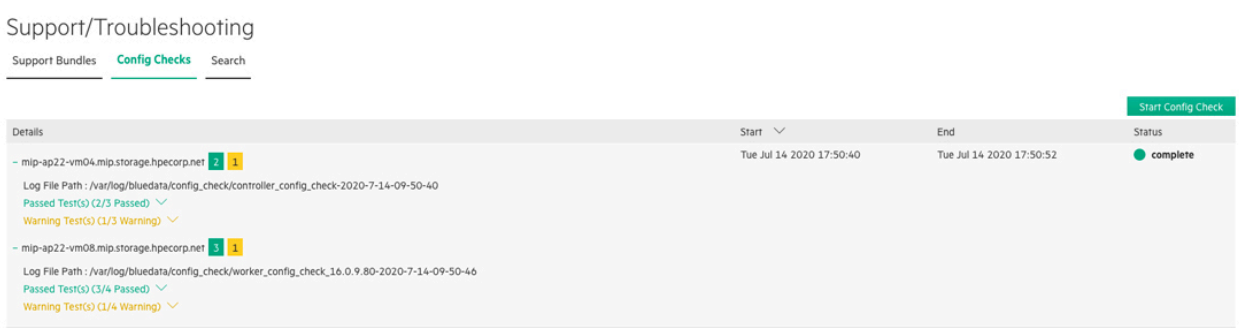
HPE Ezmeral Runtime Enterprise includes a set of built-in configuration checks. To use these checks:

1. Click the **Help** button in the **Toolbar**, and then select **Support** in the pull-down menu.

   The **Support/Troubleshooting** screen appears with the **Support Bundles** tab selected.

2. Select the **Config Checks** tab.
3. Click the **Start Config Check** button.

A series of configuration checks will take place, and the results will be reported in the **Config Checks** tab.



In the results:

- A number in a green square indicates the number of successful checks performed on a host.
- A number in an orange square indicates the number of checks that ended with a warning status on a host.
- A number in a red square indicates the number of failed checks performed on a host.
- Clicking a hostname toggles expanding or collapsing the test results for the selected host.
- Clicking a down arrow next to **Passed Test(s)**, **Warning Test(s)**, or **Failed Test(s)** entry expands the details for that entry.

- Clicking an up arrow next to a **Passed Test(s)**, **Warning Test(s)**, or **Failed Test(s)** entry collapses the details for that entry.

Kubernetes Worker Installation Overview
Describes how to add a host to HPE Ezmeral Runtime Enterprise as a Kubernetes worker for compute workloads.

Prerequisites

- Hewlett Packard Enterprise recommends enabling platform High Availability before adding a large number of Kubernetes.
- Ensure that hosts conform to the requirements described in Host Requirements on page 813 and Kubernetes Host/Node Requirements on page 833.

  If the `firewalld` service is installed and enabled on the Controller, and the `firewalld` service is installed and enabled on all hosts before they are added to the deployment, the installer for HPE Ezmeral Runtime Enterprise automatically configures firewall rules to open the required ports listed in Port Requirements on page 809 and Kubernetes Port Requirements on page 836.

  CAUTION:

  Numerous configuration changes occur to the host during installation that are required in order for the platform to function. These changes are not completely reversible and may impact any other applications and processes that are currently running on the host. It is strongly recommended that you install HPE Ezmeral Runtime Enterprise on a host that is not being used for any other purpose in order to avoid possible disruptions to your business processes.

  Installing HPE Ezmeral Runtime Enterprise on any host that does not meet all applicable requirements may lead to unpredictable behavior and/or data loss.

- For best results, it is recommended that all compute hosts in a cluster share the same configuration (CPU, RAM, storage, OS, etc.).
- If this host has MIG-enabled GPUs that are supported by HPE Ezmeral Runtime Enterprise, install the NVIDIA driver on the host and configure MIG before adding the host to HPE Ezmeral Runtime Enterprise.

  See Deploying MIG Support on page 840.
- If you want to install the Falco Kernel Module on the host as part of the Falco Container Runtime Security feature, install the module on the host after you install the host OS but before you add the host to HPE Ezmeral Runtime Enterprise.
- See Falco Container Runtime Security on page 499.

About this task

This article describes adding Kubernetes hosts for compute workloads.

- If you visited this article intending to add Data Fabric nodes, see Kubernetes Data Fabric Node Installation Overview on page 531.
- If you visited this article intending to add Shadow Controller or Arbiter hosts, see Enabling Platform High Availability on page 740.
- If you visited this article intending to add a gateway host, see Installing a Gateway Host on page 758.

Procedure

1. Prepare the hosts to be added as Kubernetes hosts.
   - If your environment is running the SSHD service (see Configuration Requirements on page 826), then skip to Kubernetes Host: Add the Public SSH Key on page 537.
   - If your environment does not allow key-based SSH login, then proceed to Agent-Based Kubernetes Host Installation on page 532.
2. If you are adding hosts to expand a Kubernetes cluster that has not been migrated to the Hewlett Packard Enterprise distribution of Kubernetes, create the following touch file on each host:

   ```
   touch /tmp/k8s_docker_override
   ```

   Creating the touch file specifies that the Docker container runtime is used instead of the containerd runtime.
3. In the web interface, select the hosts to add as Kubernetes Workers.

   See Kubernetes Host: Select the Hosts on page 538.
4. Add the Worker hosts.

   See Kubernetes Host: Add the Hosts on page 539.
5. Select the hard drives on the Worker hosts.

   See Kubernetes Host: Select Hard Drives on page 540.
6. Place HPE Ezmeral Runtime Enterprise into Lockdown mode.

   For more information, see Kubernetes Host: Enter Lockdown Mode on page 542.
7. Install the hosts as Kubernetes Workers.

   See Kubernetes Host: Add the Hosts as Workers on page 542.

   HPE Ezmeral Runtime Enterprise verifies that the number of CPU cores in the hosts do not exceed the licensed maximum, and then proceeds with the installation. The UI displays a green **Installing** bar for each of the new hosts.
8. Exit Lockdown mode.
9. On each host, prevent the `yum update` command from updating the Kubernetes repo by setting `enabled=0` in the following file: `/etc/yum.repos.d/bd-kubernetes.repo`
10. Validate that the new Kubernetes Worker has been correctly added and is functioning properly.

    See Kubernetes Host: Validate the Worker Installation on page 543.

Related reference

Deploying MIG Support on page 840
This topic describes how to configure and deploy a supported MIG-enabled GPU on HPE Ezmeral Runtime Enterprise.

Air Gap Tab on page 799
The **Air Gap** tab of the **System Settings** screen enables the Platform Administrator to specify settings to be used when the Kubernetes hosts, clusters, and tenants do not have connectivity to the Internet.

More information

Falco Container Runtime Security on page 499

The Falco Container Runtime Security feature of HPE Ezmeral Runtime Enterprise improves container security and threat detection.

Kubernetes Air-Gap Requirements on page 834

Using the Air Gap Utility on page 869
Describes how to use the air gap utility to download files in an air-gapped HPE Ezmeral Runtime Enterprise environment.

Licensing Your Deployment

Prerequisites

- You have purchased a license and obtained the license file.

  To purchase a license, contact Hewlett Packard Enterprise.

- If this is an HPE Ezmeral Runtime Enterprise Essentials deployment, ensure that you have HPE Ezmeral Runtime Enterprise Essentials license. Except for the HPE Ezmeral Instant-On license, after license that includes the full-featured HPE Ezmeral Runtime Enterprise license is uploaded, the deployment cannot be changed to HPE Ezmeral Runtime Enterprise Essentials.

- **Required access rights:** Platform Administrator

About this task

The deployment automatically installs the HPE Ezmeral Instant-On evaluation license for an unlimited number of CPU cores. The HPE Ezmeral Instant-On is valid for 30 days from the installation date. Before the license expires, add a purchased license.

CAUTION:

If the HPE Ezmeral Instant-On and all other evaluation licenses expire before before a purchased license has been applied, then the deployment will go into Lockdown mode (see Lockdown Mode on page 916). The Platform Administrator will not be able to exit Lockdown mode until a purchased license is applied.

Procedure

To add a license, you upload the license file from the**License** tab of the **System Settings** screen.

For more information about licenses, see Licensing.

Upgrading to HPE Ezmeral Runtime Enterprise 5.6.x

This article describes the process to upgrade to the latest 5.6.x version of HPE Ezmeral Runtime Enterprise.

IMPORTANT:

Before upgrading from a previous version of HPE Ezmeral Runtime Enterprise, read the **Upgrade** and **Issues and Workarounds** sections in the Release Notes for any known issues that might apply to your upgrade scenario.

Upgrade Paths

When you upgrade to HPE Ezmeral Runtime Enterprise release 5.6, you are installing HPE Ezmeral Runtime Enterprise 5.6.4, which is the latest public release of HPE Ezmeral Runtime Enterprise 5.6. x

IMPORTANT:

- Before upgrading to HPE Ezmeral Runtime Enterprise 5.6.x, HPE Ezmeral Product and Engineering team recommends upgrading all pre-5.5.1 deployments to HPE Ezmeral Runtime Enterprise 5.5.1, and to perform EzKube migration for the pre-5.5.1 Kubernetes clusters. Contact the HPE Support team for any questions related to HPE Ezmeral Runtime Enterprise and Kubernetes support.

You can upgrade directly from the following previous versions of HPE Ezmeral Runtime Enterprise only:

- 5.6.2
- 5.6.3

IMPORTANT:

EPIC deployments cannot be upgraded to a release later than HPE Ezmeral Runtime Enterprise version 5.4.1.

Upgrading Kubernetes Bundles

Beginning with HPE Ezmeral Runtime Enterprise 5.5.0, HPE decouples the upgrade of the HPE Ezmeral Runtime Enterprise platform from Kubernetes-related components.

NOTE:

By upgrading to HPE Ezmeral Runtime Enterprise 5.5.0 or later, you will automatically get the latest available Kubernetes versions.

Starting from the next release that follows HPE Ezmeral Runtime Enterprise 5.5.0, Kubernetes bundles can be upgraded without performing the platform upgrade.

For more details, see Upgrading Kubernetes Bundles on page 903.

Air-Gapped Kubernetes Deployment

An air-gapped Kubernetes deployment refers to a deployment in which Kubernetes hosts, clusters, and tenants do not have connectivity to the Internet. Air-gapped deployments are also called disconnected sites or dark sites.

For information about requirements for air-gapped environments, see Kubernetes Air-Gap Requirements on page 834.

Impact on Workloads During Platform Upgrade

During an upgrade of the HPE Ezmeral Runtime Enterprise, workloads are affected as follows:

Kubernetes Clusters and workloads	• New Kubernetes clusters cannot be created during the control plane upgrade process. Workloads running on existing Kubernetes clusters or within a tenant namespace will be impacted when the Container Runtime is restarted. Service endpoints will be inaccessible due to the unavailability of the authentication proxy.

	• Any new service points (e.g. NodePort services) that are in flight for port remapping via a Gateway host will be queued up until the control plane upgrade completes. Affected service endpoints will be remapped via a Gateway host and made available for users when the control plane resumes operating. • All existing service endpoints across one or more Kubernetes clusters and/or tenants will continue to serve traffic during the control plane upgrade except for a brief (approximately 5 seconds) interval while the `haproxy` component restarts during the Gateway upgrade. • Kubernetes users across all roles (e.g. Cluster Administrator or Member) will not be able to interact with the Kubernetes API server via either kubectl, the web interface, or any other means during the upgrade process. Terminal access to Kubernetes cluster is still possible via SSH, assuming that administrators can access Kubernetes nodes. • Cluster manipulation activities such as upgrading existing kubernetes clusters or expanding master/worker nodes are not allowed until the upgrade completes.
Non-Kubernetes virtual nodes and workloads	• Existing non-Kubernetes virtual nodes/containers will be down and/or inaccessible during the upgrade process because all Docker processes will be “paused” (or stopped) for all “active” virtual clusters before the upgrade starts. • All orchestration activities (e.g. expand/shrink, ActionScripts, Gateway port-mapping, node migration, etc.) will be unavailable until the upgrade completes. • Tasks are not queued during the upgrade; you must reinitiate any such tasks after the upgrade.

Upgrade Process Summary

The following is a summary of the process to upgrade to HPE Ezmeral Runtime Enterprise and its related components. Detailed procedures are provided in the tasks linked to by this summary.

1. Before you upgrade the platform, you might need to upgrade host software to a version that is supported both on the HPE Ezmeral Runtime Enterprise version you are running and on the HPE Ezmeral Runtime Enterprise version to which you are upgrading.

   a. Upgrade host operating system software on each host.

   b. If using an air-gapped environment, Configure or change Kubernetes air gap settings.

   c. Upgrade Kubernetes versions on each Kubernetes cluster. Because Kubernetes versions must be upgraded one at a time, you might need to perform multiple upgrades.

   Other tasks you perform before upgrading the software include obtaining the software and executing pre-check scripts on all hosts.

2. If your deployment includes hosts that have GPUs, you must remove the hosts from the Kubernetes clusters and remove the hosts from HPE Ezmeral Runtime Enterprise. You complete this task in Before Upgrading the Platform on page 889.

   IMPORTANT:

   HPE Ezmeral Runtime Enterprise adds support for MIG-enabled GPUs. For all GPUs to be recognized by the system after the upgrade, all hosts that have GPUs must be removed from HPE Ezmeral Runtime Enterprise before the upgrade, and then added back to the configuration after the upgrade process is complete. This requirement applies to all GPUs, including those GPUs are that are not MIG-enabled.

3. If your environment includes **HPE Ezmeral Data Fabric on Kubernetes** clusters, proceed to Upgrade HPE Ezmeral Data Fabric on Kubernetes.

   If your environment implements Embedded Data Fabric, there are no specific upgrade tasks to perform for that feature. You can procede to upgrading the platform software.

   IMPORTANT:

   A deployment of HPE Ezmeral Runtime Enterprise can include one Data Fabric cluster that implements **HPE Ezmeral Data Fabric on Kubernetes**, **or** the deployment can include an Embedded Data Fabric, but not both (see Storage on page 804). If this deployment already includes an Embedded Data Fabric, do not attempt to add another Kubernetes Data Fabric cluster.

4. Upgrade the platform software on the Controller host.

5. If you want to install Falco Kernel modules on your Kubernetes hosts, consider installing them before you upgrade the Kubernetes add-ons. See Install the Falco Kernel modules on each Kubernetes host.

6. If you want to upgrade Kubernetes add-ons, see Upgrade Kubernetes add-ons on each Kubernetes cluster.

7. If you want to upgrade Kubernetes clusters to later versions, see Upgrade Kubernetes clusters to later versions of Kubernetes. HPE Ezmeral Runtime Enterprise supports multiple versions of Kubernetes. Depending on the applications you have installed or plan to install, you might want to upgrade Kubernetes to a later version than the version you installed before the platform upgrade.

8. If you are using the RHEL 8, find the Data Fabric hosts and reboot those hosts.

   NOTE: You can see the Data Fabric hosts by looking at the **Kubernetes Hosts** page that lists all the hosts. The Data Fabric hosts have `Data fabric: true` tag.

9. Restart HPE Ezmeral Data Fabric on Kubernetes cluster. See Restarting the Data Fabric Cluster on page 620 for details.

10. On the Controller node, re-establish the tenant mount, using the following commands:

```
ERTS_PATH=/opt/bluedata/common-install/bd_mgmt/erts-*/bin
NODETOOL=/opt/bluedata/common-install/bd_mgmt/bin/nodetool
NAME_ARG=`egrep '^-s?name' $ERTS_PATH/../../releases/1/vm.args`
RPCCMD="$ERTS_PATH/escript $NODETOOL $NAME_ARG rpcterms"
$RPCCMD bd_hypervisor_controller_common redo_tenant_storage_mounts
```

11. Verify the upgrade.

12. If you want to add hosts that include GPUs, or you want to upgrade Kubernetes to later versions, do those tasks as part of the Post Upgrade Tasks on page 904.

last-updated: May 06, 2024

Before Upgrading the Platform

This topic describes the tasks that you must complete before you upgrade the HPE Ezmeral Runtime Enterprise software. Hewlett Packard Enterprise highly recommends performing a configuration and upgrade pre-check and resolve issues before upgrading HPE Ezmeral Runtime Enterprise.

Verify Upgrade Path

Verify that the version of HPE Ezmeral Runtime Enterprise that you are upgrading from is a valid starting point when upgrading to HPE Ezmeral Runtime Enterprise 5.4.x. For information about upgrade paths, see Upgrading to HPE Ezmeral Runtime Enterprise 5.6.x on page 885.

Plan for Impact on Workloads

Upgrading to this version of HPE Ezmeral Runtime Enterprise involves multiple tasks, some of which require node reboots or pod restarts. See Upgrading to HPE Ezmeral Runtime Enterprise 5.6.x on page 885.

Upgrade Kubeflow

If your environment includes Kubeflow and you are upgrading HPE Ezmeral Runtime Enterprise, contact Hewlett Packard Enterprise support for assistance before you begin the upgrade. Several manual steps must be performed to replace the existing version of Kubeflow with the new version of Kubeflow.

Upgrade OS Versions

If your HPE Ezmeral Runtime Enterprise installation is based on an OS version that is not supported by HPE Ezmeral Runtime Enterprise 5.6.x, you must upgrade the OS version to at least the minimum supported version supported by HPE Ezmeral Runtime Enterprise.

For a list of supported operating system versions, see OS Support on page 85.

To upgrade the operating system, see System Maintenance on page 802.

(Optional) Update or Configure Air Gap Settings

If you are using Kubernetes in an air-gapped environment or you want to change your current environment to air gap your Kubernetes objects, configure the air gap settings before you upgrade Kubernetes. Changes to Air gap settings are not propagated to the Kubernetes hosts until the host is rebooted or the Kubernetes version is upgraded.

IMPORTANT: Changing an existing HPE Ezmeral Runtime Enterprise configuration from a non-airgapped environment to an air-gapped environment forces a reinstall of Kubernetes clusters.

If you are changing an existing HPE Ezmeral Runtime Enterprise configuration from a non-airgapped environment to an air-gapped environment, contact Hewlett Packard Enterprise support for assistance before you begin the transition. Several manual steps must be performed to transition to an air-gapped environment.

For more information, see the following:

- Kubernetes Air-Gap Requirements on page 834
- Using the Air Gap Utility on page 869

Upgrade Kubernetes

If your current environment is using Kubernetes, you must update Kubernetes to at least the minimum version supported by this version of HPE Ezmeral Runtime Enterprise. Ensure that the version that you upgrade to is also supported on your current version of HPE Ezmeral Runtime Enterprise.

Kubernetes requires upgrading one version at a time, so you might have to perform this upgrade multiple times until the clusters are running a supported version of Kubernetes.

For information about upgrading Kubernetes, see Upgrading Kubernetes on page 487.

Optionally, you can upgrade to later versions of Kubernetes after all the tasks involved in upgrading HPE Ezmeral Runtime Enterprise, such as upgrading add-ons, are complete.

For a list of supported Kubernetes versions, see Support Matrixes on page 54.

Obtain the HPE Ezmeral Runtime Enterprise Software

Your Hewlett Packard Enterprise representative can provide information about obtaining the correct HPE Ezmeral Runtime Enterprise upgrade package for your environment. You will copy the package bundle to the controller host as part of running the upgrade pre-checks.

Run Configuration and Upgrade Pre-Checks

Hewlett Packard Enterprise highly recommends performing both a configuration check and an upgrade pre-check before upgrading HPE Ezmeral Runtime Enterprise. Ensure that you address any issues reported by these checks before performing the actual upgrade.

1. Verify that all HPE Ezmeral Runtime Enterprise services are operating in **Healthy** (green) status using the **Services** tab of the Platform Administrator **Dashboard** screen. See Dashboard - Platform Administrator on page 570.
2. Copy the upgrade package to the `/srv/bluedata/bundles` folder on the Controller host.
3. Execute the command `chmod 770 <bin-file-name>`, where `<bin-file-name>` is the full name of the package that you copied in Step 2.
4. Verify that the upgrade package appears in the **Available Upgrades** tab.
5. Run the configuration check as described in Config Checks Tab.
6. Review the output of this check, and resolve any errors.
7. Download the `hpe-cp-prechecks-<version>.bin` script to each host, where `<version>` is the version number, such as `5.5`.

RHEL/CentOS	5.6.1 RHEL/CentOS Pre-check script
SLES	5.6.1 SLES Pre-check script

8. On one of the hosts, execute the command `<bin_file> --upgrade`, where `<bin_file>` is the complete name of the .bin file.
9. Review the script output and resolve any errors.
10. Repeat Steps 8 and 9 on each of the remaining hosts in HPE Ezmeral Runtime Enterprise.

The upgrade pre-check script returns output that is similar to the output shown in the following table. The **Error Resolution** column of the table lists the most common errors encountered by each check, along with diagnosis and remediation instructions.

Option	Expected Result	Error Resolution

`Checking integrity ... GOOD.`
`Extracting contents ... done.`
`##`
`## ##`
`## ##`
`## ## ## ##`
`## ## ## ##`
`## ## ## ## HPE Software, Inc.`
`## ## ## ## ##`
`## ## ##`
`## ## ## ##`
`## ##`
`HPE Ezmeral Container Platform Enterprise-Docker debug <version> (minimal)`
`Executing UPGRADE (PLHA: [false|true] NODE: <A.B.C.D>`
`Logging to /tmp/bds-<time_stamp>.log`
`Pre-install checks for HPE Ezmeral Container Platform Enterprise-Docker <version>`
`Operating system configuration`		

`Checking OS Family:`	`PASSED`	This check fails if OS type for the installer does not match with the OS. Use the correct installer.
`Checking running kernel version:`	`PASSED`	This check fails if the following kernel versions are not installed: • 2.6.32 or later for CentOS7/Rhel7. • 3.10.0 or later for Rhel8. • 4.12.14 or later for SLES 15. You can upgrade the versions if needed.
`Checking SELinux setting:`	`PASSED`	This check only generates a warning if SELinux is disabled; re-enable if necessary.
`Checking IPtables/ Firewalld configuration:`	`PASSED`	This check fails if either: • `iptables` is configured to run at boot time but is currently stopped. • `iptables` is currently running but is to run at boot time. • If `iptables` is not running for some reason.
`Checking rsyslog setting:`	`PASSED`	This check will fail if either: • `/etc/rsyslog.d` is not included in `rsyslog.conf`. • The `imuxsock` module is not loaded in `rsyslog.con`.
`Checking user and group specified:`	`PASSED`	For non-root installs, this check verifies that the user exists and is part of the specified group.

Option	Expected Result	Error Resolution
`Checking dnsmasq user and group specified:`	`PASSED`	The check fails if user and group specified in `--dnsmasquser` and `--dnsmasq group` does not exist. If needed, you can create user and group.
`Checking cgconfig kernel params:`	`PASSED`	Verify that `cgconfig` is not disabled in the kernel boot parameters. This is for `cgroup` checks.
`Checking for presence of erlang cookie:`	`PASSED`	The check fails if erlang cookie generated by controller is not present.
`Total: 9 -- Failed: 0 -- Warning: 0 -- Forced(success): 0`		
`Checking Monitoring status:`	`PASSED`	The monitoring service must be installed and running correctly.
`Checking HDFS status:`	`PASSED`	HDFS must be installed and running correctly.
`Checking MapR status:`	`PASSED`	MapR must be installed and running correctly.
`Checking BDMGMT status:`	`PASSED`	BDMGMT must be installed and running correctly.
`Checking Data Server status:`	`PASSED`	The data server must be installed and running correctly.
`Total: 5 -- Failed: 0 -- Warning: 0 -- Forced(success): 0`		

`**********`
 `Aggregate tests summary:`
 `Total: 14`
 `Failed: 0`
 `Warning: 0`
 `Forced(success) : 0`
 `Additional information for debugging is written to/tmp/bd_prechecks.<process_id>.log`
 `**********`		

After you are satisfied that the pre-check has completed correctly, do the following:

1. If you have hosts that have GPU devices, remove those hosts from the Kubernetes cluster and then remove the hosts from HPE Ezmeral Runtime Enterprise. See Remove Hosts That Have GPUs on page 893.

2. If your environment includes HPE Ezmeral Data Fabric on Kubernetes and want to upgrade HPE Ezmeral Data Fabric on Kubernetes before you upgrade the HPE Ezmeral Runtime Enterprise software. See Upgrading HPE Ezmeral Data Fabric on Kubernetes on page 894.

3. If your environment does not include HPE Ezmeral Data Fabric on Kubernetes, proceed to upgrade the HPE Ezmeral Runtime Enterprise software as described in Upgrading the Platform Software on page 897.

Remove Hosts That Have GPUs

HPE Ezmeral Runtime Enterprise adds support for MIG-enabled GPUs. For all GPUs to be recognized by the system after the upgrade, all hosts that have GPUs must be removed from HPE Ezmeral Runtime Enterprise before the upgrade, and then added back to the configuration after the upgrade process is complete. This requirement applies to all GPUs, including those GPUs are that are not MIG-enabled.

HPE Ezmeral Runtime Enterprise 5.3.5 and later deploy updated versions of the NVIDIA runtime and other required NVIDIA packages, and has changed the node label used for hosts that have GPU devices. Both of these configurations changes are made to a host at the time that the host is added to HPE Ezmeral Runtime Enterprise. You will add the hosts to HPE Ezmeral Runtime Enterprise as one of the post upgrade tasks.

To remove a host from HPE Ezmeral Runtime Enterprise:

1. Remove the host from the Kubernetes cluster.

   See Expanding or Shrinking a Kubernetes Cluster on page 483.

2. Delete the host from HPE Ezmeral Runtime Enterprise.

   See Decommissioning/Deleting a Kubernetes Host on page 555.

Before Starting the Upgrade

Before proceeding to HPE Ezmeral Runtime Enterprise 5.5.0 upgrade, you must consider the following:

- Ensure that all Kubernetes clusters are updated to either 1.20.x or 1.21.x.
- HPE Ezmeral Runtime Enterprise upgrade will fail on the controller if:
  - Installation has EPIC virtual clusters.
  - Installation has `Exthosts` configured.
  - Installation has more than three EPIC workers (controller, shadow, arbiter).

    NOTE: Gateway hosts are not considered as EPIC workers. So this limitation does not apply for Gateway hosts.

- By default, all existing EPIC tenants will be deleted during upgrade. You may lose data stored in tenant storage for these tenants. You can modify this behaviour using the following command on the primary controller, before starting the upgrade:

  ```
  echo "bd_mgmt_config:update(bds_cleanup_tenant, false)." >>/opt/bluedata/
  common-install/bd_mgmt/tmp.w
  ```

  IMPORTANT: If the `bds_cleanup_tenant` flag is set to **false** and the upgrade is attempted, you will no longer be able to access the tenants from the WebUI . Reach out to HPE support if you are in this situation and want to delete the invisible tenant.

- All pre-5.5.0 Kubernetes clusters are preserved during the HPE Ezmeral Runtime Enterprise upgrade process. You will be able to expand (with a manual step), shrink and delete those pre-5.5.0 clusters. For details, see Post Upgrade Tasks on page 904. As the older Kubernetes distributions are no longer used, you will not be able to upgrade them. However, it is possible to migrate the Kubernetes cluster to the HPE-Kubernetes-distribution.

- After the successful upgrade to HPE Ezmeral Runtime Enterprise 5.5.1, all new Kubernetes hosts will be created by default with `Containerd`, and all new Kubernetes clusters will use the HPE-Kubernetes-distribution.

  NOTE: Contact HPE support for more information on migrating pre-5.5.0 cluster to the HPE-Kubernetes-distribution.

Kubernetes Bundles Upgrade

Starting HPE Ezmeral Runtime Enterprise 5.5.0, HPE decouples the upgrade of the HPE Ezmeral Runtime Enterprise platform from Kubernetes-related components.

With this feature, the user can upgrade the following Kubernetes related components, without performing the complete HPE Ezmeral Runtime Enterprise platform upgrade. For more details, see Upgrading Kubernetes Bundles on page 903

Upgrading HPE Ezmeral Data Fabric on Kubernetes

This procedure describes upgrading HPE Ezmeral Data Fabric on Kubernetes clusters as part of upgrading HPE Ezmeral Runtime Enterprise. This task does not apply to Embedded Data Fabric implementations.

Prerequisites

NOTE:

This task is applicable only when upgrading to HPE Ezmeral Runtime Enterprise 5.4.1 or later only.

If you are upgrading to HPE Ezmeral Runtime Enterprise 5.4.0, contact your Hewlett Packard Enterprise support for upgrade assistance.

- **Required access rights:** Platform Administrator

About this task

If your environment implements Embedded Data Fabric, there are no specific upgrade steps to complete. Skip this task and proceed to Upgrading Kubernetes Add-Ons on page 900.

Procedure

On the Kubernetes master node, perform the following steps:

1. **IMPORTANT:** If you are using Kubernetes 1.19.x or later, skip this step and proceed to the next step.

   If you are using Kubernetes 1.18.x, execute these commands to download and install `kubectl` component for Kubernetes 1.19.x. You must only download the `kubectl`, and make sure you do not upgrade to Kubernetes 1.19.x.

   ```
   cd /tmp
   curl -LO https://dl.k8s.io/release/v1.19.0/bin/linux/amd64/kubectl
   mv /usr/bin/kubectl /usr/bin/kubectl.orig
   chmod 777 kubectl
   mv kubectl /usr/bin/kubectl
   ```

   If you are using Kubernetes 1.17.x or older, contact Hewlett Packard Enterprise support.

2. Upgrade HPE Ezmeral Data Fabric on Kubernetes as follows:

a) If you are using Centos 7, install `python36` using following command:

```
yum install -y python36
```

NOTE: You must remove the `python36` at the end of this procedure.

b) Make sure you get HPE Ezmeral Data Fabric on Kubernetes 1.5.1, using following commands:

```
git clone https://github.com/HPEEzmeral/df-on-k8s.git
cd df-on-k8s/bootstrap/p1.5.1
sed -i 's/BOOTSTRAP_PYTHON=.*$/BOOTSTRAP_PYTHON="python3"/'
bootstrap.sh
```

c) In the HPE Ezmeral Data Fabric on Kubernetes repository that you cloned in step **2 b.**, run the following command:

```
./bootstrap.sh upgrade --std_csimount
```

You can see the following prompts:

```
WARNING: Updating the CSI driver is a disruptive operation. All pods
using CSI will need to be restarted manually. If the objectstore pod is
running, it will also need to be restarted manually.yes
>>> Update Ezmeral Data Fabric CSI driver? (yes/no) [yes]: yes
/bin/sh: no: No such file or directory
/bin/sh: no: No such file or directory
```

The preceding step is optional. If you update the Data Fabric CSI driver, then anything that uses CSI, such as notebooks, must be restarted. However, it is good to upgrade CSI drivers, as you are upgrading the entire system, and most of the pods will be restarted anyway.

```
Use Airgapped Docker Registry? Note: All bootstrap containers must
exist in airgap registry! (yes/no) [no]: no
```

`Use Airgapped Docker Registry?`: If you use an Air-gapped Docker Registry, you will be prompted for more information.

For a reliable upgrade experience, it is recommended that you must answer yes and take Data Fabric offline during the upgrade process. If the Data Fabric is not offline, upgrade process can get disrupted.

```
Do you want to also take the Data Fabric named <Data Fabric> offline
and have it upgraded at this time? (yes/no) [no]: yes
```

NOTE: In certain steps, such as upgrading Kubernetes versions, most of the pods get restarted, including Data Fabric pods like `cldb` and `zk`. You must maintain quorum when upgrading `zk` pods, and this process can take a long time.

For a reliable upgrade experience, HPE recommends that you answer **no** and keep Data Fabric offline during the upgrade process. If the Data Fabric is not offline, upgrade process can get disrupted.

```
Would you like the Data Fabric to restart automatically after it is
upgraded? Keep it offline if you are going to perform any other major
upgrades next such as upgrading Kubernetes. (yes/no) [yes]: no
```

```
We are now ready to upgrade your Kubernetes components...yes
>>> Continue with upgrade? (yes/no) [yes]: yes
```

Ensure you enter **yes**.

You can use **no** if you want to stop the upgrade process now, and continue at another time. For example, you may not want to bring down your Data Fabric at this time.

d) If you are upgrading Data Fabric as part of Hewlett Packard Enterprise upgrade, then skip to Upgrading to HPE Ezmeral Runtime Enterprise 5.6.x on page 885. If you are upgrading HPE Ezmeral Data Fabric on Kubernetes, proceed to next step.

e) The bootstrap process takes some time for upgrading the components and creating the respective containers.

For example:

```
mcs-0    0/1     ContainerCreating   0          10s
```

When most pods finish creating their containers, the same example looks like:

```
mcs-0    0/1     Running   0          17m
```

NOTE: In the preceding example, Most pods are in `Running` state, but they are not in `Ready` state, and therefore the Data Fabric is offline. One exception is the `objectstore` pod, which is in `Init` state.

If the `objectstore` pod is in `Terminating` stage as shown in the following example, it will not create a new container:

```
objectstore-zone1-0 0/1 Terminating 5 10h
```

Delete the pod using following command:

```
kubectl delete pod objectstore-zone1-0 -n <Data Fabric
namespace> --force
```

Ensure that pod is in `Init` state, and not in `Running` sate, for example:

```
objectstore-zone1-0 0/1 Init:0/1 0 5s
```

f) If you are using Centos 7, remove the `python36` using the following command:

```
yum remove -y python36
```

Upgrading the Platform Software

This procedure describes upgrading the platform software of HPE Ezmeral Runtime Enterprise. This procedure is one of the tasks that is part of upgrading to HPE Ezmeral Runtime Enterprise 5.6.

Prerequisites

- You have performed all the tasks described in Before Upgrading the Platform on page 889. You copy the upgrade bundle to the correct location as part of the pre-check process.
- **Required access rights:** Platform Administrator

  The Platform Administrator must have login access to the Controller host as either `root` or the user who performed the original installation, as appropriate.
- You have upgraded HPE Ezmeral Data Fabric on Kubernetes, if applicable.
- All hosts are powered on and accessible.

Upgrade Process Overview

- Each of the hosts in will be upgraded. If the upgrade fails on one or more hosts, then the entire upgrade process will be rolled back on all hosts.

- The upgrade process updates the Controller host near the beginning of the upgrade process. At this point, the management service will be momentarily interrupted in order to change over to the new version; thus, explicitly refreshing the web UI during this interval may result in a browser error. If this occurs, refresh the page after a brief wait in order to get the latest version of the page.
- If you have applied a custom-authentication-bundle to **userconfig.tgz** file as described in Modifying the Authentication Package, back up the **userconfig.tgz** file before proceeding with platform upgrade process.

For information about the impact of the upgrade operation on existing workloads, see Upgrading to HPE Ezmeral Runtime Enterprise 5.6.x on page 885.

After you complete this task, you perform other upgrade tasks, depending on what you have implemented in your environment.

Upgrade Procedure

1. Enter Lockdown mode via the web interface by opening the **Quick Access** menu and then selecting .

   The **Enter system lockdown** dialog appears.

2. Enter a descriptive reason for the lockdown in the **Enter Reason** field, and then enter Lockdown mode by clicking **Submit**.

3. In the **Available Upgrades** tab of the **Installation** screen, click the **Upgrade** button for the build that you want to upgrade to.

   Lockdown mode will be confirmed, upgrade compatibility with your current version will be confirmed, and installation will then either proceed or display an error message.

   The **Confirm Upgrade** popup appears.

4. Enter a brief note in the **Notes** field if you like. Any notes you enter will be saved and can be viewed later.

5. (Optional) Select the **Auto-Roll** check box. Enabling this option upgrades the Controller hosts first, one at a time. After all the controller hosts are upgraded successfully, the upgrade is committed, and then the remaining hosts in the deployment are upgraded in batches. After the upgrade is committed, you can continue using HPE Ezmeral Runtime Enterprise while the remaining hosts are being upgraded.

6. Click **Proceed** to continue the upgrade.

7. Monitor the status of the upgrade process:

   The **Upgrade Progress** section appears on the **Available Upgrades** tab to display the status of the upgrade progress. Status values are the following:

   - **Pending:** The upgrade has not yet started.
   - **Upgrading:** The upgrade is taking place. Additional details appear during this phase, as HPE Ezmeral Runtime Enterprise extracts the upgrade package, upgrades the Controller host and Worker hosts, and finishes the upgrade. The package version being applied also appears.
   - **Finalizing:** Post-upgrade cleanup is occurring.
   - **Complete:** The upgrade has completed successfully.
   - **Rolling Back:** Upgrade has encountered an error and is reverting HPE Ezmeral Runtime Enterprise back to the original version.
   - **Error:** The upgrade did not complete successfully.

8. Monitor the status of the upgrade of the nodes. You can check the status of the nodes at any time during the upgrade process.

   To show the status of the nodes, click the **Show Upgrade Details and Actions** link to display the **Upgrade Details and Actions** window.

   Per-node status values should be interpreted in the same way as the status values of the overall upgrade process. There is also an additional `Upgraded` status, which indicates that the individual node has upgraded successfully while the overall upgrade process continues on other nodes.

   - After a host has a status of `Upgraded`, click the **Commit Upgrade** button to commit the HPE Ezmeral Runtime Enterprise upgrade on the hosts that successfully upgraded.
   - If the upgrade fails before the Primary Controller host is updated, then the message `Upgrade Failed` will appear and you will see the status `error` displayed for the `primary` host. You will need to investigate the Primary Controller host to resolve any issues, and then try again.
   - If the upgrade fails on another host after the Primary Controller has been upgraded, then the message `Awaiting manual recovery` will appear and you will see the status `error` displayed for the failed hosts. You will need to investigate the affected hosts to resolve any issues, and then try again.

9. If errors occur during the upgrade, do the following:

   - Check the affected host and resolve issues, then retry installation on the host by clicking **Retry Host**.

     The message `Retrying` appears in the **Available Upgrades** tab.
   - After a host has a status of `Upgraded`, commit the HPE Ezmeral Runtime Enterprise upgrade on the host by clicking **Commit Upgrade**.

   To cancel the upgrade process on all hosts and revert to the previously-installed version of HPE Ezmeral Runtime Enterprise, Click **Cancel and Rollback Upgrade**.

   You can view the completed upgrade in the **Upgrade History** tab of the **Installation** screen. .

10. (Optional) If you had applied a custom-authentication-bundle to **userconfig.tgz** file before the platform upgrade as described in Modifying the Authentication Package, do the following steps:

    - Restore the backed up the **userconfig.tgz** file to `/opt/bluedata/catalog/postconfig/` after the upgrade.
    - In the HPE Ezmeral Runtime Enterprise web UI, access the **Platform Authentication** page as **Site Admin**, and click **Submit** (without changing any settings). This will update `/etc/sssd/sssd.conf` file in all EPIC Apps, with the SSSD configuration generated by the customized authentication bundle.

11. Do one of the following:

    - If your environment uses Kubernetes:

      NOTE:

      Before you can create or edit Kubernetes clusters, You must exit Lockdown Mode on page 916.

      - If you choose to install the Falco Kernel Modules, after the platform upgrade is completed, you can proceed to (Optional) Installing Falco Kernel Modules on Hosts on page 900

- If you choose not to install the Falco Kernel Modules, proceed to Verifying the Upgrade on page 903.

- If your environment uses EPIC for container management, proceed to Verifying the Upgrade on page 903.

(Optional) Installing Falco Kernel Modules on Hosts

After you upgrade the HPE Ezmeral Runtime Enterprise software, you can install the Falco Kernel Module (optional) on the operating system of each Kubernetes host.

Prerequisites

- You have upgraded the platform software.
- You have obtained the correct version of the Falco Kernel Module (see Falco Container Runtime Security on page 499)

About this task

This task is optional. After upgrading the platform, If you choose to install the Falco Kernel Module on Kubernetes hosts in an existing Kubernetes cluster, Hewlett Packard Enterprise recommends that you install the module on each Kubernetes host before you upgrade the Kubernetes add-ons.

This task is not required for hosts that are not running Kubernetes.

Procedure

1. Install the Falco Kernel Module on the operating system of each Kubernetes host.

   For instructions, see Falco Container Runtime Security on page 499.

2. Tag each host with the `falco: true` label.

Upgrading Kubernetes Add-Ons

Use this procedure to upgrade the Kubernetes add-ons and to install new required add-ons on existing Kubernetes clusters in HPE Ezmeral Runtime Enterprise.

Prerequisites

- **Required access rights:** Platform Administrator
- You have Upgraded the platform software.
- Exit Lockdown mode. See Exiting Lockdown Mode on page 917.
- You have upgraded HPE Ezmeral Data Fabric on Kubernetes, if applicable.

About this task

In this procedure, you execute a script on each Kubernetes cluster. The script lists the deployed add-ons compared to the Kubernetes manifest, upgrades deployed add-ons, and installs new required add-ons.

IMPORTANT:

If you are upgrading from HPE Ezmeral Runtime Enterprise 5.3.1, the Kubernetes add-on upgrade steps for GPU hosts are different. See the issues and workarounds in the Release Notes on page 11.

If your environment includes HPE Ezmeral Data Fabric on Kubernetes clusters, ensure that you have ugraded HPE Ezmeral Data Fabric on Kubernetes before you upgrade the Kubernetes add-ons in this procedure.

Procedure

1. On the primary HPE Ezmeral Runtime Enterprise controller, change to the directory: **/opt/hpe/kubernetes/tools**

2. List all deployed add-ons compared to the add-ons in the manifest by executing the script using the `-t` (test) and `--required-only` parameters.

   In the following example, the HTTPS is enabled on the controller at IP address 192.0.2.5. The `-x` parameter specifies that HTTPS is used. The command specifies cluster name `my-k8s-cluster`. The `-f` parameter specifies the path to a JSON file that contains the username and password to use to connect to the controller host, and directs the script to use the Site Administartor tenant when updating the Kubernetes manifest file. For more information about the credentials file, see Kubernetes Add-On Upgrade Script on page 908.

   ```
   python k8s_addon_upgrade.py -c 192.0.2.5 -f /tmp/cred.json -x -k
   my-k8s-cluster --required-only -t
   ```

   The output of the command lists all of the manifest add-ons (both required and optional), and their versions, and all the add-on versions that are currently deployed on the cluster.

   By executing the script with the `-t` (or `--dry-run`) parameter, you can compare the deployed and manifest versions of optional add-ons, such as Istio, to determine whether or not you want to upgrade that add-on. If you have an applications that requires an earlier versions of an add-on, for example, you might not want to include that add-on when you execute the script to perform the upgrade.

   In the following example output, the `argocd` add-on is new to the `my-k8s-cluster` cluster, so there is no value displayed for the deployed version.

   ```
   Cluster my-k8s-cluster required add-ons info:
     argocd:
       deployed version: , tools:
       manifest version: 1.8.4-1, tools: 0.4
   ...
     hpecp-agent:
       deployed version: 1.1.2-1, tools: 0.1
       manifest version: 1.1.5-4, tools: 0.4
   ...
   dry run: upgrade add-ons ...<list-of-add-ons>... on cluster
   my-k8s-cluster
   Done
   ```

   When you execute the script without the `-t` (or `--dry-run`) parameter, the script installs the manifest versions of the add-ons you specify in the command. The `--required-only` parameter enables you to specify only the required add-ons without having to list each required add-on in the command individually.

3. (Upgrading from HPE Ezmeral Runtime Enterprise versions prior to 5.4.0 only.)

   (Optional) If you have enabled the Spark Operator, you will see the Spark Operator add-on:

   ```
   picasso-compute:
   deployed version: picasso-1.4.1-drop7-43-2, tools: 0.4
   manifest version: picasso-1.5.0-P150RC5-65-0, tools: 0.5
   ```

   Delete the Spark Operator add-on using the following command:

   ```
   python k8s_addon_upgrade.py -c 192.0.2.5 -f /tmp/cred.json -x -k
   my-k8s-cluster --cleanup-k8scluster-record
   ```

4. Peform the upgrade by executing the script with the `-t` parameter omitted.
   For example:

   ```
   python k8s_addon_upgrade.py -c 192.0.2.5 -f /tmp/cred.json -x -k
   my-k8s-cluster --required-only
   ```

   The add-on upgrade process can take more than 30 minutes to complete.

5. Verify that the add-ons are updated.

   Execute the script again with the `-t` and `--required-only` parameters. The output should show that the deployed versions and manifest versions are the same.

   If any add-on upgrades or installations have failed, you can run the script with the `--refresh` option to retry the operation. For information, see Kubernetes Add-On Upgrade Script on page 908.

6. (Optional) Upgrade optional add-ons, such as Istio.

   After you upgrade the required add-ons on each Kubernetes cluster, you can use the same script to upgrade optional add-ons on each cluster.

   The following example shows using the `-a` parameter to update only the Istio add-on.

   ```
   python k8s_addon_upgrade.py -c 192.0.2.5 -f /tmp/cred.json -x -k
   my-k8s-cluster -a istio
   ```

7. (Optional) If you have HPE Ezmeral Data Fabric on Kubernetes tenant add-on, in the Data Fabric cluster, you will see the information shown in the following example:

   ```
   picasso-tenant:
     deployed version: picasso-1.4.1-drop7-43-2, tools: 0.4
     manifest version: picasso-1.5.0-P150RC8-68-0, tools: 0.5
   ```

8. Proceed to Upgrading Kubernetes to a Later Version on page 902.

Related reference

Kubernetes Add-On Upgrade Script on page 908
Description of the Kubernetes add-on upgrade script with syntax and script options.

Upgrading Kubernetes to a Later Version

HPE Ezmeral Runtime Enterprise supportes multiple versions of Kubernetes. Depending on the applications you have installed or plan to install, you might want to upgrade Kubernetes to a later version than the version you installed before you upgraded the controller software.

For information about the versions of Kubernetes supported by this version of HPE Ezmeral Runtime Enterprise, see Support Matrixes on page 54.

For information about upgrading Kubernetes, see Upgrading Kubernetes on page 487.

Upgrading Kubernetes Bundles

Use this procedure to upgrade the Kubernetes to latest versions, without performing the complete HPE Ezmeral Runtime Enterprise upgrade. However, An upgrade of Kubernetes that requires changes to platform can be done only through HPE Ezmeral Runtime Enterprise upgrade.

Prerequisites

- **Required access rights:** Platform Administrator

About this task

Starting from HPE Ezmeral Runtime Enterprise 5.5.0, Kubernetes bundles can be upgraded, without performing the platform upgrade.

Procedure

1. If you have non-airgapped environment, ignore this step and skip to the next step. If you have an air-gapped environment:
   a. Download the latest Kubernetes add-on images and the latest Bootstrap images.
   b. Copy all the downloaded files to the appropriate air-gap repository locations.
2. Download the Kubernetes bundle, for example, `hpe-kubernetes-<version>.bin` , and place it in `/srv/bluedata/bundles` directory on the controller node of HPE Ezmeral Runtime Enterprise.
3. On **Settings Update**, under the **Available Kubernetes Bundle Updates** section, select the Kubernetes bundle, and click the respective **Update** button.

   The **Confirm Kubernetes Bundle Update** appears. Check the details and click **Update** button.

   NOTE: After successful update, you can click **View Kubernetes Bundle Update History** to see the update history.
4. If you want to perform the standard Kubernetes Upgrade on existing Kubernetes cluster, see Upgrading Kubernetes to a Later Version on page 902.
5. If you want to upgrade the Kubernetes add-ons, see Upgrading Kubernetes Add-Ons on page 900.

Verifying the Upgrade

Use this procedure to verify the upgrade done on existing Kubernetes clusters in HPE Ezmeral Runtime Enterprise.

You can verify that certain new features have been installed as follows:

- Verify that the left navigation screen of the HPE Ezmeral Runtime Enterprise UI includes **Policy Management**.

You can also perform your standard system status monitoring checks.

For example:

- Access the Kubernetes Administrator **Dashboard** screen. The Kubernetes Administrator **Dashboard** screen presents a high-level overview of current Kubernetes activity.
  - View the **Services** tab to verify that there are no errors or warnings.
  - View the dashboard tables and graphs to verify that they show valid data.

Proceed to Post Upgrade Tasks on page 904.

Post Upgrade Tasks

This article describes the post upgrade tasks of HPE Ezmeral Runtime Enterprise

Configure and Add GPU Hosts

Configure the hosts that contain GPUs and add them to configuration:

1. (Optional) Install Falco Kernel Modules.

   If you did not install Falco Kernel Modules on the GPU hosts you removed from HPE Ezmeral Runtime Enterprise before the upgrade, you can install them before you add the host to the configuration. See (Optional) Installing Falco Kernel Modules on Hosts on page 900.

2. Update the NVIDIA drivers and, if applicable, configure MIG. Then add the hosts to the HPE Ezmeral Runtime Enterprise and to the Kubernetes cluster.

   In Deploying MIG Support on page 840, Follow the instructions for hosts that have not yet been added to HPE Ezmeral Runtime Enterprise.

Update Kubernetes Dashboard

After upgrading HPE Ezmeral Runtime Enterprise from version 5.4.0 to 5.4.1, you must update the Kubernetes dashboard.

Follow the steps described in EZESC-1370 on Issues and Workarounds on page 15.

Upgrading Data Fabric Tenants

This procedure describes upgrading the Data Fabric Tenant services, without full tenant recreation through the UI. This procedure might be useful when upgrading from previous versions of HPE Ezmeral Runtime Enterprise.

Prerequisites

You have upgraded the Tenant Operator.

About this task

After upgrading the Tenant Operator, you might upgrade the Data Fabric Tenant services without full tenant recreation through UI.

Procedure

1. Remove owner reference from tenant namespace, using the command:

   ```
   kubectl edit ns [tenant-namespace]
   ```

   See the Owner Reference highlighted in the following example, remove the owner references.

```yaml
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: "2021-06-24T16:45:36Z"
  labels:
    hpe.com/cluster: hcp.mapr.cluster
    hpe.com/namespacetype: Tenant
    hpe.com/tenant: embedded
    hpe.com/version: 6.2.0
    hpecp.hpe.com/hpecptenant: hpecp-tenant-4
    istio-injection: disabled
  name: embedded
  ownerReferences:
  - apiVersion: hcp.hpe.com/v1
    blockOwnerDeletion: true
    controller: true
    kind: Tenant
    name: embedded
    uid: 636adec8-61b8-48c1-8356-be3cc625fa5b
  resourceVersion: "21029"
  uid: c32653ec-0e2d-44d5-9eca-88b5a7dd5adc
spec:
  finalizers:
  - kubernetes
status:
  phase: Active
~
```

2. Save the tenant as a YAML file, using the command:

```
kubectl get tenant [tenant-name] -o yaml > my_tenant.yaml
```

3. Delete the old tenant, using the command:

```
kubectl delete tenant [tenant-name]
```

4. Edit the YAML file by executing the following steps:
   a. Remove all managed fields, e.g: all from `metadata`, `name`, `status`, and so on.
   b. Change tag in `spec.baseimagetag` to the newer one.
   c. If you have individual image tags for services, ensure to update them. Also, if the value of an image tag matches `baseimagetag`, remove that tag.

      Refer to the following example. In the example, check the color of the rectangle and take the appropriate action:

      - Red: Remove the Items marked in red.
      - Yellow: Individual image tags.
      - Green: Update the information in baseimagetag tag.

```yaml
apiVersion: hcp.hpe.com/v1
kind: Tenant
metadata:
  creationTimestamp: "2021-06-24T16:45:51Z"
  generation: 1
  name: embedded
  resourceVersion: "21169"
  uid: 636adec8-61b8-48c1-8356-be3cc625fa5b
spec:
  baseimagetag: 202009090453C
  clustername: hcp.mapr.cluster
  clustertype: external
  corelocation: /var/lib/docker/mapr/cores
  grouplist:
  - hpecp-tenant-4-member
  - hpecp-tenant-4-admin
  imageregistry: gcr.io/mapr-252711
  loglocation: /var/lib/docker/mapr/logs
  podinfolocation: /var/lib/docker/mapr/podinfo
  security:
    environmenttype: hcp
    externalclientsecret: mapr-client-secrets
    externalconfigmap: mapr-external-cm
    externalhivesiteconfigmap: mapr-hivesite-cm
    externalserversecret: mapr-server-secrets
    externalusersecret: mapr-user-secrets
    usesssd: true
  tenantservices:
    sparkhs:
      count: 1
      image: spark-hs-2.4.4:202009090453C
    tenantcli:
      count: 1
      image: tenantcli-6.1.0:202009090453C
status:
  hivemetastate: {}
  livystate: {}
  overallstate:
    currentstate: READY
  sparkhsstate: {}
  sparkstate: {}
  sparktsstate: {}
  tenantclistate: {}
~
```

5. Create a tenant by applying the modified YAML file, using the command:

```
kubectl apply -f my_tenant.yaml
```

Updating Existing Tenant KubeDirector Applications

This procedure describes updating KubeDirector applications on existing tenants. This procedure might be useful when upgrading from previous versions of HPE Ezmeral Runtime Enterprise.

Prerequisites

- **Required access rights:** Host user/install user

About this task

After upgrading HPE Ezmeral Runtime Enterprise, existing tenants do not upgrade their KubeDirector application images automatically.

For example, consider the case in which an application was at version X for one release of HPE Ezmeral Runtime Enterprise, and is at version Y for the new release. After you upgrade HPE Ezmeral Runtime Enterprise, a tenant that was created **before** the upgrade continues to use application version X, even though a new version of the application exists. However, tenants you create **after** the upgrade use application version Y.

You can update the application version used by exisitng tenants manually. To obtain the information you need to update the existing tenant, you must create a new tenant. You can later delete the new tenant you created for this purpose.

On the Kubernetes master node, follow this procedure for each existing tenant.

Procedure

1. To view a list of available KubeDirector applications on a tenant namespace:

   ```
   kubectl get kdapp -n <tenant-namespace>
   ```

2. To view information about a specific KubeDirector application in the tenant namespace:

   ```
   kubectl describe kdapp <app-name> -n <tenant-namespace>
   ```

3. Delete all existing KubeDirector clusters that reference the KubeDirector application you are updating.

   For example, if there are existing instances of Jupyter Notebook created on a tenant, you cannot update the Jupyter Notebook application. You must delete all KubeDirector clusters referencing the Jupyter Notebook instances before you can modify the Jupyter Notebook application.

4. For some releases of HPE Ezmeral Runtime Enterprise, there might be changes in addition to image version. To get the latest version of the YAML file, proceed as follows:

   a. Create a new tenant.

   b. Display the information for the new tenant:

      ```
      kubectl describe kdapp <app-name> -n <tenant-namespace>
      ```

   c. Copy this information to a YAML file.

5. When you have the latest YAML file, perform one of the following:

   - **Option 1:**

a. Delete the existing KubeDirector application:

```
kubectl delete kdapp <kdapp-name> -n <tenant-namespace>
```

For example:

```
kubectl delete kdapp jupyter-notebook-new -n my-name-space
```

b. Create a new KubeDirector application with the latest YAML file version:

```
kubectl apply -f <new-kdapp-yaml-file> -n <tenant-namespace>
```

- **Option 2:** Edit the KubeDirector application:

```
kubectl edit kdapp <app-name> -n <tenant-namespace>
```

Replace the entire KubeDirector application YAML file with the latest version.

6. Save your changes.

Kubernetes Add-On Upgrade Script

Description of the Kubernetes add-on upgrade script with syntax and script options.

Starting HPE Ezmeral Runtime Enterprise 5.5.0, HPE decouples the upgrade of the HPE Ezmeral Runtime Enterprise platform from Kubernetes-related components.

With this feature, the user can upgrade the following Kubernetes related components, without performing the complete HPE Ezmeral Runtime Enterprise platform upgrade. For more details, see Upgrading Kubernetes Bundles on page 903.

```
python k8s_addon_upgrade.py -c <controller> -f <credentials_file> -k
<cluster-name>
{ --required-only | -a <add-on-list> } [ -t | --dry-run ]
```

```
python k8s_addon_upgrade.py -c <controller> -f <credentials_file> -k
<cluster-name> { -r | --refresh }
```

Description

The Kubernetes add-on upgrade script, `k8s_addon_upgrade.py`, lists the installed and manifest versions of the specified add-ons, and upgrades or installs the specified add-ons on the specified Kubernetes cluster.

- **Required privileges:** Platform Administrator
- This script is intended to be executed after upgrading the HPE Ezmeral Runtime Enterprise software and **after** upgrading HPE Ezmeral Data Fabric on Kubernetes clusters.
- Execute the script from the following directory of the primary HPE Ezmeral Runtime Enterprise controller:

  `/opt/hpe/kubernetes/tools`
- This script must be run on each Kubernetes cluster.

Parameters

`-h, --help`	Shows the help and exits.
`--required-only`	Specifies that only the required add-ons are to be upgraded or installed.
`-a ADDONS, --addons=ADDONS`	ADDONS is a comma-separated list of add-ons to upgrade. On Data Fabric clusters, this script does not support upgrading the `datafabric`, `picasso-tenant`, or `picasso-compute` add-ons. Those add-ons are upgraded as part of procedure to upgrade HPE Ezmeral Data Fabric on Kubernetes. When you use the `-a` or `--addons` parameter, only the add-ons you specify in the command are upgraded. If you have an add-on that is deployed on the cluster and you do not include it in the list, that add-on is not upgraded. To display a list of the required add-ons (currently deployed and that need to be deployed) and the current optional add-ons, run the command using the `-t` or `--dry-run` parameter.
`-c CONTROLLER, --controller=CONTROLLER`	CONTROLLER is the HPE Ezmeral Runtime Enterprise controller, specified as one of the following: • The IP address of the controller host • The cluster IP address (if platform HA is enabled) • The IP address of a Gateway host
`-f CREDFILE, --file=CREDFILE`	CREDFILE specifies the path to a JSON file that specifies the site admin tenant, and that contains the username and password to use to connect to the controller host. This JSON file stores the username under the key `user` and the password under the key `password`. For example: `{ "user": "myadminuser", "password": "admin789", "tenant": "/api/v1/tenant/1" }` If you do not specify the site admin tenant as shown in the preceding example, users that have Platform Administrator privilges but are not the default Platform Administrator (Site Admin) might receive a 404 error when the add-on upgrade script attempts to access the Kubernetes manifest.
`-k K8SCLUSTER, --k8scluster=K8SCLUSTER`	K8SCLUSTER specifies the name of the Kubernetes cluster.
`-l LOGFILE, --logfile=LOGFILE`	LOGFILE specifies the path and name of the log file. Default: `./k8s_addon_upgrade.log`
`-t, --dry-run`	Prints the operations without applying them.

	The output of the command lists the deployed add-ons, the version that is deployed on the cluster, and the version in the manifest. It also lists all the add-ons that can be upgraded or deployed. By executing the script with the `-t` or `--dry-run` parameter, you can compare the deployed and manifest versions of optional add-ons, such as `istio`, to determine whether or not you want to upgrade that add-on. If you have applications that require earlier versions of an add-on, for example, you might not want to include that add-on when you execute the script to perform the upgrade.
`-x, --https`	Connect to the controller using HTTPS.
`-r, --refresh`	Specifies that failed add-on upgrade or installation attempts be retried. On success, the operation clears any warnings and puts the cluster into a ready state. When this parameter is specified, the `-a` and `--required-only` parameters are ignored.

Usage

The recommended procedure is the following:

1. Execute the script with the `-t` parmeter to display the list of deployed and available add-ons. For the optional add-ons, compare the deployed and manifest version and determine whether you want to upgrade the add-on.
2. Execute the script with the `--required-only` parameter to upgrade and deploy the required add-ons.
3. Optionally, upgrade or deploy optional add-ons by executing the script with the `-a` parameter and listing the optional add-ons you want to upgrade or deploy.

The script upgrades only the add-ons that you specify in the command, either all required add-ons when you specify `--required-only`, or only the add-ons that you specify with the `-a` parameter.

The script can be run multiple times. Add-ons that have matching manifest and deployed versions are not affected.

In the following example:

- HTTPS is enabled on the controller at IP address 192.0.2.5
- The `-x` parameter specifies to use HTTPS for the connection.
- The cluster name is `my-k8s-cluster`
- The `--required-only` parameter is used instead of the `-a` parameter.
- The `-t` (test) parameter is specified, which means that both required and optional add-ons will be listed but not upgraded.
- A partial output of the command is shown. In the example, the `argocd` add-on is new to the `my-k8s-cluster` cluster, so there is no value displayed for the deployed version.

```
python /opt/hpe/kubernetes/tools
python k8s_addon_upgrade.py -c 192.0.2.5 -f /tmp/cred.json -x -k
```

```
my-k8s-cluster --required-only -t

Cluster my-k8s-cluster required add-ons info:
  argocd:
    deployed version: , tools:
    manifest version: 1.8.4-1, tools: 0.4
...
  hpecp-agent:
    deployed version: 1.1.2-1, tools: 0.1
    manifest version: 1.1.5-4, tools: 0.4
...
Cluster my-k8s-cluster deployed additional add-ons info:
...
Done
```

Upgrading from HPE Ezmeral Runtime Enterprise Essentials

Upgrade from HPE Ezmeral Runtime Enterprise Essentials to the full-featured HPE Ezmeral Runtime Enterprise or to HPE Ezmeral ML Ops by uploading a license. No additional steps are required.

Prerequisites

- You have purchased an upgraded license and obtained the license file.

  To purchase a license, contact Hewlett Packard Enterprise.

- **Required access rights:** Platform Administrator

About this task

You can upgrade from HPE Ezmeral Runtime Enterprise Essentials to one of the following license types:

- The full-featured HPE Ezmeral Runtime Enterprise.
- HPE Ezmeral ML Ops, which includes HPE Ezmeral Runtime Enterprise.

  You must upgrade to HPE Ezmeral Runtime Enterprise before you can upgrade to HPE Ezmeral ML Ops.

CAUTION: Downgrading from other licenses to HPE Ezmeral Runtime Enterprise Essentials is not supported.

You upgrade HPE Ezmeral Runtime Enterprise Essentials by uploading an HPE Ezmeral Runtime Enterprise license.

Procedure

1. On the **System Settings** screen, select the **License** tab.
2. Click the **Upload license** button to navigate to and select your new license file.

Results

HPE Ezmeral Runtime Enterprise deploys the appropriate add-on options and functions into your existing environment. When the actions complete, the GUI displays the additional management interfaces, endpoints, and so forth. For examples, see Navigating the GUI on page 143.

After you upload an HPE Ezmeral Runtime Enterprise license, the HPE Ezmeral Runtime Enterprise Essentials licenses is no longer valid, but it is not deleted.

Related concepts

Licensing on page 734

Related tasks

Adding Licenses on page 736

Manually Restarting HPE Ezmeral Runtime Enterprise Services

This topic describes restarting HPE Ezmeral Runtime Enterprise services in non-Kubernetes hosts.

NOTE: This article does not apply to Kubernetes hosts.

To manually restart HPE Ezmeral Runtime Enterprise services:

1. Log in to the web interface as a Platform Administrator, as described in Launching and Signing In on page 136.
2. Determine which host is the Controller host. If Platform High Availability protection is enabled (see High Availability), then also determine which hosts are the Shadow Controller and Arbiter.
3. Log in to the Controller host as either `root` or the ID that was used to install HPE Ezmeral Runtime Enterprise.
4. If Platform High Availability protection is enabled, then execute the following command to suspend this protection:

   ```
   /opt/bluedata/bundles/<epic install bin folder>/startscript.sh --action
   suspendha
   ```

5. Execute the following commands on every host except Gateway hosts to shut down services:

   ```
   systemctl stop bds-monitoring
   systemctl stop bds-worker
   systemctl stop bds-controller
   ```

   NOTE: The name `controller` in the service name `bds-controller` may be misleading, as it runs on both the Controller and Worker hosts.

6. On every host (including the Controller), bring the services up in the following order:

   ```
   systemctl start bds-cgroup
   ```

   This step is only needed after a system reboot.

   ```
   systemctl start bds-worker
   systemctl start bds-controller
   systemctl start bds-monitoring
   ```

7. Within each host, verify that the following services are up and running:

   ```
   systemctl status bds-worker
   systemctl status bds-controller
   systemctl status bds-monitoring
   ```


last-updated: May 06, 2024

8. If applicable, re-enable platform High Availability protection by executing the following command on the Controller host:

```
/opt/bluedata/bundles/<epic install bin folder>/startscript.sh --action
resumeha
```

Uninstalling and Reinstalling HPE Ezmeral Runtime Enterprise

There are many reasons why you may need to uninstall HPE Ezmeral Runtime Enterprise from the Controller host and any installed Worker hosts and then start over, such as:

- HPE Ezmeral Runtime Enterprise installed as `root` when you meant to install as a non-root user. In this case, a subsequent non-root installation will probably fail if the hosts have not been refreshed. If this happens, contact Hewlett Packard Enterprise for support.
- Unrecoverable error.
- Configuration changes to the host or infrastructure.
- Moving from a test environment to a production environment.

There are two basic way to uninstall and reinstall HPE Ezmeral Runtime Enterprise:

- Completely refresh the Controller host and any Worker hosts to a "bare metal" state, reinstall the operating system, and then reinstall HPE Ezmeral Runtime Enterprise. This is the preferred method, because installation makes numerous configuration changes to the hosts in the deployment that are not completely reversible and that may impact the reinstallation process. Completely refreshing the hosts is beyond the scope of this documentation. Once the hosts are refreshed, you may begin the installation process again, as described in Installation Overview.
- Run the HPE Ezmeral Runtime Enterprise uninstaller on the Controller host and, if needed, on any Worker hosts. You may need to use this option if completely refreshing the hosts cannot be accomplished easily. This article describes this method.

Backing up the Configuration

If you plan to rebuild the deployment on another host and want to carry over settings from the deleted HPE Ezmeral Runtime Enterprise deployment, then back up the following:

1. Collect a Level 2 support bundle. See Support Bundles Tab.
2. Take screenshots of all platform and tenant/project settings. (The support bundle already captures these settings, but having screenshots will help you apply similar settings when redeploying HPE Ezmeral Runtime Enterprise.)
3. Back up any customization changes, which includes but is not limited to:
   - **Authentication Package:**`/opt/bluedata/catalog/postconfig/userconfig.tgz`
   - **Monitor changes:**`/etc/curator.actions.yaml` (inside the `monitor` container)
   - **Custom feeds:** Execute the `change_feed` command on the new HPE Ezmeral Runtime Enterprise deployment.

Running the Uninstaller

To run the uninstaller:

1. Back up all data.

2. Remove all FS mounts. See The FS Mounts Screen.

3. Log into the host that you will be using as the Controller host using either the root account and password or your assigned username and password.

4. On the Controller host, execute the following command:

```
/opt/bluedata/bundles/<hpecp_install_folder>/
startscript.sh --erase --force
```

5. Monitor the erase process and address any reported problem. The log file is located at `/tmp/worker_setup_<timestamp>`.

   If HPE Ezmeral Runtime Enterprise was not installed using the agent, then the Controller will delete the Worker and Gateway hosts remotely. In the unlikely event that remote deletion does not succeed, you can manually uninstall HPE Ezmeral Runtime Enterprise on each host by executing the following commands for a non-agent-based installation by proceeding to Step 8. If the procedure does work, then skip to Step 9.

   NOTE: `--erase --force` command uninstalls the HPE Ezmeral Runtime Enterprise installation. However, Python and Docker packages, that were installed during HPE Ezmeral Runtime Enterprise installation, will not be uninstalled.

6. For an agent-based installation, log in to the host and then execute the following commands:

   - **Worker:** `/opt/bluedata/bundles/<hpecp_install_folder>/<common-hpecp.bin> -ef --onworker --node-type worker --worker <worker-ip>`

   - **Gateway:** `/opt/bluedata/bundles/<hpecp_install_folder>/<common-hpecp.bin> -ef --onworker --node-type proxy --gateway-node-ip <worker-ip>`

7. Reboot all of the hosts in the platform.

8. Execute the following commands to verify that HPE Ezmeral Runtime Enterprise has been successfully deleted:

   - `bdconfig -sysinfo`. The system should return the message command not found.

   - `rpm -qa | grep hpe-cp`. The system should return an empty response.

9. Proceed as follows:

   - If this host will not be reused as a Worker, then you have completed the uninstallation process.

   - If you plan to reuse this host as a Worker, then proceed to the next step.

10. Verify that the `VolBDSCStore` thin pool volume has been deleted. If not, then you will need to delete the volume before proceeding. The following example shows that `VolBDSCStore` still exists on the disk partition `/dev/sdc`:

```
lsblk

 NAME                              MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT

 sda                                 8:0    0 465.7G  0 disk
 ├─sda1                            8:1    0   500M  0 part /boot
 ├─sda2                            8:2    0 465.2G  0 part
   ├─rootvg-lv_root              253:0    0   200G  0 lvm  /
   ├─rootvg-lv_swap              253:1    0    54G  0 lvm  [SWAP]
   ├─rootvg-lv_var_log_bluedata  253:4    0   100G  0 lvm  /var/log/bluedata
 sdb                                 8:16   0   3.7T  0 disk
 ├─bluedatavg-lv_opt_bluedata    253:5    0   300G  0 lvm  /opt/bluedata
 ├─bluedatavg-lv_srv             253:6    0   300G  0 lvm  /srv
 ├─bluedatavg-lv_wb              253:7    0   500G  0 lvm  /wb
 sdc                                 8:32   0   3.7T  0 disk
 └─sdc1                            8:33   0   3.7T  0 part
   ├─VolBDSCStore-thinpool_tmeta 253:2    0  15.8G  0 lvm
   │ └─VolBDSCStore-thinpool     253:8    0    36T  0 lvm
   ├─VolBDSCStore-thinpool_tdata 253:3    0    36T  0 lvm
   │ └─VolBDSCStore-thinpool     253:8    0    36T  0 lvm
```

11. Delete the volume group `VolBDSCStore` by executing the following command:

```
sudo vgremove VolBDSCStore
```

12. Delete all physical volumes being used for the volume group `VolBDSCStore` by executing the following command:

```
sudo pvremove $(pvs | grep VolBDSCStore | awk '{print $1}')
```

13. If the above steps do not delete volume, then consider using a "brute force" method, such as `wipefs`, as follows:

```
sudo wipefs -a -f /dev/sdc
```

14. Ensure that `/var/lib/docker` is empty. If not, then delete everything below `/var/lib/docker`.

15. Verify that `/etc/sysconfig/docker-storage` has `DOCKER_STORAGE_OPTIONS` equal to nothing (e.g. `DOCKER_STORAGE_OPTIONS=`).

Support and Troubleshooting

This section contains the following articles:

- **Lockdown Mode:** How to enter and exit Lockdown mode in order to perform tasks that require the deployment to be in a quiescent state. See Lockdown Mode on page 916.
- **The Support/Troubleshooting Screen:** Describes the **Support/Troubleshooting** screen. See The Support/Troubleshooting Screen.
- **Support Bundles Tab:** Describes the **Support Bundles** tab of the **Support/Troubleshooting** screen. See Support Bundles Tab.

- **Config Checks Tab:** Describes the **Config Checks** tab of the **Support/Troubleshooting** screen. See Config Checks Tab.
- **Search Tab:** Describes the **Search** tab of the **Support/Troubleshooting** screen. See Search Tab.
- **Generating a Support Bundle:** Guides you through the process of generating a support bundle that can be sent to Hewlett Packard Enterprise to help Technical Support personnel diagnose and remediate issues with your deployment. See Generating a Support Bundle.
- **Troubleshooting Overview:** Links to articles with specific troubleshooting steps for a range of issues. See Troubleshooting Overview.

More information

Support and Other Resources on page 75

Lockdown Mode

Lockdown mode prevents all users who do not have Platform Administrator privileges from making any changes to the deployment, such as creating/editing ActionScripts, launching applications or the Kubernetes Web Terminal, or creating DataTaps. This mode ensures that the deployment will remain stable while the Platform Administrator makes configuration changes, such as (but not limited to):

- Adding additional hosts to the platform.
- Upgrading the HPE Ezmeral Runtime Enterprise version.

You may also use the Lockdown mode during other maintenance activities outside the scope of HPE Ezmeral Runtime Enterprise; however, Lockdown mode does not prevent users from logging into any virtual nodes that already exist and performing activities within the virtual nodes.

Do not enter Lockdown mode when creating or editing a Kubernetes cluster. Creating or editing a Kubernetes cluster while the site is in Lockdown mode can result in errors related to the cluster connections to services, or in service endpoints not being displayed for that Kubernetes cluster.

Entering Lockdown mode will happen immediately if no tasks are running. If one or more tasks are in progress when the Platform Administrator enters Lockdown mode, then HPE Ezmeral Runtime Enterprise will complete those tasks and prevent additional changes. For example:

- New hosts will finish installing.
- Apps will finish launching.

Entering Lockdown Mode

To enter Lockdown mode:

1. Open the **Quick Access** menu and then select **Enter system lockdown**.

   The **Enter system lockdown** dialog appears.

2. Enter a descriptive reason for the lockdown in the **Enter Reason** field and then click **Submit**.

   The red **Entering system lockdown indicator** appears in the **Toolbar** while HPE Ezmeral Runtime Enterprise finishes any jobs that are currently running. This indicator changes to **System lockdown**

once HPE Ezmeral Runtime Enterprise has completed all running jobs and finished entering Lockdown mode. Any user who attempts to make any changes will receive a popup warning that this mode is in effect. Clicking the red notification area opens a popup that lists the user who enabled Lockdown mode and the listed reason.

Exiting Lockdown Mode

To exit Lockdown mode, open the **Quick Access** menu and then click **Exit system lockdown**. Exiting from Lockdown mode is instantaneous and allows normal usage to resume, as long as no protected tasks are running. If so, then the tasks will complete before normal usage can resume.

More information

Support and Other Resources on page 75

Alerting

HPE Ezmeral Runtime Enterprise provides a basic framework that enterprises can leverage to create alerting mechanisms at both the platform and container/virtual node level.

Platform

Nagios is the event alerting framework on platform-monitored system services. Customers are expected to configure Nagios and/or write standard plugins/scripts using the following configuration:

- **URL:** `http://<controller_ip>:8085/nagios`
- **Username:** `nagiosadmin`
- **Password:** `nagiosadmin`

Containers/Virtual Nodes

For Docker containers (virtual nodes), the web interface displays the status of core services, such as **Auth** and **HPE Agent**, along with some custom services, such as Cloudera Manager or Ambari, that are registered as part of the specific application image; however, it does not provide any out-of-the-box alerting functionality on this level. Best practice is to use any alerting functionality that may be included in the application that is running in the container, such as Cloudera Manager or Ambari.

A RESTful API is also included that can obtain the status of services registered with and monitored by `vagent`. This data can be used to create custom alerting functionality using existing enterprise monitoring/alerting tools. The RESTful API for monitoring service status is:

```
https://github.com/bluedatainc/solutions/blob/master/APIs/
v1_bluedata_apis_doc.md#42-get-cluster-service-status
```

You may also install and configure a monitoring tool of your choice (e.g. Nagios) in the virtual nodes as part of application image and/or via ActionScripts. For example, see Setting up Nagios Email Alerts.

More information

Support and Other Resources on page 75

Setting up Nagios Email Alerts

Nagios is an open source system and network monitoring application that watches specified hosts and services and alerts you when a monitored service changes state from normal to having an issue and when that issue is resolved. Click here (link opens an external website in a new browser tab/window) for more information about Nagios.

NOTE: You may also configure Nagios email and/or SNMP alerts using the HPE Ezmeral Runtime Enterprise interface, as described in The Notification Settings Screen.

You may access the Nagios interface by navigating to `http://<controller_ip>:8085/nagios/`, where `<controller_ip>` is the IP address of the Controller host.

NOTE: The default password may be changed as described in Updating External Service Passwords.

The default login credentials are:

- **Username:** `nagiosadmin`
- **Password:** `nagiosadmin`

To configure Nagios to send email alerts:

1. SSH into the Controller host and then execute the command `docker ps` to list the containers running on the Controller host.

   This command returns a tabulated list of all of the containers that are running on the Controller host.

2. Search the `IMAGE` column of the tabulated list for the entry that includes the word `nagios`, and make note of the `CONTAINER ID` for that container.

3. Execute the command `docker exec -it <container_id> /bin/bash`, where `<container_id>` is the alphanumeric string listed in the `CONTAINER ID` column for the `nagios` container.

4. The `contacts.cfg` file contains the information needed to send Nagios alerts to groups and/or individual users. Edit this file by executing the command `vi /etc/nagios/objects/contacts.cfg`. In this file:

   - The `CONTACTS` section lists individual users who will receive email alerts. Within this section, each `define contact` block defines an individual user.
   - The `GROUPS` sections lists user groups who will receive email alerts.

5. Either:

   - Add a new `define contact` block to add a new user who will receive Nagios email alerts. See The Define Contact Block, below.
   - Modify an existing `define contact` block to change an existing user who currently received Nagios email alerts. See The Define Contact Block, below.

6. Execute the command `:x` to save your changes and exit the text editor.

7. Execute the command `supervisorctl restart nagios` to update Nagios with the changes you just made.

8. If needed, configure your email server to work with Nagios as described here (link opens an external website in a new browser tab/window).

last-updated: May 06, 2024

This procedure updates the Nagios email configuration, and the specified user(s) will start receiving email alerts.

You may also see the Other Resources, below, for additional help configuring Nagios email alerts.

The Define Contact Block

Each defined contact block in the `contacts.cfg` file appears as follows:

```
define contact {
          contact_name <username> ;
          use generic-contact ;
          alias <full name of user> ;
          email <email_address> ;
}
```

For example, if you have a user named Jane Doe, the defined contact block for her may look like this:

```
define contact {
          contact_name Jane ;
          use generic-contact ;
          alias Jane Doe ;
          email jane_doe@emailaddress.com ;
}
```

Other Resources

You may also view these external resources for additional information, if desired (links open external websites in anew browser tabs/windows):

- https://searchdatacenter.techtarget.com/tip/Nagios-notifications-Setting-up-alerts-in-the-network-monitoring-tool
- https://www.linux.com/learn/setting-email-alerts-network-monitoring-nagios
- https://library.nagios.com/library/products/nagios-core/manuals/

More information

Support and Other Resources on page 75

Platform Logs

Log Names and Locations

The following log names are always for the most recent log. Older logs have a date string appended in the same way as when rotating or archiving `/var/log/` messages.

Platform Logs in `/var/log/bluedata`

The following logs are stored on each host under `/var/log/bluedata`.

`bds-audit.log`	Captures API requests and responses to and from the Management service. By default, only state-changing requests are logged here; some routine operations, such as polling to repeatedly read the status of an object will not appear here. This log is only appended on the currently active Controller host.

`bds-cachingnode.log`	The Caching Node service provides an accelerated I/O channel between the storage services referenced by DataTaps and the applications running in the pods. On the Controller host, this service also provides the back end for DataTap browsing. This log captures all these transactions.
`bds-cgroup.log`	The `cgroup` service is a foundational dependency for the `bds-controller` and `bds-worker` services. This log can provide insights on startup issues with the `bds-controller` and `bds-worker` services, such as potential issues with the Linux user used to start the services.
`bds-controller.log`	The `bds-controller` service is the Management service. It runs on all the hosts. On the Primary and Shadow Controller, this service also serves the RESTful API. On Worker hosts, this service manages the hypervisor agent. Use this log only when the `bds-controller start` or `stop` commands report issues.
`bds-dataserver.log`	This service registers the containers of a host with the Caching Node service. On the Controller host, the Data Server also receives DataTap browsing queries from the Management service. This log captures all these registrations and query requests.
`bds-mgmt.log`	Tracks Management service activities. On the currently active Controller host, this log contains a superset of the information sent to `bds-audit.log`. The service logs internally initiated tasks, API requests and responses, and some details about the phases and progress of tasks.
`bds-worker.log`	The `bds-worker` service manages the `dataserver` and the caching node (`cnode`) services. This log file is the best starting point when the `dataserver` or `cnode` services (that comprise DataTaps) have issues. Use this log for troubleshooting when the `cnode` or `dataserver` services do not come up.

NOTE: The Erlang console logs will still show some messages from Erlang VM; these messages are rarely used.

Other platform logs are contained in subdirectories of `/var/log/bluedata` or in other directories.

`/var/log/bluedata/pl_ha/log.0`	Logs the platform HA setup information.
`/var/log/bluedata/install/`	The install directory contains log files related to installation or upgrade of the platform or Kubernetes nodes.
`/var/log/secure`	The platform does not log messages directly to this file. However, if a security or permission violation problem occurs, then this log is a good place to check.

Log Rotation Properties

Log rotation properties are configured in `/etc/logrotate.d/bds`. The default properties are:

- Weekly rotation

- Four (4) older logs retained.
- All but one of the older logs is in a gzip file.

Related reference

Data Fabric Core Logs on page 921
The fluentd component reads and parses the following Data Fabric Core log files on each node in the cluster.

More information

HPE Kubernetes Cluster Troubleshooting on page 935
Troubleshooting Kubernetes clusters that are running the Hewlett Packard Enterprise distribution of Kubernetes can involve examining the `.service` files, environment variables, and using journald to examine logs.

Support and Other Resources on page 75

Data Fabric Core Logs

The fluentd component reads and parses the following Data Fabric Core log files on each node in the cluster.

The fluentd component reads and parses the following Data Fabric Core log files on each node in the cluster.

Table

Service Name	Parsing Method	Description
Opentsdb	Multi-line	Logs from `/opt/mapr/cluster_logs/<clustername>/<cluster install timestamp>/opentsdb/opentsdb-0/opentsdb/opentsdb_daemon.log`
Apiserver	Multi-line	Logs from `/opt/mapr/cluster_logs/<clustername>/<cluster install timestamp>/apiserver/mcs-*/apiserver/apiserver.log`
Kibana	Multi-line	Logs from `/opt/mapr/cluster_logs/<clustername>/<cluster install timestamp>/kibana/kibana-*/kibana/kibana_daemon.log`
Collectd	Multi-line	Logs from: • `/opt/mapr/cluster_logs/<clustername>/<cluster install timestamp>/cldb/cldb-/collectd/collectd_daemon.log` • `/opt/mapr/cluster_logs/<clustername>/<cluster install timestamp>/cldb/cldb-/collectd/collectd.log`

Table (Continued)

Service Name	Parsing Method	Description
Hivemetastore	Multi-line	Logs from `/opt/mapr/cluster_logs/<clustername>/<cluster install timestamp>/hive/hive*/hive/mapr/mapr-metastore*.log`
Mapr_monitoring	Multi-line	Logs from `/opt/mapr/cluster_logs/<clustername>/<cluster install timestamp>/elasticsearch/elasticsearch*/elasticsearch/MaprMonitoring.log`

More information

Support and Other Resources on page 75

The Support/Troubleshooting Screen

Clicking the **Support** button in the **Quick Access** menu opens the **Support/Troubleshooting** screen. This screen has the following tabs:

- **Support Bundles:** Allows you to create, delete, and download support bundles to help you troubleshoot problems. See Support Bundles Tab.
- **Config Checks:** Allows you to perform basic configuration checks and view results. See Config Checks Tab.
- **Search:** Allows you to search logs by keyword(s) and/or date range. See Search Tab.

More information

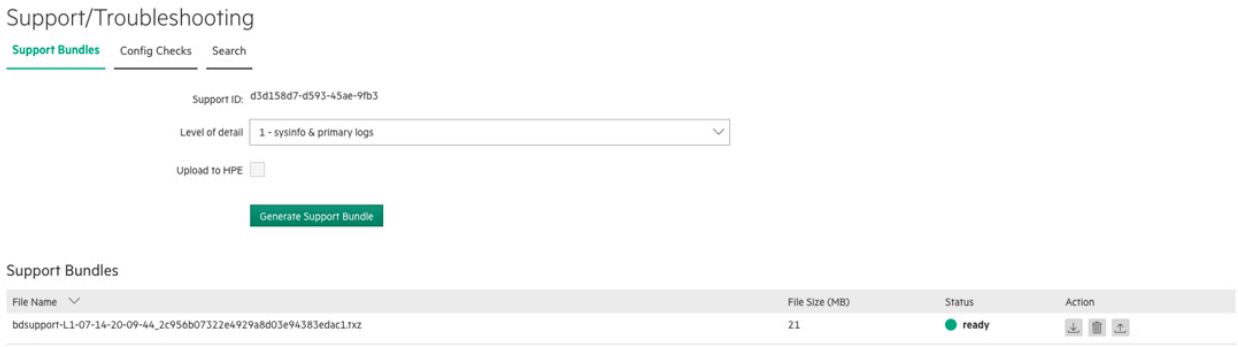
Support and Other Resources on page 75

Support Bundles Tab

Describes the **Support Bundles** tab of the **Support/Troubleshooting** screen.

NOTE: This feature is not available for HPE Ezmeral Runtime Enterprise deployments that are running SLES. Please see Collecting Support Bundles for instructions.

The **Support Bundles** tab of the **Support/Troubleshooting** screen (see The Support/Troubleshooting Screen) enables you to create, delete, and download support bundles to help you troubleshoot problems.



If you contact Hewlett Packard Enterprise Support, you might be requested to forward a support bundle for support purposes. You may either:

- Upload the support bundle directly from the **Support Bundles** tab.
- Download the support bundle to your computer and forward it to Hewlett Packard Enterprise by other means.

To upload a support bundle, your deployment must have both of the following:

- Internet access, either directly or through a proxy.
- DNS service that can resolve references to Internet addresses.

The **Support Bundles** table contains the following information for all currently available support bundles:

- **File Name**: Name of the support bundle file.
- **File Size (MB)**: Size of the file in megabytes.
- **Status**: Status of the support bundle. Status is one of the following:
  - **Generating**: The support bundle is being created.
  - **Ready**: The support bundle is ready for download or upload.
  - **Uploading**: The support bundle is being uploaded to Hewlett Packard Enterprise. The report can be downloaded, but not deleted.
  - **Uploaded**: The most recent attempt to upload the support bundle to Hewlett Packard Enterprise succeeded. You may download, re-upload, or delete this report.
  - **Upload failed**: The most recent attempt to upload the support bundle to Hewlett Packard Enterprise failed. You may download, re-upload, or delete this report.
  - **Error**: The support bundle is in an unknown state that is probably corrupted. You may delete this report.
- **Action**: You can perform the following actions for each available support bundle:
  - **Download**: Clicking the **Download** icon (envelope) opens an OS-default **Download** window that enables you to retrieve the selected support bundle.
  - **Delete**: Clicking the **Delete** icon (trash can) deletes the selected support bundle. A popup warning appears asking you to confirm or cancel the action. To proceed, click **OK**. To exit without deleting the support bundle, click **Cancel**.
  - **Upload**: Clicking **Upload** icon (up arrow) uploads the selected support bundle to Hewlett Packard Enterprise.

NOTE: It is preferable to raise a support ticket and attach the support bundle to that ticket.

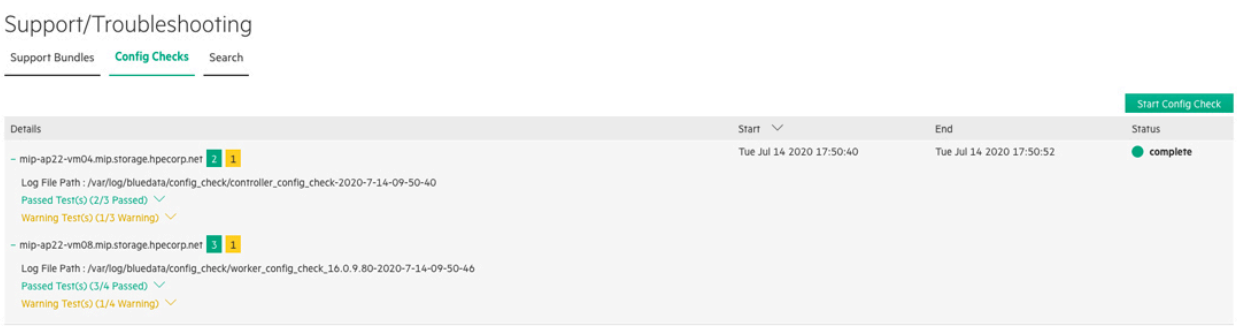
To generate a support bundle, see Generating a Support Bundle. If needed, you may also collect support bundle information from the command line, as described in Collecting Support Bundles.

More information

Support and Other Resources on page 75

Config Checks Tab

The **Config Checks** tab of the **Support/Troubleshooting** screen (see The Support/Troubleshooting Screen) allows you to perform a series of basic configuration checks on the host(s) and view the results of those checks.



To use these checks:

1. Click the **Help** button in the **Toolbar**, and then select **Support** in the pull-down menu.

   The **Support/Troubleshooting** screen appears with the **Support Bundles** tab selected.

2. Select the **Config Checks** tab.

3. Click the **Start Config Check** button.

4. HPE Ezmeral Runtime Enterprise will perform a series of configuration checks and report the results in the **Config Checks** tab.

In the results:

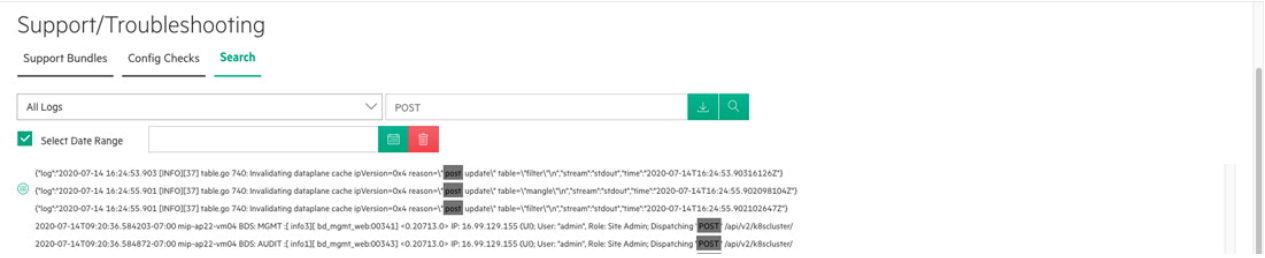
- A number in a green square indicates the number of successful checks performed on a host.
- A number in an orange square indicates the number of checks that ended with a warning status on a host.
- A number in a red square indicates the number of failed checks performed on a host.
- Clicking a hostname toggles expanding or collapsing the test results for the selected host.
- Clicking a down arrow next to **Passed Test(s)**, **Warning Test(s)**, or **Failed Test(s)** entry expands the details for that entry.
- Clicking an up arrow next to a **Passed Test(s)**, **Warning Test(s)**, or **Failed Test(s)** entry collapses the details for that entry.

More information

Support and Other Resources on page 75

Search Tab

The **Search** tab of the **Support/Troubleshooting** screen (see The Support/Troubleshooting Screen) allows you to search logs by keyword(s) and/or date range.



This screen has the following functions:

- **Logs:** Use the left pull-down menu to select the logs to include in the search. The available options are:
  - HPE Management Logs
  - CNODE Logs
  - All Logs (searches both of the above)
- **Query**: Enter an Elasticsearch query in the field on the right. Click here for information about Elasticsearch queries (link opens an external website in a new browser tab/window). You may also enter one or more keyword(s) separated by spaces or commas.
- **Download:** Generating a search and then clicking the **Download** button (arrow) downloads the search results in .csv format.
- **Search:** Selecting logs, a query, and/or a date range and then clicking the **Search** button (magnifying glass) performs the specified search.
- **Select Date Range:** Checking this check box displays the **Calendar** function.
- **Calendar:** Displays your selected date range, if the **Select Date Range** check box has been changed. The **Calendar** icon opens a popup that allows you to select your desired range by clicking the starting and ending dates.
- **Delete:** Clicking the **Delete** icon (trash can) removes your selected date range.
- **Results:** The results of your search appear in the bottom of the tab with matching query terms highlighted after you click the **Search** icon (magnifying glass).

To search the logs:

1. Use the **Logs** pull-down menu to select the log(s) to search.
2. If desired, enter either an Elasticsearch query or one or more keyword(s) separated by commas or spaces in the **Query** field.
3. If desired, check the **Select Date Range** check box and then proceed to Step 4. Otherwise, skip to Step 6.
4. Click the blue **Calendar** button to open a **Calendar** popup.
5. Navigate to the desired month, then click the desired starting date, and then click the desired ending date.
6. Click the **Search** icon (magnifying glass) to execute the search. Results will appear in the bottom of this tab with matching query terms highlighted.
7. If desired, click the **Download** button (arrow) to download the search results in .csv format.

More information

Support and Other Resources on page 75

Generating a Support Bundle

Use the **Support/Troubleshooting** screen to generate a support bundle.

To generate a support bundle, do the following:

1. Open the **Support/Troubleshooting** screen, and then select the **Support Bundles** tab (see Support Bundles Tab).
2. If you want to upload the support bundle to Hewlett Packard Enterprise, then check the **Upload to HPE** check box; otherwise, leave this check box cleared to prevent the report from being uploaded.
3. Select the desired level of detail using the **Support Bundle Level** pull-down menu. You may select a number from 1-3, where **1 - sysinfo & primary logs** is the least detailed and 3 is the most detailed. Selecting **3 - large files** may generate a report that is tens or even hundreds of megabytes in size, depending on your installation and the circumstances.
4. Click the blue **Generate Support Bundle** button.
5. You may download and/or re-upload the new support bundle once the **Status** changes from **Generating to Ready**.

NOTE: You may generate and/or upload a single support bundle at a time.

NOTE: It is preferable to create a support ticket at https://HPE.zendesk.com and attach the support bundle to that ticket.

For information about manually collecting logs to generate a support bundle, seeCollecting Support Bundles.

For information about the contents of support bundles, see Support Bundle Contents.

Troubleshooting

Symptom	Recommended Action
The support bundle is empty, or you are unable to collect a support bundle via the web interface.	Collect a support bundle using the command line, as described in Collecting Support Bundles.

More information

Support and Other Resources on page 75

Collecting Support Bundles

As an alternative to using the **Support/Troubleshooting** screen to generate a support bundle, you can manually collect logs to include in a support bundle by entering commands.

There are two ways to collect support bundles:

- Using the **Support/Troubleshooting** screen, as described in The Support/Troubleshooting Screen.
- Manually collect support logs from one or more hosts.

To manually collect support logs from the hosts, execute the following command on the RHEL or SLES hosts you want to collect HPE Ezmeral Runtime Enterprise logs as either:

- The user who installed HPE Ezmeral Runtime Enterprise

- `root`
- or using `sudo`

Executing the appropriate script generates aggregated `.txz` files and places them in `/tmp/bluedata/sosreport/customerlog`, which is the same directory used by the web interface when collecting Support Bundles, meaning that you can view and access script-generated reports using the **Support Bundles** tab of the **Support/Troubleshooting** screen (see Support Bundles Tab). The generated `.txz` file will have a name similar to `bdsupport-L1-06-30-20-08-01_66e23857a14f46e0802ea8bda2b1ddf8.txz`.

This file is an aggregation, meaning that it contains one or more embedded `.txz` files. You must therefore `untar` the generated .txz file and then `untar` the embedded `.txz` files to view the Support Bundle contents. To do this, execute the following command:

```
tar Jxvf <filename>.txz
```

This file is an aggregation if run from `bluedata-report-sc.py` and therefore contains 1-n other `supportconfig` packages. You must therefore untar twice:

- Once for the outer package
- Once for the inner packages.

The package contains a `.txz` file named something like `scc_hpecp_level_3_44631395593.txz`. Untarring this package reveals a number of text (`.txt`) files and the `hpecp` directory, which contains HPE Ezmeral Runtime Enterprise plugin output.

If you need to obscure your organization's IPv6 and MAC addresses, then you will need to manually edit the `.txt` files.

RHEL

Execute the following command:

```
/opt/bluedata/common-install/scripts/bluedata-report.py --gen [1,2,3]
```

where `1`, `2`, or `3` is the level of detail to include in the support bundle.

SLES

Execute the following command:

```
/opt/bluedata/common-install/scripts/bluedata-report-sc.py -- gen [1,2,3]
```

where `1`, `2`, or `3` is the level of detail to include in the support bundle.

NOTE: The name of the SLES version of this script is different than the RHEL version.

There are two ways to collect support bundles:

- Using the **Support/Troubleshooting** screen, as described in The Support/Troubleshooting Screen.
- Manually collect support logs from one or more hosts.

To manually collect support logs from the hosts, execute the following command on the hosts you want to collect HPE Ezmeral Runtime Enterprise logs as either:

- The user who installed HPE Ezmeral Runtime Enterprise

- `root`
- or using `sudo`

Executing the appropriate script generates aggregated `.txz` files and places them in `/tmp/bluedata/sosreport/customerlog`, which is the same directory used by the web interface when collecting Support Bundles, meaning that you can view and access script-generated reports using the **Support Bundles** tab of the **Support/Troubleshooting** screen (see Support Bundles Tab). The generated `.txz` file will have a name similar to `bdsupport-L1-06-30-20-08-01_66e23857a14f46e0802ea8bda2b1ddf8.txz`.

This file is an aggregation, meaning that it contains one or more embedded `.txz` files. You must therefore `untar` the generated .txz file and then `untar` the embedded `.txz` files to view the Support Bundle contents. The command to do this is:

```
tar Jxvf <filename>.txz
```

CAUTION: All of the files in the `hpecp` directory contained within each support bundle are sanitized for MAC and IP addresses. However, not all of the .txt files can be sanitized during support bundle generation. If you want to remove this information for security purposes, then you must find and remove it manually.

More information

Support and Other Resources on page 75

Support Bundle Contents

A support bundle consolidates the following diagnostic and configuration information from both the RHEL/Centos system and the HPE Ezmeral Runtime Enterprise software:

- **Container Platform data:**
  - All of the logs described in Platform Logs.
  - SASL (Simple Authentication and Security Layer) logs
  - Container diagnostics and logs
  - Data Server and SASL logs
- **System data:**
  - Hardware (hardware components and other BIOS information; `dmidecode` output)
  - Kernel modules (`lsmod` output)
  - System diagnostics (memory/BIOS)
  - Memory status
  - Package/YUM and Anaconda logs
  - Nagios configurations and log
  - Network/Open vSwitch (OVS) configurations and logs
  - Apache/`httpd` configuration and logs

More information

Support and Other Resources on page 75

Troubleshooting Overview

HPE Ezmeral Runtime Enterprise depends on the proper functioning and configuration of its underlying components such as hardware, network, security, and external storage. Platform functionality also depends on successful integration with a wide variety of Big Data applications and services, and other enterprise systems. This section describes operational and troubleshooting scenarios for operations and internal customer support teams.

NOTE: Hewlett Packard Enterprise does not provide direct support for third-party or open source Big Data components; however, Hewlett Packard Enterprise Technical Support can assist with deploying the platform with these components. It is the customer's responsibility to configure those components and to ensure that other applications function in the desired configuration before escalating an issue to Hewlett Packard Enterprise Technical Support. In addition, Hewlett Packard Enterprise may work with and provide information to a third-party vendor. If a defect in a third-party component causes the deployment to perform less optimally, then Hewlett Packard Enterprise will help the customer resolve the solution with the appropriate vendor or open source.

The following articles break down the key platform services and monitoring/alerting hooks. They also provide instructions for diagnosing and correcting common errors.

- **Services:** These articles help you troubleshoot services and restart services on Gateway hosts.
  - **Troubleshooting Services:** See Troubleshooting Services.
  - **Gateway Services:** See Gateway Services.
- **Basic Troubleshooting:** This article contains information on some of the most common issues and solutions. See Basic Troubleshooting.
- **Kubernetes issues:** These articles describe troubleshooting Kubernetes issues.
  - **Misc. Kubernetes Issues:** See General Kubernetes Application/Deployment Issues.
  - **Cluster Creation:** See Kubernetes Cluster Creation Issues.
  - **Installation:** See Kubernetes Installation Issues.
  - **Nodes:** See Kubernetes Node Issues.
  - **Node Port Services:** See Kubernetes Node Port Service Issues.
  - **Kubernetes Pods:** See Kubernetes Pod Issues.
  - **Kubernetes Tenant Management:** See Kubernetes Tenant Management Issues.
  - **Web Interface:** See Kubernetes Web Interface Issues.
- **Issue Resolution:** These articles describe how to troubleshoot and correct a variety of issues across several categories.
  - **Virtual Node (Container) Issues:** See Container Issues.
  - **DataTap Issues:** See DataTap Issues.
  - **Interface Issues:** See Interface Issues.
  - **Kerberization Issues:** See Kerberization Issues.
  - **Miscellaneous Issues:** See Miscellaneous Issues.
  - **Networking Issues:** See Networking Issues.

last-updated: May 06, 2024

- **Storage Issues:** See Storage Issues.
- **Upgrade Issues:** See Upgrade Issues.
- **User Authentication Issues:** See User Authentication Issues.

More information

Support and Other Resources on page 75

Troubleshooting Services

The **Services** tabs of the Platform Administrator and Kubernetes Administrator **Dashboard** screens display the status of each service. The Platform Administrator **Dashboard** screen also includes general HPE Ezmeral Runtime Enterprise services, such as monitoring.

See Dashboard - Kubernetes Administrator and Dashboard - Platform Administrator on page 570.

A service that is one of the following degraded states may require troubleshooting, corrective action, or both:

- **Warning:** Yellow
- **Critical:** Red.

Audit

This service audits all user access to the platform interface, and specifically CRUD operations on clusters, but does not audit requests sent to specific Kubernetes clusters. This service runs on the Controller only.

To troubleshoot this service, view the log file on the Controller host at:

```
/var/log/bluedata/bds-audit.log
```

This log file provides a comprehensive history of all interface-level user actions and is a subset of the `bd-mgmt.log`. Contact Hewlett Packard Enterprise Support if you require assistance to resolve an issue with this service.

Caching Node

This service is a critical component for running Big Data jobs against the tenant storage, external DataTaps, or both. I/O pressure, memory issues, or incompatibility with a remote DataTap can cause issues.

In Kubernetes deployments of HPE Ezmeral Runtime Enterprise, the caching node (cnode) runs a sidecar container is always named `dtap`. When troubleshooting the caching node, you can use the standard kubectl logs commands. For example, to output the caching node log of `mypod` in `mynamespace`, enter the following command:

```
kubectl logs -f -n mynamespace mypod -c dtap
```

If this service continues to restart, or if it remains in a critical state, then contact Hewlett Packard Enterprise Support.

HA Engine

This service runs the HA process for the platform. If the status of this service is **Critical**, then contact Hewlett Packard Enterprise Support.

HA Engine logs are stored in `/var/log/bluedata/pl_ha/` and `/var/log/pacemaker`.

HA Proxy

This service runs on the Gateway hosts in the platform and is managed by the platform. If this service becomes **Critical** (red dot), then collect `/var/log/bludata/bds-mgmt.log` and `/var/log/messages` on the affected Gateway host, and then contact Hewlett Packard Enterprise Support.

Management

This service is a key component that manages the overall system, including:

- The physical hosts
- Submitting jobs
- The UI and RESTful APIs.

If this service is in a degraded state, then the web interface will not be accessible. You can access the Nagios interface directly by navigating to:

```
http://<controller-ip-address>:8085/nagios
```

The management service can fail for a variety of reasons, including:

- Low availability of resources on the Controller host.
- Disk failure on the root volume of the Controller node.

To restart this service, execute the following commands:

```
stop bds-controller
start bds-controller
```

If this service still fails, then contact Hewlett Packard Enterprise Support.

The `/var/log/bluedata/bds-mgmt.log` file contains detailed interface-based operations, including:

- CRUD of various objects such as tenants, DataTaps, clusters, and flavors.
- Errors related to cluster creation failures, network connectivity issues between containers.
- Other related items.

Restarting Services

After restarting the monitoring container, services might fail to start.

To restart the management service, see Management on page 931.

To restart gateway services, see Restarting Gateway Services on page 931.

To restart a service manually, on the Controller host, execute the following command:

```
docker exec <id-of-dontainer-running-the-monitoring-image> service
metricbeat restart
```

More information

Support and Other Resources on page 75

Restarting Gateway Services

Some issues with Gateway services on HPE Ezmeral Runtime Enterprise can be resolved by restarting the `epmd`, `haproxy`, `beam.smp`, and `nagios` services on the Gateway hosts.

Prerequisites

Required access rights: Platform Administrator

About this task

Some issues with Gateway services can be fixed by restarting services on each Gateway host. Perform this task on each Gateway host in the deployment.

Procedure

1. Execute the following commands:

```
pkill -9 epmd
pkill -9 haproxy
pkill -9 beam.smp
```

2. Restart the `epmd`, `haproxy`, and `beam.smp` services by executing the following command:

```
service bds-controller restart
```

3. Restart the Nagios container:

```
# Fetch the container id of the nagios container
CONTAINER_ID=$(docker ps | grep epic-nagios | awk '{print $1}')

# Restart the nagios container
docker restart $CONTAINER_ID
```

4. Wait until the services have restarted.

Basic Troubleshooting

This article contains basic troubleshooting steps for some common issues that may occur while using HPE Ezmeral Runtime Enterprise.

PROBLEM/IMPACT	PREPARATION/BEST PRACTICE	RECOVERY PROCEDURE
Remote DataTap HDFS storage failure	Follow all applicable best practices and other guidelines from your storage vendor to mitigate any storage failure.	Obtain any applicable recovery procedures from your storage vendor. The Caching Note (`cnode`) service will automatically retry when certain HDFS errors occur before propagating the error to the application/interface.

Controller host failure When the Controller host fails: • If platform HA is enabled, then any pending virtual cluster and/or tenant creation will resume when the Shadow Controller takes over. • Virtual clusters that are already running will continue to run, and users can continue interacting with them normally, either directly (routable container network) or via the Gateway hosts (non-routable container network). • Running jobs may be interrupted, and the affected users may need to restart the affected jobs.	HPE recommends storing any critical system files (such as custom keytab files and TLS certificates) on a shared file server that is mounted on both the Primary and Shadow Controller hosts.	The Arbiter host will detect the primary Controller host failure and begin a failover transition to the Shadow Controller host. The deployment will then be running in a degraded state until the Shadow Controller becomes the primary Controller. The interface will be in Lockdown mode during the transition, and no administration tasks will be possible during this period. Users may need to restart any running jobs that were interrupted as a result of the failure/transition.
Shadow Controller host failure If the Shadow Controller host fails or crashes, then the primary Controller host will continue operating; however, the platform will be running a degraded state and will not be protected against any failure of the primate Controller host. The interface displays a warning message when this occurs.	See Controller host failure, above	HPE Ezmeral Runtime Enterprise analyzes the cause of the host failure and attempts to recover the failed host automatically. If recovery is possible, then the failed host will come back up, and HPE Ezmeral Runtime Enterprise will resume normal operation. If the problem cannot be resolved, then the affected host will be left in a degraded state. You will need to manually diagnose and (if possible) repair the problem, and then reboot that host. If rebooting solves the problem, then the failed host will come back up, and HPE Ezmeral Runtime Enterprise will resume normal operation with High Availability protection enabled. Container Platform does not currently support designating another Worker host as the new Shadow Controller. Please contact HPE Technical Support for assistance if you are unable to resolve the issue.
Arbiter host failure If the Arbiter host fails or crashes, then the Controller and Shadow Controller hosts will continue operating; however, the platform will be running in a degraded state and will not be protected against any failure of the Controller or Shadow Controller host. The interface displays a warning message when this occurs.		See Shadow Controller host failure, above.

Gateway host failure The Gateway host may fail or crash while one or more users are connected to virtual clusters.	HPE highly recommends setting up multiple Gateway hosts to provide both High Availability and load balancing.	IT can either diagnose and repair the failed Gateway host, or provision a new Gateway host. If the deployment has two or more Gateway hosts, then sessions connected through the failed hosts will be moved to the available hosts. The in-flight TCP connections might need to reset as they are moved to the backup host. The deployment includes a load-balancer in front of the Gateway hosts, and users should therefore experience no performance impacts.
Expired TLS certificates or Keytab files The underlying KDC keytab files that the virtual cluster uses to access the HDFS may be expired. Hadoop services will not be able to run because they cannot access the underlying HDFS.	Ensure that expiration date of all applicable SSL certificated and/or keytab files is sufficient for the lifespan of the cluster.	

HPE Ezmeral Runtime Enterprise can experience adverse operational impact caused by changes in system files or system settings.
Changes to various system settings may cause unpredictable behavior in HPE Ezmeral Runtime Enterprise. Some examples include:
• `/etc/sysconfig/iptables`
• `/etc/sysconfig/network`
• `SELinux` context
• `/etc/rsyslog.d/bds`
• `umask` settings
• `ipforward` settings
• RPM package deletion/changes.
  • Do not manually install Network Manager
  • RHEL subscription becomes inactive
• Do not delete or alter any service user accounts.
	Coordinate with your system administration teams to ensure that Chef/Puppet or other configuration management systems do not modify these settings/files on the HPE Ezmeral Runtime Enterprise hosts.
	On either a regular basis (e.g. weekly) or when there is a significant configuration change in the environment (e.g. OS patch update, network configuration change), perform the HPE Ezmeral Runtime Enterprise configuration checks described in Config Checks Tab, and pay attention to any problem/warning reported by these checks.
		Contact HPE Technical Support for assistance.

More information

Support and Other Resources on page 75

HPE Kubernetes Cluster Troubleshooting

Troubleshooting Kubernetes clusters that are running the Hewlett Packard Enterprise distribution of Kubernetes can involve examining the `.service` files, environment variables, and using journald to examine logs.

This article contains troubleshooting information that is specific to Kubernetes clusters that are running the Hewlett Packard Enterprise distribution of Kubernetes, which uses the containerd runtime.

Service Files

Each Kubernetes service has its own `.service` file.

- `/etc/systemd/system/kubelet.service`
- `/etc/systemd/system/kube-proxy.service`
- `/etc/systemd/system/kube-apiserver.service`

Environment Variables

Environment variables for some of the services that `ezctl` configures are saved at the following location:

```
/opt/ezkube/bootstrap/systemd
```

The environment variables are editable, but changes might be overwritten when updating Kubernetes.

Logs

Journald manages the logs for all the services.

- By default, all log messages are logged to: `/var/log/messages`
- To get the log for a service, enter the appropriate `journalctl` command.

  For example, the following command jumps to the end of the log of the `kube-apiserver` unit:

  ```
  journalctl -e -u kube-apiserver
  ```

Kubernetes Audit Information

The Kubernetes audit policy is defined in the following file:

```
/opt/ezkube/k8s-audit-policy.yaml
```

The Kubernetes audit log is contained in the following file:

```
/var/log/ezkube/audit/k8s-audit.log
```

Kubernetes Node Bring-Up

The following configuration file is used when orchestrating the Kubernetes node bring-up:

```
/var/log/bluedata/install/ezctl-config-<date>.yaml
```

The following logs are related to Kubernetes node bring-up. The contents of the logs are also logged to the standard `/var/log/bluedata/install/k8s_cluster_*` log files:

- `/var/log/bluedata/install/ezctl-<date>.log`

- `/var/log/bluedata/install/ezctl_pod_deletes.log`

Related concepts

Hewlett Packard Enterprise Distributions of Kubernetes on page 321
The Hewlett Packard Enterprise distribution of Kubernetes, identified by the `-hpe<number>` suffix, incorporates the `containerd` runtime, which is required for all Kubernetes clusters created with HPE Ezmeral Runtime Enterprise version 5.5.0 and later.

Related reference

Platform Logs on page 919

More information

Support and Other Resources on page 75

Kubernetes Issues

The topics in this section describe support and troubleshooting for Kubernetes issues in HPE Ezmeral Runtime Enterprise.

More information

Support and Other Resources on page 75

General Kubernetes Application/Deployment Issues

This article contains troubleshooting steps related to Kubernetes application and deployment issues.

Kubernetes Application Issues

See Troubleshooting Applications in the Kubernetes Documentation for instructions (link opens an external website in a new browser tab/window).

Kubernetes Pod Deployment Issues

See Troubleshooting Kubernetes Deployments for instructions (link opens an external website in a new browser tab/window).

Kubernetes Node Upgrade Issues

Kubernetes master node upgrade fails	Kubernetes upgrade assumes that pods are not scheduled to run on master nodes. Master nodes are therefore not drained during the Kubernetes upgrade process. This could cause the Kubernetes upgrade to fail if pods are running on master nodes. By default, master nodes have a `NoSchedule` taint that prevents pods from being scheduled on them. This taint should not be removed. Also, pods should not have a `NoSchedule` toleration, as this would make is possible for them to run on master nodes, even when the `NoSchedule` taint is present. If Kubernetes upgrade fails because pods are running on master nodes, the `NoSchedule` taint should be reinstated on the master nodes if it has been removed and the `NoSchedule` toleration should be removed from any pods if it has been added. After pods are no longer running on master nodes, the Kubernetes upgrade operation should be run again.
Upgrade of a Kubernetes worker node fails	Kubernetes upgrade drains worker nodes before upgrading them. If it is not possible to drain a worker node, upgrade of that node may fail. Do not configure resources such as persistent volume claims that prevent worker nodes from being drained.

If Kubernetes upgrade fails on a worker node because the node couldn't be drained (the Status message for the host says `failed to drain node`) the resources which are preventing the node from being drained should be removed from the node and the upgrade retried for any failed worker nodes.

Upgrade fails because nodes are not drained

One of the following errors occurs when upgrading Kubernetes:

- `Unable to drain node "<K8s hostname>": "<K8s hostname> cordoned error: unable to drain node "<K8s hostname>\", aborting command.`
- `There are pending nodes to be drained: <K8s hostname> error: cannot delete Pods not managed by ReplicationController, ReplicaSet, Job, DaemonSet or StatefulSet (use --force to override): default/test-pvc-pod`

HPE Ezmeral Runtime Enterprise does not force eviction of pods during the drain operation. It is likely the pod has a persistent volume (PV) attached, which is preventing pod eviction.

To complete the upgrade, manually remove the persistent volume claim (PVC) from that node.

NOTE:

When running `kubectl`, enable the `-v` (verbose) option. For example:

```
kubectl -v=10 config current-context
```

More information

Support and Other Resources on page 75

Kubernetes Cluster Creation Issues

This article contains troubleshooting steps related to Kubernetes cluster creation.

Symptom	Logs to collect/Diagnostic steps

Kubernetes cluster creation	Set-up log (from the UI) On the Controller: `/var/log/bluedata/bds-mgmt.log` On the Kubernetes master:
	`kubectl get events`
	`journalctl (or /var/log/messages)`
	`/var/log/bluedata/install/k8scluster_set-up-<time-stamp>`
	`/var/log/bluedata/install/k8scluster_set-up-<time-stamp>.xtrace`
	For additional information, click here (link opens an external website in a new browser tab/window).

Kubernetes node failed to fetch join ***Example:*** Controller: `/var/log/bluedata/bds-mgmt.log`: `Feb  5 09:53:50 dl380-002 BDS: MGMT  :[  error][  src/k8s/bd_mgmt_api_k8s.erl:01122] <0.32028.17> exception reason: {k8s_cluster_creation, ["6","failed to fetch join command"]}`	It is very likely that the Controller and Worker hosts do not have network connectivity. Possible root causes: • Firewall • Mis-configured proxy setting • Cloud (AWS) blocking traffic • Router issue On the Controller: `/var/log/bluedata/bds-mgmt.log`
Failed to execute on building K8s operator ***Example:*** Controller: `/var/log/bluedata/bds-mgmt.log` `Failed to exec: kubectl -n hpecp create -f /opt/bluedata/bundles/bluedata-HPE Ezmeral Runtime Enterprise-entdoc-minimal-debug-5.0-3002/scripts/iucomponents/k8s_cluster/operator-templates/config-crs/cr-hpecp-config.yaml ERROR: Failed executing 06_operators.sh SKIPPING rollback`	Collect the Kubernetes events On Kubernetes Master node: `kubectl get events`

Error message:
`Post https://hpecp-validator.hpecp.svc:443/validate?timeout=30s: Service Unavailable`
Example:
`kubectl -n hpecp create -f /opt/bluedata/bundles/bluedata-epic-entdoc-minimal-debug-5.0-3002/scripts/iucomponents/k8s_cluster/operator-templates/config-crs/cr-hpecp-config.yaml`
`....Error from server (InternalError): error when creating "/opt/bluedata/bundles/bluedata-epic-entdoc-minimal-debug-5.0-3002/scripts/iucomponents/k8s_cluster/operator-templates/config-crs/cr-hpecp-config.yaml": Internal error occurred: failed calling webhook "hard-validate.hpecp.hpe.com": Post https://hpecp-validator.hpecp.svc:443/validate?timeout=30s: Service Unavailable`
	It is likely there is a network problem between the Controller host and the Kubernetes hosts.
	Work with Network IT to verify that there are no connectivity issue between these hosts.

More information

Support and Other Resources on page 75

var/log/bluedata/bd Installation Issues

This article contains troubleshooting steps related to Kubernetes.

Symptom	Logs to collect/Diagnostic steps
Host installation problem on the Controller host.	On the Controller: • `/tmp/bds*` • `/var/log/bluedata/install/install_<timestamp>` • `/var/log/bluedata/install/install_<timestamp>.xtrace`
Host installation problem on the Kubernetes nodes or Worker host(s).	On the Controller host: • `/var/log/bluedata/install/addworker.out_<timestamp>.log` • `/var/log/bluedata/bds-mgmt.log` On the Worker host: • `/var/log/bluedata/install/worker_setup_<timestamp>` • `/var/log/bluedata/install/worker_setup_<timestamp>.xtrace`
Host installation problem on the Gateway host(s).	On the Controller host: • `/var/log/bluedata/install/addworker.out_<timestamp>.log` • `/var/log/bluedata/bds-mgmt.log` On the Gateway host: • `/var/log/bluedata/install/worker_setup_<timestamp>` • `/var/log/bluedata/install/worker_setup_<timestamp>.xtrace`

More information

Support and Other Resources on page 75

Kubernetes Node Issues

This article contains troubleshooting steps related to Kubernetes nodes.

Symptom	Logs to collect/Diagnostic steps

Unable to connect to the server: EOF ***Example:***
`# kubectl get nodes`
`Unable to connect to the server: EOF`
	On the Kubernetes Master: `journalctl` (or `/var/log/messages`)
	Diagnostic Steps:
	• Check if local Kubernetes API server is responding or not.
	• Try running the kubectl command from a different client.

Kubernetes node failed to fetch join. ***Example:*** Controller: `/var/log/bluedata/bds-mgmt.log: Feb 5 09:53:50 dl380-002 BDS: MGMT :[ error][ src/k8s/bd_mgmt_api_k8s.erl:01122] <0.32028.17> exception reason: {k8s_cluster_creation, ["6","failed to fetch join command"]}`	It is very likely that controller and worker does not have network connectivity. Possible root causes: • Firewall • Mis-configured proxy setting - • Cloud (AWS) blocking traffic • Router issue On the Controller: `/var/log/bluedata/bds-mgmt.log`
Failed to execute on building Kubernetes operator ***Example:*** Controller: `/var/log/bluedata/bds-mgmt.log Failed to exec: kubectl -n hpecp create -f /opt/bluedata/bundles/bluedata-epic-entdoc-minimal-debug-5.0-3002/scripts/iucomponents/k8s_cluster/operator-templates/config-crs/cr-hpecp-config.yaml ERROR: Failed executing 06_operators.sh SKIPPING rollback`	Collect the Kubernetes events. On Kubernetes master node: `kubectl get events`

More information

Support and Other Resources on page 75

Kubernetes Node Port Service Issues

This article contains troubleshooting steps related to Kubernetes NodePort services.

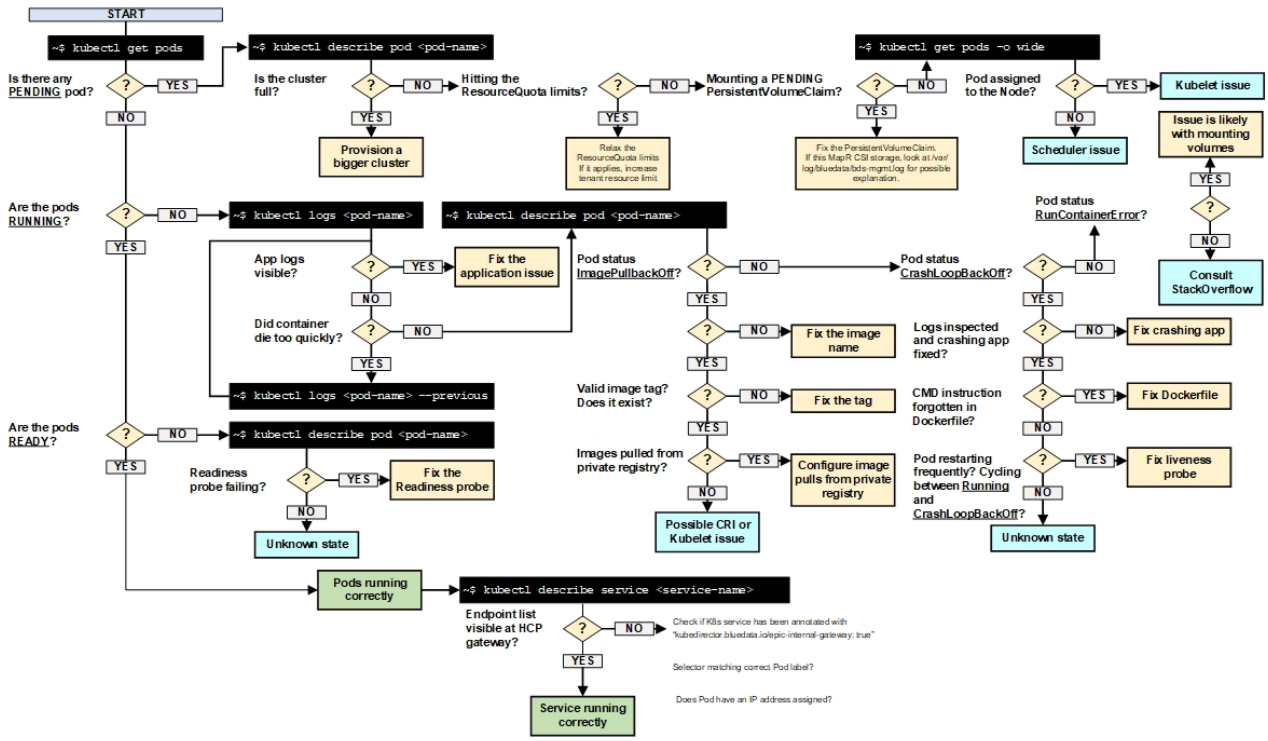
Symptom	Logs to collect/Diagnostic steps
HPE CP Gateway : Unable to annotate NodePort service	`# kubectl describe services <NodePort Service Name>`

More information

Support and Other Resources on page 75

Kubernetes Pod Issues

This diagram contains troubleshooting steps for Kubernetes pods.



More information

Support and Other Resources on page 75

Kubernetes Tenant Management Issues

Kubernetes Tenant Management Issues

Symptom	Logs to collect/Diagnostic steps
Unable to download the Kubectl plug-in from the Kubernetes **Dashboard** screens.	You may be using an unsupported browser. See Browser Requirements.

Symptom	Logs to collect/Diagnostic steps
Unable to download the Kubectl plug-in from the Kubernetes **Dashboard** screens.	You may be using an unsupported browser. See Browser Requirements.

More information

Support and Other Resources on page 75

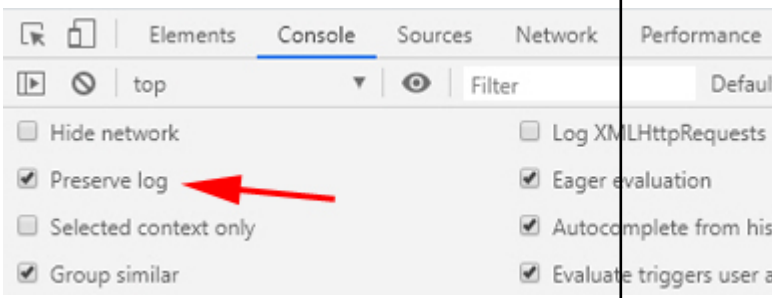
Kubernetes Web Interface Issues

This article contains troubleshooting steps related to the Kubernetes web interface.

Symptom	Logs to collect/Diagnostic steps

Web interface hangs.
 	
The browser may be present with various errors. For example:
• “Internal Server Error. The server encountered an internal error or misconfiguration and was unable to complete your request. Please contact the server administrator at root@localhost to inform them of the time this error occurred, and the actions you performed just before this error.”
• The browser is getting constant “attempting to connect” message, but not making any connection.
• A “Service Unavailable” error message appears on the screen.
	If platform High Availability is enabled, verify that you are attempting to access the correct Controller host via either the cluster IP address or the IP address of a Gateway host.
	It is possible that there was a failover and that you are trying to connect to the wrong Controller host.
	
	Validate that the same problem occurs from command line. Verify that the API service is running.
	On the Controller host, execute the following command:
	`# curl -k GET https://localhost:8080`
	or
	`# curl -k GET https://<controller-IP>:8080`
	
	Expect to receive a "Could not resolve host" message.
	`# curl -k GET  https://localhost:8080`
	`curl: (6) Could not resolve host: GET; Unknown error`
	If this command works, then the management server is working properly, and the problem is in either the web browser or the connection to the Controller host.
	`# curl https://localhost:8080/`
	`curl:  (7) Failed to connect to ::1: No route to host`
	It is possible that nothing is listening on port 8080. To verify:
	`# netstat -nlp | grep 8080`
	
	Double-check that the HPE Ezmeral Runtime Enterprise Controller host is running.
	On the Controller host, verify that the HPE Ezmeral Runtime Enterprise Controller service is up.
	`# systemctl status bds-controller`
	`# systemctl status bds-worker`
	If it is down, then you need to start it up. See Manually Restarting Services. If the bds-controller service is enabled and active, then proceed to the next step.
	Verify that the HPE Ezmeral Runtime Enterprise management service is responding. Run a basic CLI command to verify that the management service is active and responding.
	`# bdconfig --getallenv`
	Check if the Apache Server has encountered an issue. Look for obvious issues in the following files on the Controller host:
	`/var/log/httpd/error_log`
	`/var/log/httpd/access_log`
	You may need to search online for solutions based on any significant errors. If you suspect this is an Apache `httpd` transient problem or need to reproduce

Unable to download the Kubectl plug-in from the Kubnernetes **Dashboard** screens.	You may be using an unsupported browser. See Browser Requirements.

General error or hang in the UI.	**Collect Apache logs**
	On the Controller:
	`/var/log/httpd/error_log`
	`/var/log/access_log`
	Collect diagnostic data from browser Turn on Developer Mode. (On the Chrome browser, right click, and then select **Inspect**.)
	• Select the **Console** tab, click the **Settings** icon (gear), and then check the **Preserve log** check box.
	 
	
	• Repeat this for the **Network** tab.
	• Reproduce the UI problem, and then examine the debugging details.

More information

Support and Other Resources on page 75

General Issues

The topics in this section describe support and troubleshooting for general issues in HPE Ezmeral Runtime Enterprise.

More information

Support and Other Resources on page 75

Container Issues

There are two common ways to log into a container:

- `ssh -i <pem_file> bluedata@<IP-address-of-container>`
- `docker exec -it <hostname> bash`

Symptom	Recommended Action

Error message `Checking erlang rpc ports: FAILED. Port(s) 9000,9001 must be available.`
This error occurs when these ports are already in use by other applications.
	Execute the following commands to find out which process is using these ports:
	`# ss -lntp | grep 9000`
	`# ss -lntp | grep 9001`
	To resolve this error:
	• If only the `epmd` process is running, then it can be terminated safely.
	• If `beam.smp` is running, then HPE Ezmeral Runtime Enterprise is already installed.
	• Otherwise, remove the other applications or processes to free up these ports.

Yum update errors may occur because some of the required RPMs are not accessible.	Verify that all the required RPMs are accessible and are not on a block list. The Kubernetes RPMs file for air gap installations contains the required RPMs. See Configuring Air Gap Kubernetes Host Settings on page 868. If you need a separate list of RPMs for this version of HPE Ezmeral Runtime Enterprise, contact your Hewlett Packard Enterprise support representative.
Installation fails because of a security/permission problem.	Collect and review `/var/log/secure` to determine the cause of the problem.
Application tasks take longer when running on a specific host.	Determine whether the performance impact is caused by HPE Ezmeral Runtime Enterprise by running the same application on another host which is not running HPE Ezmeral Runtime Enterprise. If the performance problem is observed only on those hosts running HPE Ezmeral Runtime Enterprise, then disable virtual node assignment for the affected host until the problem is resolved.
Pods exit prematurely.	Access the **Host(s) Info** and **Services Status** tabs of the **Cluster Details** screen to determine whether the nodes or services are experiencing issues (yellow **Warning** or red **Critical** dots).

More information

Support and Other Resources on page 75

DataTap Issues

If creating a DataTap fails, you can view the following logs on the primary Controller to diagnose the issue:

```
/var/log/bluedata/bds-mgmt.log
/var/log/bluedata/bds-dataserver.log
/var/log/bluedata/bds-cachingnode.log
```

For more information about the logs, see Platform Logs on page 919.

Symptom	Troubleshooting/Resolution
Server `<Kerberos TGT>` not found in Kerberos database.	This may be due to a Kerberos configuration issue. Collect and analyze `/var/log/bluedata/bds-cachingnode.log` for information about the problem.

Container is reporting an HDFS error when running a Kerberized DataTap.	**1.** Determine whether the DataTap is configured for proxy or passthrough mode. **2.** Collect and analyze `/var/log/bluedata/bds-cachingnode.log` from the physical host where the container resides. **3.** If the `cnode` log indicates a connect() error, then collect the following: • `core-site.xml` and `hdfs-site.xml` on the remote HDFS. • Get the Datanode log file. • Verify that the Datanode is listening on the port 1004 for non-Kerberized or on port 50010 for Kerberized DataTap by executing the command `netstat -tulnp | grep <port_num>`, where `<port_num>` is the port number. • Verify that the HPE Ezmeral Runtime Enterprise Worker host can ping the DataNode and vice-versa. • Copy the file to the remote HDFS directly by executing the command `hdfs://namenode:port/path` in the Kerberized compute cluster to bypass the HPE Ezmeral Runtime Enterprise Java client code.

Related reference

Platform Logs on page 919

More information

Support and Other Resources on page 75

Elasticsearch Issues

This article describes the following Elasticsearch troubleshooting procedures:

- Elasticsearch Architecture
- Querying Elasticsearch
- Unstable Elasticsearch Service
- Cleaning up Elasticsearch Indices

Elasticsearch Architecture

HPE Ezmeral Runtime Enterprise generates the following performance metrics at short time intervals from the Docker `stats` API and `cgroup` data:

- Memory usage
- CPU load
- Network throughput in both Docker containers and on Worker hosts.

This information:

- Populates the HPE Ezmeral Runtime Enterprise **Dashboard** screens. If these screens display current data that is being constantly refreshed, then Elasticsearch monitoring is functioning correctly.

The Metricbeat service runs in the `epic-monitoring` containers on each Controller, Kubernetes Worker, and EPIC Worker host. This service collects the metrics and forwards them to the Elasticsearch database. For deployments without Platform HA (single Controller host), the Elasticsearch database is a single-node cluster hosted only on the Controller host. For deployments with Platform HA enabled, Elasticsearch runs as a 3-node cluster across the Primary Controller, Shadow Controller, and Arbiter hosts. In this case, the Elasticsearch master is chosen using the standard master selection process and does not necessarily reside on the Primary Controller host.

The Elasticsearch service is containerized; however, the database and logs are stored in `/var/lib/monitoring` on the underlying physical host(s) that make up the Elasticsearch cluster. Verify that this directory has enough disk space on each host.

If the **Dashboard** screen are not displaying current, updated data, then the monitoring stack is either degrading or has failed. To check this:

1. Open the **Services** tab of the Platform Administrator **Dashboard** screen (see Dashboard - Platform Administrator on page 570).
2. In the **BlueData** section of the screen, look for the two **Monitoring** columns.
   - The left-hand column displays the status of the `epic-monitoring` collector service running on each host (Metricbeat).
   - The right-hand column is for the monitoring database (Elasticsearch cluster). If Platform HA has been enabled, then three dots appear in this column (one for each node in the Elasticsearch cluster).

Querying Elasticsearch

The Elasticsearch service listens on port 9210 of its host nodes. (This is different than the default ELasticsearch port 9000.) Authentication is provided by the SearchGuard service. You must therefore supply a username and password with all your queries. Execute the following commands to obtain the username and password on an HPE Ezmeral Runtime Enterprise host:

```
bdconfig --getvalue bdshared_elasticsearch_admin
bdconfig --getvalue bdshared_elasticsearch_adminpass
```

You may now query the database to verify that the Elasticsearch service is listening.

Unstable Elasticsearch Service

An insufficient Java memory heap can cause stability issues that may bring down the Monitoring Database service. These errors will appear as follows:

- The graphs in the **Dashboard** screen will be empty, except that a spinning icon may appear where the graphs would be.
- The **MONITORING DATABASE** section of the **Services** tab of the Platform Administrator **Dashboard** screen may display red dots.

To confirm the issue, check `/var/lib/monitoring/logs/hpecp-monitoring.log` for the following errors:

```
Caused by: java.lang.OutOfMemoryError: Java heap space
[2018-05-30T16:45:39,564][ERROR]
[o.e.b.ElasticsearchUncaughtExceptionHandler] [] fatal
error in thread [elasticsearch[CJ_O7Il][fetch_shard_store][T#49]], exiting
java.lang.OutOfMemoryError: Java heap space
[2018-05-30T16:45:39,579][ERROR]
[o.e.b.ElasticsearchUncaughtExceptionHandler] []
fatal error in thread [elasticsearch[CJ_O7Il][bulk][T#1]], exiting
java.lang.OutOfMemoryError: Java heap space
```

The following procedure will increase the size of the Java heap allocation in each of the monitoring containers housing the Elasticsearch service. You must perform this procedure on the following host(s):

- **When platform HA is not enabled:** Controller host only.
- **When platform HA is enabled:** Controller, Shadow Controller, and Arbiter hosts.

To increase the size of the Elasticsearch Java heap:

1. SSH into the physical host.
2. Find the ID of the monitoring container by executing the command `# docker ps -a`.

   You will see a result similar to the following:

   ```
   CONTAINER ID          IMAGE
           2101ffa232f3          epic/monitoring:1.1
   ```

3. Access the monitoring container by executing the command `# docker exec -it <container_id> bash`
4. Modify the `jvm.options` file by expanding the Java memory heap to 4GB (near Line 22):

   ```
   # vi /etc/elasticsearch/jvm.options
           -Xms4g
           -Xmx4g
   ```

5. Save the file and quit.
6. Press [CTRL]+[D] to detach from the container.

7. Restart Elasticsearch by executing the following command:

```
# /opt/bluedata/bundles/<epic install bin folder>/
startscript.sh --action enable_monitoring
```

This procedure should resolve any Elasticsearch stability issues caused by the Java heap size.

Cleaning up Elasticsearch Indices

This process applies if you need to clean-up Elasticsearch indices and restore then monitoring service by refreshing a crashed Elasticsearch instance, deleting the indices data, and restoring monitoring. In this section, the generic terms *host* and *hosts* refer to the Controller and, if platform HA is enabled, the Shadow Controller and Arbiter.

NOTE: This process will delete all of the metrics data stored in Elasticsearch.

1. Execute the following command on the Controller host and, if platform HA is enabled, the Shadow Controller and Arbiter hosts:

```
systemctl stop bds-monitoring
```

2. Remove the monitoring container on each host by executing the following command on each host:

```
docker rm -f HPE Ezmeral Runtime Enterprise-monitoring-<host-ip>
```

3. Check for dangling processes on each host:

```
ps -ef | grep 'elasticsearch\|filebeat\|metricbeat\|devcron\|supervisord'
```

The result should show only two processes running. For example:

```
root      4127  4053  0 Apr06 ?        00:00:01 /usr/bin/
python2 /usr/bin/supervisord -c /etc/supervisord.conf
        root     15218 13147  0 00:37 pts/2
00:00:00 grep --color=auto elasticsearch\|filebeat\|metricbeat\|devcron\|
supervisord
```

If processes are still running, then reboot the host.

NOTE: If platform HA is enabled, then be sure to reboot the Arbiter host first, then the Shadow Controller host, and then (once all HA services are back up) the Primary Controller host. You may want to execute the command `bdconfig --hafailover` on the current Primary Controller host to fail-back to the original primary/shadow configuration.

4. Delete the Elasticsearch indices directory on the Controller host and, if platform HA is enabled, the Shadow Controller and Arbiter hosts by executing the following command:

```
rm -rf /var/lib/monitoring/elasticsearch/nodes
```

5. On the Primary Controller host, restart monitoring by executing the following command:

```
# /opt/bluedata/bundles/<epic install bin folder>/
startscript.sh --action enable_monitoring<version-build></version-build>
```

6. Verify that Elasticsearch is running by querying the indices on the Controller host and, if platform HA is enabled, on the Shadow Controller and Arbiter hosts by executing the following command:

```
curl -u elastic:$(bdconfig --getvalue bdshared_elasticsearch_adminpass)
https://localhost:9210/_cat/indices
```

The output should look like this:

```
green open metricbeat-6.6.1-2020.03.10    VUXBNUmnRg-56PyvrWgZEw 5 1
78030 0  36.3mb    18mb
        green open nvidiagpubeat-6.5.5-2020.03.11 gVixDuwdS1q1mmTiNorGIQ
5 1     678 0 818.9kb 409.4kb
        green open nvidiagpubeat-6.5.5-2020.03.10 kyxwozAYTqacasI50irMqQ
5 1      70 0   402kb   201kb
        green open metricbeat-6.6.3-2020.03.10    B7TCuV7qRMuOffO7IgIhCA
5 1   86490 0  94.1mb  47.4mb
        green open metricbeat-6.6.1-2020.03.11    Jm9yQe1gQTyXATMMU3Ez_w
5 1  739983 0 324.9mb 168.1mb
        green open bdlogging_v1-6.6.1-2020.03.11  K7A2iEQyS2GtJH8u0n15Aw
5 1 6799977 0   7.5gb   3.6gb
        green open metricbeat-6.6.3-2020.03.11    PlxsbXABQEiETLCG04ewLg
5 1  382342 0  84.6mb 193.5mb
        green open searchguard U                 _dvWvihTHmqAR5Kxw3F6UVA
1 2       5 0 117.2kb  39.9kb
        green open .kibana_1                      Vy567V6US-Sp-JHUi9AE7A
1 1       3 0  29.1kb  14.5kb
        green open bdlogging_v1-6.6.1-2020.03.10  tAR-jf0eS-uOgWlMbWGINw
5 1 1656180 0   1.6gb 851.1mb
```

More information

Support and Other Resources on page 75

HPE Ezmeral Data Fabric Issues

You can view the status of HPE Ezmeral Data Fabric services in the following locations:

- **Virtual clusters:Services** tab of the **Cluster Details** screen, or the **Services** tab of the **Training Cluster Details** or **Deployment Cluster Details** screen, as appropriate.
- **Kubernetes virtual clusters:Services Status** tab of the **Kubernetes Cluster Details** screen.

Checking Service Status

If the HPE Ezmeral Data Fabric (**MapR**) service does not appear in any **Services** tab, then it may not be running. You can determine the status of this service by executing the following commands:

- **Deployment Controller host:** `docker ps -a`
- **Kubernetes Data Fabric Master node:**`kubectl get po -A` (if the deployment includes a Kubernetes Data Fabric cluster)

Troubleshooting Errors

This article provides guidance in case any of the HPE Ezmeral Data Fabric services go into an ERROR state (red dot), or if you need to remove stale node IDs.

HPE Ezmeral Data Fabric Service	Description	Diagnostics Steps / Corrective Action

Container Location Database (CLDB)	Tracks critical metadata about every container in Data Fabric, cluster file servers, and node activity. The CLDB service on multiple nodes distributes lookup operations across those nodes for load balancing and also provides high availability.	Look at `/opt/mapr/logs/cldb.log`. Restart CLDB services, as described here (link opens an external website in a new browser tab/window).
Warden	A light Java application that runs on all the nodes in a cluster and coordinates cluster services. Warden's job on each node is to start, stop, or restart the appropriate services, and allocate the correct amount of memory to them.	Get more context on the error by looking at the Warden logs located at `/opt/mapr/logs/warden.log` in the HPE Ezmeral Data Fabric container. Refer to the troubleshooting steps here (link opens an external website in a new browser tab/window). Consider restarting the Zookeeper and Warden services, as described here (link opens an external website in a new browser tab/window.
Posix Clients	HPE Ezmeral Data Fabric POSIX clients allow Docker to read and write directly and securely on the filesystem exposed by HPE Ezmeral Data Fabric FUSE (Filesystem in Userspace).	Look at `/opt/mapr/logs/posix-client-basic.log`. Turn on HPE Ezmeral Data Fabric tracing to collect more information, as described here (link opens an external website in a new browser tab/window.
AdminApp	This is the web application that allows users and administrators to control and configure an HPE Ezmeral Data Fabric cluster.	Look at `/opt/mapr/apiserver/logs/apiserver.log`. The admin application is normally controlled by the Warden process, which should restart it if it fails. The primary repair action is to tell the warden on the appropriate node to restart this service.
Zookeeper	ZooKeeper is a coordination service for distributed applications. It provides a shared hierarchical namespace that is organized like a standard file system.	Look at `/opt/mapr/zookeeper/zookeeper-3.4.11/logs/`.
Fileserver	The `mapr-fileserver` service is the actual process that stores data on disks. This service needs to be running on every machine that is storing data. Having more file servers running will increase both failure tolerance and overall I/O bandwidth.	Look at `/opt/mapr/logs/mfs.log*`. The warden will try three times to restart the service automatically. After an interval (30 minutes by default), the warden will again try three times to restart the service. The interval can be configured using the parameter `services.retryinterval.time.sec` in the `warden.conf` file. If the warden successfully restarts the File Server service, then it should return back to NORMAL (green) status. If the warden is unable to restart the File Server service, then you may need to contact HPE Technical Support.

Removing staleid node records from the MapR cluster	A `staleid` node record in the HPE Ezmeral Runtime Enterprise list of Data Fabric cluster nodes indicates that the host was removed from the deployment and then re-added within five minutes. This record will appear underneath a valid record for the same host. There is no problem with the cluster or deployment, and you can safely delete this record.
	To check for records in this state, execute the following command on the primary Controller host:
	`bdmapr maprcli node list -columns h,svc,id`
	Stale IDs will appear in the output as shown here:
	`host.enterprise.net! fileserver,mastgateway,hostats,posixclientbasic@ ! 16.143.22.202 0 7640315902262614304`
	`host.enterprise.net_staleid_4873757504540959328 16.143.22.202 4 4873757504540959328`
		Execute the following command to delete the `staleid` using the full hostname:
		`/usr/lib/python2.7/site-packages/bluedata/mapr/bds-mapr-config.py removeNode --host-name <hostname>_staleid_4873757504540959328`
		Do not delete the corresponding valid host entry, which does not have the `staleid` reference.

More information

Support and Other Resources on page 75

Web Interface Issues

Symptom	Recommended Action

The web interface (GUI) is not responding	Restart the web server.
	• On RHEL and CentOS, on the Controller, enter the following:
	`systemctl restart httpd`
	The web server log is the following: `/var/log/httpd`
	• On SLES, on the Controller, enter the following:
	`systemctl restart apache`
	The web server log is the following: `/var/log/apache`

Symptom	Recommended Action

Unable to log in to the web interface.	Check the `httpd` status on the Controller host:
	`service httpd status /var/log/http`
	Check the Controller host status by executing the following command:
	`status bds-controller`
	Check whether the iptables setting has changed after rebooting the Controller host.

More information

Support and Other Resources on page 75

Kerberization Issues

Symptom	Troubleshooting/Resolution
Unable to create a Kerberized cluster.	The `Guestconfig.log` shows `kinit Cannot contact any KDC for realm <KDC Realm name> while getting initial credential`. • Validate proper KDC configuration by creating a non-Kerberized cluster, and then Kerberize the cluster manually. Consider using a generic utility cluster. • Create a CentOS or RHEL utility node, and then update `krb5.conf`, `kadmin.acl`, and `kdc.conf` with the correct KDC information, and then restart these two services by executing the following commands: `/sbin/service krb5kdc start /sbin/service kadmin start` Once the services have restarted, perform either a `kinit` or `ktutil`, which should connect to the KDC server. Please see this article for details on setting up the KDC configuration (link opens an external website in a new browser tab/window).

More information

Support and Other Resources on page 75

Miscellaneous Issues

Symptom	Recommended Action
Unable to expand a virtual cluster.	Collect and analyze `bd-mgmt.log`.
Unable to create a root disk.	Collect and analyze `/var/log/secure`.
Infinite loop occurs post-configuration.	Collect and analyze all logs under `/var/log/bluedata` on the Controller host.
The `bds-controller` service stopped unexpectedly.	Collect a Level 2 support bundle, as described in Generating a Support Bundle.

last-updated: May 06, 2024

Symptom	Recommended Action
The `UserAuth` service fails on several clusters.	Collect `/var/log/sssd.log`, and then restart the `sssd` service by executing the following command: `service sssd restart`
There is a problem with the `bds-mgmt` service.	On the Controller host, collect and analyze `/var/log/bluedata/bds-mgmt.log`.
Miscellaneous DataTap issues.	Collect and analyze the following: • `/var/log/bluedata/bds-cachingnode.log` • `/var/log/bluedata/bds-dataserver.log`
Miscellaneous problem inside a container.	On the affected container, collect and analyze `/var/log/bluedata/vagent`.
A container has crashed.	**1.** On the Controller host, collect and analyze `/var/log/bluedata/bds-mgmt.log`. **2.** Log in to the physical host where the container was running using the command `bdconfig --getvms`. **3.** In `/var/log/messages`, look for `Out of memory`. **4.** Collect the following files after the container crash: • **Container Platform host:**`/var/log/messages` • **Container Platform host:**`/var/log/docker` • **Container Platform host:**`"% docker stats" output`
BDFS error 6 (`mount_point_initialization failure`) occurs.	See Storage Issues.

For 7.x OS installs, rebooting a host may cause some services that depend on network services to go down and not restart, as shown in the **Services** tab of the Platform Administrator **Dashboard** screen.	Restart the services after disabling the NetworkManager by executing the following commands on the Container Platform host:
	`systemctl stop NetworkManager`
	`systemctl disable NetworkManager`
	`systemctl restart network`
	`systemctl restart bds-controller`
	`systemctl restart bds-worker`

After restarting the monitoring container, services may fail to start, as shown in the **Services** tab of the Platform Administrator **Dashboard** screen.	On the Controller host, restart the service manually by executing the following command: `docker exec <Id-of-Container-running-"epic/monitoring"-Image> service metricbeat restart`

More information

Support and Other Resources on page 75

Networking Issues

Symptom	Troubleshooting/Resolution

Unable to load `vagent` on one container, but it works on another container.	This may be caused by an inconsistent `iptables` setting across all the Worker hosts.
	On the affected Worker host, edit `/etc/bluedata/bluedata.conf` and adjust the `iptables` setting to match that of the other Worker host(s) in the Container Platform deployment. Then restart the `bds-controller` service by executing the following commands:
	`stop bds-controller`
	`start bds-controller`

The container hostname cannot be resolved, but the container can be accessed via its IP address.	Execute the command `# ps -ax | grep dnsmasq`. If this is not pointing to `/etc/bluedata/bds-dnsmasq.conf`, then adjust this by executing the following commands:
	`sudo service dnsmasq stop`
	`sudo chkconfig -del dnsmasq`
	`sudo /usr/sbin/dnsmasq --conf-file=/etc/bluedata/bds-dnsmasq.conf`

The Controller cannot access or ping a container (virtual node).	Validate the following network settings:
	• Floating IP range
	• CIDR info
	• IP next hop address
	Collect the web interface screenshot of these settings.
	Verify that you can ping between the physical hosts. While running the ping test, collect the packet trace on both the Controller and the affected virtual node. This will help narrow down the network failure. The Gateway may be improperly configured, or the wrong CIDR may be in use. Use the `tcpdump` tool, such as:
	`tcpdump -i bond0 icmp or arp`
	Check this at the following locations:
	• **Host:** bond0/eth0
	• **Internal gateway:** `bd_public`
	• **Inside the container:** Verify that the `tcpdump` RPM package is installed, and then execute the following:
	`route -n`
	`tracepath -n <destination>`
	`traceroute <destination>`

More information

Support and Other Resources on page 75

Storage Issues

Symptom	Troubleshooting/Resolution
HDFS decommissions.	Collect a Level 2 support bundle, as described in Generating a Support Bundle, and also collect the caching node log `/var/log/bluedata/ds-hdfs-config.log` . The most probable cause is either a bad/overwritten `bdhdfs-krb5.conf` file, or expired Kerberos credentials. If the Kerberos credentials are expired, try refreshing them by executing the command `% bdhdfs hadoop dfs admin`.

BDFS Error Codes

Code	Descriptions
Posix error	See this article (link opens an external web site in a new browser tab/widow).
2	bdfs_file_or_directory_not_found
3	no_available_cluster_contexts
4	bdfs_cmd_not_supported
6	mount_point _initialization failure This failure can occur if the dtap:// is incorrectly configured. More often, it happens if there is an error obtaining SASL-authentication and connection with the name node. Some items to consider • Is this a local or remote HDFS? • Is the DataTap information entered correctly? • Has the same DataTap been accessed successfully before? Collect a snippet of the application's exception output and the bds-cachingnode log from the host where the error occurred.
7	bdfs_no_fre bdfs_eof
8	bdfs_no_free_bufinfo
9	bdfs_gethostname_failed
10	bdfs_invalid_path
11	register_vm_no_ctxts
12	register_vm_no_iocr
13	register_vm_mapping_failed
14	pso_init_localfile_open_failed
15	pso_init_localfile_ioc_srvr_failed

16	pso_init_no_ctxts
17	pso_init_gfs_volume_mount_failed
18	schedule_work_pthread_create_failed
19	bad_request
20	unsupported_worker_cmd
21	vmid_not_found
22	un_mmap_failed
23	failed_to_obtain_blobstore_authentication_token
24	failed_to_start_blobstore_worker_thread
25	blobstore_object_len_is_zero
26	ioc_not_configured
27	ioc_already_configured
28	dco_ops_thread_create_failed
29	dco_ops_list_populate_free_cmds_failed
30	dco_ops_list_put_cmd_work_list_failed
31	check box
32	invalid_gluster_server_ip
33	invalid_gluster_port
34	invalid_gluster_username
35	invalid_gluster_password
36	invalid_swift_username
37	invalid_swift_password
38	invalid_swift_auth_path
39	invalid_swift_container_path
40	invalid_localfile_username
41	invalid_localfile_password
42	invalid_dco_type
43	hdfs_worker_thread_failed
44	hdfs_no_mount_point_resource_available
45	hdfs_no_mount_point_cached
46	hdfs_new_builder_failed
47	hdfs_no_op_elements_available
48	hdfs_op_pending
49	hdfs_connect_failed
50	hdfs_disconnect_failed
51	hdfs_ro_file_does_not_exist
52	hdfs_open_ro_failed

53	hdfs_open_wo_failed
54	hdfs_close_error
55	hdfs_close_failed
56	hdfs_rename_file_does_not_exist
57	hdfs_rename_file_failed
58	hdfs_delete_file_does_not_exist
59	hdfs_delete_file_failed
60	hdfs_get_filestatus_src_file_does_not_exist
61	hdfs_get_filestatus_failed
62	hdfs_ops_malloc_failed
63	hdfs_ops_realloc_failed
64	hdfs_listdir_failed
65	hdfs_listdir_malloc_failed
66	hdfs_mkdir_failed
67	hdfs_rmdir_failed
68	hdfs_rmdir_does_not_exist
69	hdfs_read_failed
70	hdfs_write_failed
71	hdfs_seek_failed
72	hdfs_invalid_dco_op_type
73	hdfs_dco_query_failed
74	cephfs_worker_thread_failed
75	cephfs_rados_create_failed
76	cephfs_read_conf_file_failed
77	cephfs_connect_failed
78	cephfs_ioctx_create_failed
79	cephfs_ro_file_does_not_exist
80	cephfs_open_ro_failed
81	cephfs_close_error
82	cephfs_close_failed
83	cephfs_get_filestatus_src_file_does_not_exist
84	cephfs_get_filestatus_failed
85	cephfs_ops_malloc_failed
86	cephfs_ops_realloc_failed
87	cephfs_listdir_failed
88	cephfs_listdir_malloc_failed
89	cephfs_get_xfer_ctx_failed

90	cephfs_read_failed
91	cephfs_read_call_failed
92	cephfs_comp_create_failed
93	localfile_mount_failed
94	localfile_open_failed
95	localfile_close_failed
96	localfile_create_failed
97	localfile_malloc_failed
98	localfile_stat_failed
99	localfile_scandir_failed
100	localfile_rename_failed
101	localfile_unlink_failed
102	localfile_mkdir_failed
103	localfile_rmdir_failed
104	localfile_opendir_failed
105	localfile_readdir_r_failed
106	localfile_lseek_failed
107	localfile_io_failed
108	localfile_read_failed
109	localfile_write_failed
110	localfile_worker_thread_failed
111	gfs_common_init_failed
112	glfs_new_failed
113	glfs_set_volfile_server_failed
114	glfs_init_failed
115	glfs_open_ro_failed
116	glfs_open_wo_failed
117	glfs_get_filestatus_failed
118	glfs_rename_failed
119	glfs_mkdir_failed
120	glfs_rmdir_failed
121	glfs_close_error
122	glfs_close_failed
123	glfs_ops_malloc_failed
124	glfs_ops_realloc_failed
125	glfs_pwrite_failed
126	glfs_creat_failed

127	glfs_unlink_failed
128	glfs_opendir_failed
129	glfs_readdir_r_failed
130	glfs_stat_failed
131	glfs_set_logging_failed
123	glfs_path_too_long
133	cache_buffers_init_failed
134	cache_init_failed
135	cache_no_request_ctxt
136	cache_internal_error
137	unsupported_pso_type
138	pso_stat_failed
139	unsupported_request_op
140	swift_init_info_malloc_failed
141	swift_init_info_curl_failed
142	swift_get_contents_curl_op_failed
143	swift_get_contents_bad_header_status
144	swift_get_contents_curl_setup_failed
145	swift_get_data_length_curl_setup_failed
146	swift_get_data_length_curl_op_failed
147	swift_get_auth_curl_setup_failed
148	swift_get_auth_curl_op_failed
149	swift_buffer_malloc_failed
150	swift_buffer_realloc_failed
151	swift_results_array_malloc_failed
152	swift_get_auth_bad_header_status
153	swift_get_data_length_bad_http_header
154	swift_ssl_worker_thread_create_failed
155	swift_ssl_ctx_init_failed
156	swift_no_ssl_ctx_available
157	swift_ssl_connect_failed
158	swift_ssl_get_failed
159	swift_bio_connect_failed
160	swift_bio_handshake_failed
161	swift_ssl_write_failed
162	swift_ssl_read_failed
163	swift_get_contents_length_failed

164	swift_get_file_attribs_failed
165	swift_get_header_length_failed
166	swift_get_auth_token_string_not_found
167	swift_get_auth_token_string_error
168	swift_open_ro_failed
169	swift_close_failed
170	swift_invalid_op_rename
171	swift_invalid_op_mkdir
172	swift_invalid_op_rmdir
173	swift_object_len_is_zero
174	invalid_username
175	invalid_password
176	ensure_enough_memory_open_failed
177	ensure_enough_memory_read_failed
178	ensure_enough_memory_parse_failed
179	ensure_enough_memory_sscanf_free_failed
180	ensure_enough_memory_sscanf_total_failed
181	ensure_enough_memory_sscanf_cached_failed
182	ensure_enough_memory_sscanf_buffers_failed
183	ensure_enough_memory_not_enough_memory
184	invalid_gluster_auth_marshalling
185	invalid_swift_auth_marshalling
186	invalid_localfile_auth_marshalling
187	could_not_get_curl_handle
188	mmap_ioctl_set_pid_failed
189	mmap_open_failed
190	mmap_mmap_failed
191	un_mmap_munmap_failed
192	failed_to_parse_blobstore_data_object
193	register_bdfs_vm_no_ctxts
194	bdfs_mmap_open_failed
195	ioctl_bdfs_register_vm_failed
196	bdfs_mmap_pid_open_failed
197	mmap_pid_mmap_failed
198	bdfs_mmap_out_of_memory
199	bdfs_bad_manifest_offset
200	bdfs_bad_num_of_buffers

last-updated: May 06, 2024

201	bdfs_bad_num_of_mounts
202	bdfs_bad_mount_reg_phys
203	bdfs_bad_mount_points_phys
204	bdfs_bad_version
205	bdfs_bad_mount_reg_size
206	bdfs_bad_mount_points_size
207	bdfs_bad_bufinfo_size
208	bdfs_bad_manifest_size
209	bdfs_bad_bufinfo_offset
210	bdfs_not_enough_bufinfo_phys
211	bdfs_no_workers
212	schedule_bdfs_work_pthread_create_failed
213	bdfs_invalid_worker_thread
214	bdfs_dco_init_invalid_dco
215	bdfs_ops_unimplemented_command
216	bdfs_ops_unknown_command
217	bdfs_ops_unknown_type
218	bdfs_cache_open_failed
219	bdfs_cache_bad_open_type
220	bdfs_cache_read_init_failed
221	bdfs_cache_bad_write_cache_type
222	bdfs_cache_write_init_failed
223	bdfs_cache_bad_fid
224	bdfs_cache_busy
225	bdfs_cache_bad_cmd_dispatch
226	bdfs_cache_bad_cmd_len
227	bdfs_cache_bad_flush_req
228	bdfs_cache_dco_not_writable
229	bdfs_cache_dco_unrecognized
230	bdfs_cache_dco_open_failed
231	bdfs_cache_read_error
232	bdfs_cache_write_error
233	bdfs_cache_cmdlen_exceeds_bufsize
234	bdfs_cache_bufPos_mismatch
235	bdfs_cache_cmd_not_supported
236	mgmt_not_connected
237	erl_format_failed

238	bd_mgmt_rpc_failed
239	bd_mgmt_bd_info_failed
240	bdfs_erl_element_dco_tuple_failed
241	bdfs_erl_element_dco_type_failed
242	bdfs_erl_element_dco_volume_failed
243	bdfs_erl_element_dco_server_failed
244	bdfs_erl_element_dco_port_failed
245	bdfs_erl_element_dco_username_failed
246	bdfs_erl_element_dco_password_failed
247	bdfs_erl_element_dco_container_failed
248	bdfs_erl_element_dco_auth_failed
249	bdfs_erl_element_dco_jobcluster_failed
250	bdfs_erl_element_dco_namenode_failed
251	invalid_jobcluster
252	invalid_jobcluster_id
253	invalid_jobcluster_str
254	bdfs_erl_element_path_failed
255	invalid_dco_path_str
256	dco_path_alloc_failed
257	dco_clustername_alloc_failed
258	dco_jobcluster_id_mismatch
259	no_cnode_server
260	erl_rpc_to_failed
261	malloc_failed
362	get_cluster_nodes_failed
263	get_cluster_nodes_malloc_failed
264	bdfs_multi_transfer_unfinished
265	bdfs_response_dco_data_not_found
266	bdfs_bad_response_data
267	invalid_cluster_nodes_response
268	bad_cluster_node_string
269	get_cluster_nodes_bad_cmd_args
270	get_cluster_nodes_bad_node_list
271	localfs_readdir_failed
272	localfs_path_too_long
273	localfs_stat_failed
274	localfs_unlink_failed

last-updated: May 06, 2024

275	localfs_rmdir_failed
276	cnode_stats_open_failed
277	cnode_stats_thread_start_failed
278	cnode_stats_bad_vm_idx
279	cnode_stats_bad_mpc_idx
280	cnode_stats_bad_bic_idx
281	cnode_stats_mmap_failed
282	missing_create_dirs_flag
283	create_dirs_failed
284	missing_isitlocal_flag
285	invalid_hdfs_dco_namenode
286	invalid_hdfs_dco_username
287	invalid_hdfs_port
288	missing_hdfs_querydco_namenode
289	invalid_hdfs_querydco_namenode
290	missing_hdfs_querydco_port
291	invalid_hdfs_querydco_port
292	missing_hdfs_querydco_username
293	invalid_hdfs_querydco_username
294	hdfs_mount_active_list_put_failed
295	hdfs_initial_mount_active_list_put_failed
296	invalid_mr_command
297	no_available_map_file
298	bdfs_vm_no_bic_buffers
299	bdfs_vm_no_buffer_mem
300	bdfs_vm_no_streaming_buffers
301	bdfs_cache_container_setup_failed
302	register_bdfs_list_populate_free_cmds_failed
303	register_bdfs_thread_create_failed
304	register_bdfs_list_put_cmd_work_list_failed
305	ioctl_cleanup_failed
306	hdfs_rpc_connect_failed
307	hdfs_rpc_subsystem_init_list_populate_failed
308	hdfs_rcp_socket_write_failed
309	hdfs_rpc_no_free_contexts
310	hdfs_rpc_socket_failed
311	hdfs_rpc_strdup_failed

312	hdfs_rpc_gethostbyname_failed
313	hdfs_rpc_pthread_create_failed
314	hdfs_rpc_list_populate_free_req_ctxt_failed
315	hdfs_rpc_out_of_rpc_req_ctxts
316	hdfs_dataxfer_subsystem_init_list_populate_failed
317	hdfs_dataxfer_no_free_conn_ctxts
318	hdfs_dataxfer_socket_failed
319	hdfs_dataxfer_gethostbyname_failed
320	hdfs_dataxfer_connect_failed
321	hdfs_dataxfer_strdup_failed
322	hdfs_dataxfer_pthread_create_failed
323	hdfs_dataxfer_free_reqs_list_populate_failed
324	hdfs_dataxfer_invalid_op
325	hdfs_dataxfer_send_version_failed
326	hdfs_dataxfer_send_read_block_op_failed
327	hdfs_dataxfer_send_op_read_proto_len_failed
328	hdfs_dataxfer_queue_full
329	hdfs_dataxfer_list_put_incoming_failed
330	hdfs_dataxfer_not_implemented
331	hdfs_dataxfer_get_locs_rpc_send_failed
332	hdfs_dataxfer_get_blocks_has_null_response
333	hdfs_dataxfer_get_blocks_no_locations
334	hdfs_dataxfer_null_block_op_response
335	hdfs_dataxfer_block_op_resp_read_failed
336	hdfs_dataxfer_block_op_resp_read_zero_bytes
337	hdfs_dataxfer_block_op_resp_bad_varint
338	hdfs_dataxfer_block_op_resp_length_read_zero_bytes
339	hdfs_dataxfer_block_op_resp_length_read_failed
340	hdfs_dataxfer_block_op_resp_read_fewer_bytes
341	hdfs_dataxfer_block_op_resp_bad_status
342	hdfs_dataxfer_read_packet_length_failed
343	hdfs_dataxfer_packet_header_length_read_fewer_bytes
344	hdfs_dataxfer_read_packet_header_length_failed
345	hdfs_dataxfer_read_packet_header_proto_failed
346	hdfs_dataxfer_packet_header_proto_read_fewer_bytes
347	hdfs_dataxfer_bad_packet_header_proto
348	hdfs_dataxfer_read_checksums_failed

349	hdfs_dataxfer_read_checksums_read_fewer_bytes
350	hdfs_dataxfer_read_packet_bytes_failed
351	hdfs_dataxfer_read_packet_bytes_read_fewer_bytes
352	hdfs_rpc_null_rpc_ctxt
353	hdfs_listdir_remaining_entries
354	hdfs_dataxfer_create_rpc_send_failed
355	hdfs_dataxfer_get_block_locs_strdup_failed
356	hdfs_dataxfer_packet_header_length_write_failed
357	hdfs_dataxfer_packet_header_length_wrote_fewer_bytes
358	hdfs_dataxfer_packet_length_write_failed
359	hdfs_dataxfer_packet_length_wrote_fewer_bytes
360	hdfs_dataxfer_packet_header_write_failed
361	hdfs_dataxfer_packet_header_wrote_fewer_bytes
362	hdfs_dataxfer_packet_write_failed
363	hdfs_dataxfer_packet_wrote_fewer_bytes
364	hdfs_dataxfer_read_write_block_ack_failed
365	hdfs_dataxfer_null_write_block_ack_proto_message
366	hdfs_dataxfer_varint_length_read_failed
367	hdfs_dataxfer_varint_length_read_fewer_bytes
368	hdfs_dataxfer_bad_varint
369	hdfs_dataxfer_message_read_failed
370	hdfs_dataxfer_message_read_fewer_bytes
371	hdfs_dataxfer_close_rpc_send_failed
372	hdfs_dataxfer_invalid_block_type
373	hdfs_dataxfer_complete_response_null
374	hdfs_dataxfer_complete_failed
375	hdfs_dataxfer_fsync_response_null
376	hdfs_dataxfer_disconnect_null_context
377	hdfs_dataxfer_get_file_info_response_null
378	hdfs_dataxfer_get_file_status_invalid_type
379	hdfs_rpc_lease_expired
380	hdfs_rc_java_lang_assertion
381	hdfs_rpc_unknown_exception
382	hdfs_rpc_file_already_exists
383	hdfs_rw_init_failed
384	hdfs_rw_no_xfer_ctxt
385	hdfs_rw_no_addl_xfer_ctxt

386	hdfs_rw_null_bic_in_xfer_req
387	hdfs_rw_null_dcx_in_xfer_req
388	hdfs_rw_invalid_op_in_xfer_req
389	hdfs_rw_zero_blks_in_located_blocks
390	hdfs_rw_block_info_mismatches_req
391	hdfs_rw_dn_list_init_failed
392	hdfs_rw_no_dataconn
393	hdfs_rw_block_ctxt_unavailable
394	hdfs_rw_block_activated_too_soon
395	hdfs_rw_list_put_for_block_alloc_failed
396	hdfs_rw_list_put_on_block_failed
397	hdfs_rw_list_put_processing_failed
398	hdfs_rw_list_put_addl_req_seg_failed
399	hdfs_rw_list_put_active_xfer_reqs_failed
400	hdfs_rw_list_put_pending_xfer_reqs_failed
401	hdfs_rw_list_put_block_for_data_conn_failed
402	hdfs_rw_list_put_callback_items_failed
403	hdfs_rw_xfer_reqs_for_processing_init_failed
404	hdfs_rw_xfer_reqs_for_block_alloc_list_init_failed
405	hdfs_rw_blocks_for_dn_conn_list_init_failed
406	hdfs_rw_callback_items_list_init_failed
407	hdfs_rw_null_block_in_sm_loc_blk
408	hdfs_rw_bad_req_state_in_sm_loc_blk
409	hdfs_rw_bad_req_state_in_sm_ini_getblk
410	hdfs_rw_bad_req_state_in_sm_xfer_data
411	hdfs_rw_bad_op_in_sm_close_blk
412	hdfs_rw_bad_req_state_in_sm_close_blk
413	hdfs_rw_bic_close_already_in_progress
414	hdfs_rw_bic_open_is_a_dir
415	hdfs_rw_blk_has_no_data_nodes
416	hdfs_rw_blk_has_zero_len_in_located_blks
417	hdfs_rw_active_xfer_reqs_list_init_failed
418	hdfs_rw_xfer_reqs_locating_blk_list_init_failed
419	hdfs_rw_xfer_reqs_pending_list_init_failed
420	hdfs_rw_xfer_reqs_retry_list_init_failed
421	hdfs_rw_bad_req_state_in_sm_main
422	hdfs_rw_unable_to_add_block

last-updated: May 06, 2024

423	hdfs_rw_nn_wait_for_lease_renewal
424	hdfs_rw_open_ro_failed_blkctxt_alloc
425	hdfs_rw_open_wo_failed_blkctxt_alloc
426	hdfs_rw_write_bic_in_error_state
427	hdfs_socket_write_failed
428	hdfs_socket_short_write
429	hdfs_rpc_setsockopt_failed
430	hdfs_rpc_no_rpc_response
431	hdfs_rpc_failed_rpc_status
432	hdfs_rpc_connect_null_id
433	hdfs_dataxfer_null_create_response
434	hdfs_dataxfer_create_failed
435	hdfs_rw_close_for_bic_not_accessed
436	hdfs_dataxfer_recover_lease_response_null
437	hdfs_dataxfer_recover_lease_failed
438	hdfs_dataxfer_cannot_handle_block_offset
439	dco_ops_list_populate_free_rpc_contexts_failed
440	dco_ops_list_populate_free_hdfs_ops_failed
441	dco_ops_list_init_hdfs_active_failed
442	dco_ops_list_init_hdfs_pending_failed
443	dco_ops_list_init_hdfs_completed_failed
444	dco_ops_list_set_name_hdfs_active_failed
445	dco_ops_list_set_name_hdfs_pending_failed
446	dco_ops_list_set_name_hdfs_completed_failed
447	dco_ops_list_populate_free_query_tuples_failed
448	hdfs_rpc_queue_full
449	hdfs_rpc_rcv_buf_exceeded
450	hdfs_rpc_ebadf
451	hdfs_rpc_cannot_unpack_response_header
452	hdfs_rpc_pthread_mutex_init_failed
453	cluster_info_subsystem_init_list_populate_failed
454	list_put_vm_base_map_failed
455	list_put_cluster_ctxt_failed
456	hdfs_rpc_response_read_short
457	get_free_vm_base_map_failed
458	bad_hypvervisor_base_in_vm_map
459	bad_vmlist_in_tuple

last-updated: May 06, 2024

460	list_put_freevminfos_failed
461	get_free_vminfo_failed
462	no_free_vm_info
463	hdfs_setsockopt_rcvtimeo_failed
464	hdfs_setsockopt_sndtimeo_failed
465	hdfs_setsockopt_keepalive_failed
466	hdfs_rcv_sock_bytes_failed
467	hdfs_rpc_rcv_sock_bytes_failed
468	hdfs_rpc_rcv_sock_bytes_msg_failed
469	hdfs_rcv_sock_bytes_hit_a_timeout
470	hdfs_fcntl_for_blocking_failed
471	list_put_vmmap_failed
472	hdfs_dataxfer_no_free_thread_ctxts
473	hdfs_dataxfer_null_tc
474	hdfs_rpc_backup_connect_failed
475	hdfs_nn_ha_init_list_populate_failed
476	hdfs_ha_no_free_nn_info
477	hdfs_ha_strdup_for_backup_nn_failed
478	hdfs_ha_strdup_for_nn_failed
479	hdfs_rpc_primary_nn_down_no_secondary_nn
480	hdfs_nn_corrupted_current
481	hdfs_nn_info_switch_gethostbyname_failed
482	hdfs_nn_info_switch_no_backup
483	hdfs_rpc_getpeername_failed
484	hadfs_rpc_hdfs_nn_info_switch_failed
485	hdfs_nn_info_inet_ntop_for_saddr_failed
486	hdfs_nn_info_inet_ntop_for_nn_srvr_failed
487	hdfs_rcv_soc_bytes_exceeded_zero_retry_count
488	cluster_info_hdfs_rpc_send_failed
489	list_init_for_completed_cluster_info_failed
490	cluster_info_get_locations_failed
491	cluster_info_cannot_unpack_buffer
492	no_available_cluster_nodes
493	cluster_info_get_base_map_failed
494	process_get_file_locs_cmd_got_a_null_vm_host
495	get_cluster_nodes_bad_dco_info
496	get_cluster_nodes_bad_dco_component

497	get_cluster_nodes_bad_dco_specific_component
498	get_cluster_nodes_bad_islocal_flag
499	missing_cachable_dco_flag
500	handle_dco_info_cmd_strdup_failed
501	dco_cache_init_list_populate_failed
502	dco_cache_add_dco_failed
503	get_cluster_nodes_bad_dco_type
504	hdfs_nn_corrupted_nn_info
505	hdfs_rpc_inernal_connect_null_current_nn
506	hdfs_dataxfer_bad_msg_len
507	cnode_log_invalid_log_group
508	cnode_log_invalid_log_level
509	hdfs_update_nn_cache_backup_removed_not_supporte d
510	hdfs_update_nn_cache_strdup_failed
511	hdfs_rw_close_items_list_init_failed
512	hdfs_rw_list_put_close_items_failed
513	hdfs_rw_close_retries_list_init_failed
514	permission_denied
515	bdfs_dco_info_strdup_failed
516	hdfs_dataxfer_connection_refused
517	jobcluster_id_conversion_failed
518	listput_active_mpc_failed
519	hdfs_rw_open_sem_error
520	hdfs_rw_open_sem_timed_out
521	hdfs_dco_query_no_context
522	hdfs_sasl_client_new_failed
523	hdfs_sasl_client_start_failed
524	hdfs_sasl_conn_failed
525	hdfs_sasl_cannot_parse_response
526	hdfs_sasl_step_failed
527	krb_init_context_failed
528	krb_parse_name_flags_failed
529	krb_cc_resolve_failed
530	krb_kt_resolve_failed
531	krb_get_init_creds_keytab_failed
532	krb_get_init_creds_opt_alloc_failed
533	krb_get_init_creds_opt_set_out_ccache_failed

534	krb_get_renewed_creds_failed
535	krb_cc_store_cred_failed
536	krb_realm_not_found
537	krb_conf_file_open_failed
538	tmp_krb_conf_file_open_failed
539	krb_conf_file_close_failed
540	tmp_krb_conf_file_close_failed
541	krb_conf_file_lock_failed
542	tmp_krb_conf_file_lock_failed
543	krb_conf_file_parse_failed
544	krb_conf_file_rename_failed
545	unknown_hdfs_querydco_type
546	malformed_hdfs_querydco_type
547	missing_hdfs_querydco_kdc_host
548	invalid_hdfs_querydco_kdc_host
549	missing_hdfs_querydco_kdc_port
550	invalid_hdfs_querydco_kdc_port
551	missing_hdfs_querydco_keytab_file_path
552	invalid_hdfs_querydco_keytab_file_path
553	missing_hdfs_querydco_realm
554	invalid_hdfs_querydco_realm
555	missing_hdfs_querydco_principal
556	invalid_hdfs_querydco_principal
557	missing_hdfs_querydco_service_id
558	invalid_hdfs_querydco_service_id
559	hdfs_sasl_rpc_failed
560	hdfs_receive_complete_packet_packet_too_big
561	hdfs_rpc_cache_not_found
562	hdfs_rpc_reqs_list_populate_failed
563	hdfs_rpc_send_null_cc
564	hdfs_rpc_reqs_list_populate_failed
565	hdfs_rpc_send_strdup_failed
566	hdfs_rpc_send_queue_put_failed
567	hdfs_rpc_lookup_not_found
568	hdfs_rpc_duplicate_nn_cached
569	hdfs_rpc_add_nn_to_cache_list_put_failed
570	hdfs_rpc_invalid_callid

571	hdfs_rpc_send_unknown_method
572	hdfs_dco_no_free_vm_ctxt
573	hdfs_dco_vm_init_failed
574	hdfs_setsockopt_tcpnodelay_failed=
575	hdfs_rpc_send_invalid_rpc_cmd
576	hdfs_rpc_ticket_expiry_recovery_failed
577	hdfs_rpc_told_to_disconnect
578	hdfs_conn_null_cc
579	hdfs_leaseowner_list_put_failed
580	hdfs_leaseowner_list_remove_failed
581	container_map_open_failed
582	ioctl_get_container_manifest_failed
583	kerb_authorization_exception_in_authorization_exception
584	hdfs_get_server_defs_null_resp
585	hdfs_setsockopt_linger_failed
586	hdfs_socket_send_failed_epipe
587	hdfs_rpc_responses_null_cc
588	hdfs_rpc_responses_null_req
589	hdfs_rpc_responses_null_req_cb
590	hdfs_rpc_resp_cannot_unpack_rpc_hdr
591	hdfs_invalid_replication_factor
592	hdfs_invalid_block_size
593	bdfs_erl_element_bdfs_id_failed
594	invalid_bdfs_id_str
595	bdfs_erl_element_tenant_id_failed
596	invalid_tenant_id_str
597	missing_tenant_and_vm_ids
598	sem_trywait_failed_for_copy_keytab
599	keytab_copy_to_host_timedout
600	hdfs_get_server_defaults_null_resp
601	hdfs_get_server_defaults_bad_resp_params
602	hdfs_copy_keytab_no_response
603	hdfs_copy_keytab_no_response_atom
604	hdfs_copy_keytab_no_response_err_str
605	hdfs_copy_keytab_error_response
606	hdfs_copy_keytab_bad_path
607	hdfs_not_yet_replicated

last-updated: May 06, 2024

608	invalid_kdc_host
609	invalid_kdc_port
610	hdfs_info_user_is_null
611	hdfs_dataxfer_secondary_error
612	hdfs_not_replicated_yet
613	hdfs_rw_nn_op_retries_list_init_failed
614	hdfs_rw_nn_op_retries_list_put_failed
615	hdfs_null_dco_name
616	bdfs_vm_streaming_buffers_below_reserve
617	hdfs_rw_list_put_on_reqs_locating_block_failed
618	bdfs_mount_fs_null_dco_name
619	hdfs_socket_send_conn_timed_out
620	bdfs_null_bic
621	hdfs_protobuf_version_mismatch
622	hdfs_is_file_closed_false
623	missing_readonly_flag
624	krb5_call_to_system_failed
625	hdfs_null_keytab_file_name
626	localfile_set_permission_failed
627	hdfs_primary_eq_backup_nn
628	switch_retry_count_exceeded
629	bdfs_invalid_dco_name
630	hdfs_invalid_dc
631	hdfs_get_delegation_token_error
632	hdfs_sasl_client_base64_failed
633	hdfs_enable_security_no_credential_token
634	hdfs_get_token_not_passthrough_dtap
635	hdfs_use_token_not_match_service
636	hdfs_use_kerberos_not_match_realm
637	krb5_save_creds_failed
638	hdfs_rw_add_block_resp_out_of_range
639	hdfs_sasl_cannot_decode_response
640	hdfs_dataxfer_get_data_encryption_key_response_null
641	hdfs_dataxfer_get_data_encryption_key_null
642	hdfs_dataxfer_sasl_failed
643	hdfs_dataxfer_init_aes_failed
644	hdfs_dataxfer_aes_encryption_failed

645	hdfs_dataxfer_aes_decryption_failed
646	hdfs_get_ez_for_path_error
The following codes are defined by the driver in the VM: • **5000:** bdfs_drvr_cmd_timed_out • **5001:** bdfs_drvr_bad_command • **5002:** bdfs_drvr_read_no_cmd_in_progress	

More information

Support and Other Resources on page 75

Upgrade Issues

Symptom	Recommended Action

Upgrade failure	Collect these logs from the Controller host:
	`/var/log/bluedata/install/Upgrade-XXXXX`
	`/var/log/bluedata/bds-mgmt.log`

Yum update error dues to missing RPM	Verify that all of the RPMs defined in repo list are accessible and are not block listed.

More information

Support and Other Resources on page 75

User Authentication Issues

Symptom	Troubleshooting/Resolution
The message `403 Forbidden` appears when trying to revoke a user.	Verify that the revoke command is being executed as a Platform Administrator.
The Tenant Key Pair API command returns a key error, as follows: • **Command:** `http://<ip_address>:8080/api/v1/tenant/"+<tnt_id>+"?<private_key>` • **Output :** `{"private_key":"undefined"}`	This is normal behavior if the **Site Admin** tenant (`<tnt_id>=1`) is used in the command. This tenant does not have any virtual nodes/containers and therefore has no defined SSH key to return.

Unable to log in to a container using LDAP/AD credentials.	Validate the LDAP/AD credentials by executing the ldapsearch command from the Controller:
	`LDAPTLS_REQCERT=never ldapsearch -ZZ -x -h <ad_ldap_server_name> -p <port> -D <bind_dn> -w <bind_password> -b <subtree_dn> -s sub <filters_go_here>`
	For example:
	`LDAPTLS_REQCERT=never ldapsearch -ZZ -x -h 10.3.29.11 -p 389 -b 'dc=bluedata,dc=net' -s sub '(cn=john)`
	If that succeeds, then verify that the user is included in the membership defined in the `ldap_access_filter` property defined in `/etc/ssd/ssd.conf` in the container by logging in to the container as user `HPE` and then executing the command `sudo bash`.

When a user that has special characters in their Distinguised Name, for example `cn=Test1 (test1), cn=Users,...` attempts to log into a KubeDirector Notebook, the Notebook (JupyterHub) returns the error: `500: Internal Server Error.`	Change the configuration of the JupyterHub LDAP Authenticator Plugin to set `LDAPAuthenticator.escape_userdn = True`. With this configuration change, when authenticating in LDAP, the following special characters in `userdn` are escaped: `\` ,`*`,`(`, and `)`

More information

Support and Other Resources on page 75

App Workbench 5.1

Getting Started

App Workbench 5.1

Welcome! This page links you to the articles that comprise the App Workbench 5.1 documentation:

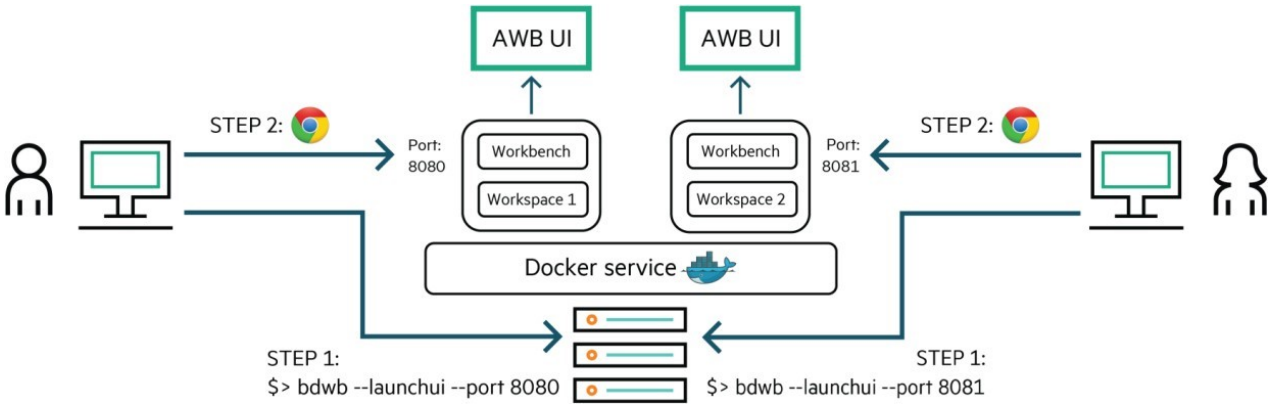
- **Getting Started:** These articles contain the information you need to get up and running with App Workbench:
  - **Architecture:** A high-level overview of the App Workbench architecture. See Architecture.
  - **What's New in Version 5.1:** New features in this version of App Workbench. See What's New in Version 5.1 on page 975.
  - **Release Notes:** Known issues and other information pertinent to this version of App Workbench. See Release Notes.
  - **Overview:** Introduction to the app-building workflow. See Overview.
  - **Browser Support:** Browser requirements for viewing the App Workbench web interface. See Browser Support.

- **Prerequisites:** Requirements for installing App Workbench. See Prerequisites.
- **Installation**: Installing App Workbench. See Installation.
- **Launching App Workbench:** Accessing the App Workbench web interface. See Launching App Workbench.
- **The Application Status Screen:** This is the screen you will see upon launching the web interface. From here, you can continue to build KubeDirector or EPIC applications. See The Application Status Screen.

- **Building KubeDirector Apps:** These articles describe the App Workbench user interface that appears when you opt to build a KubeDirector app from the **Application Status** screen. These articles appear in the order you will see them when building the application.
- **Building EPIC Apps:** These articles describe the App Workbench user interface that appears when you opt to build a legacy EPIC app from the **Application Status** screen. These articles appear in the order you will see them when building the application.
- **Custom Base Images:** These articles describe how to use custom CentOS 7 or 8, RHEL 7 or 8, or Ubuntu 18 base images for building legacy EPIC applications.
- **Resources:** These articles contain command information that you can use when building legacy EPIC applications.

Architecture

The Application Workbench for HPE Ezmeral Runtime Enterprise provides a simple interface that allows you to quickly and easily build application images for a wide variety of use cases. Version 5.1 of App Workbench is compatible with HPE Ezmeral Runtime Enterprise versions 5.1 and later. App Workbench runs as a Docker service that is loaded and instantiated via the command line and that surfaces a graphical web interface.



This page contains more information about the Application Workbench for HPE Ezmeral Runtime Enterprise, including a video introduction (link opens in a new browser tab/window).

What's New in Version 5.1

App Workbench now supports the creation of KubeDirector applications in addition to legacy EPIC applications. App Workbench 5.1 is compatible with HPE Ezmeral Container Platform versions 5.1 and later.

Release Notes

This version of App Workbench has the following known issues:

last-updated: May 06, 2024

- **EZPDM-69: Kube Director doesn't support the NVIDIA NGC-specific custom labels such as volumeMounts, tty, and stdin.**

  Workaround: The fix will be available in a future version of HPE Ezmeral Runtime Enterprise. Please contact Hewlett Packard Enterprise Technical Support.

- **EZCP-176: Kube Director Image logos don't appear in HPE Ezmeral Runtime Enterprise after deployment.**

  Workaround: The fix will be available in a future version of HPE Ezmeral Runtime Enterprise.

Overview

To use App Workbench:

1. Ensure your workstation is running one of the browsers listed in Browser Support.
2. Verify that your environment meets all of the Prerequisites.
3. Install App Workbench.
4. Launch App Workbench, and then access the **Application Status** screen. See The Application Status Screen.
5. Proceed to build your application:
   - KubeDirector
   - EPIC

KubeDirector Applications

To build a KubeDirector application:

1. Click the Create KubeDirector App button to open the **KubeDirector Application Details** screen, and then complete the fields on that screen. See The KubeDirector Application Details Screen.
2. Click the **Next** button to open the **KubeDirector Services** screen, and then complete the fields on that screen. See The KubeDirector Services Screen.
3. Click the **Next** button to open the **KubeDirector Roles** screen, and then complete the fields on that screen. See The KubeDirector Roles Screen.
4. Click the **Next** button to open the **KubeDirector Configuration** screen, and then complete the fields on that screen. See The KubeDirector Configuration Screen.
5. Click the **Next** button to open the **KubeDirector Workspace** screen, and add any files, scripts, or directories you need. See The KubeDirector Workspace Screen.
6. Click the **Next** button to open the **KubeDirector Images** screen, and then map container images to application roles. See The KubeDirector Images Screen.
7. Click the **Next** button to open the **KubeDirectorBuild** screen, and then finish building your application. See The KubeDirector Build Screen.

EPIC Applications

To build an EPIC application:

1. Click the **Create EPIC App** button to open the **EPIC Application Details** screen, and then complete the fields on that screen. See The EPIC Application Details Screen.

2. Click the **Next** button to open the **EPIC Services** screen, and then complete the fields on that screen. See The EPIC Services Screen.
3. Click the **Next** button to open the **EPIC Roles** screen, and then complete the fields on that screen. See The EPIC Roles Screen.
4. Click the **Next** button to open the **EPIC Configuration** screen, and then complete the fields on that screen. See The EPIC Configuration Screen.
5. Click the **Next** button to open the **EPIC Workspace** screen, and add any files, scripts, or directories you need. See The EPIC Workspace Screen.
6. Click the **Next** button to open the **EPIC Images** screen, and then map container images to application roles. See The EPIC Images Screen.
7. Click the **Next** button to open the **EPIC Build** screen, and then finish building your application. See The EPIC Build Screen.

Browser Support

The App Workbench web interface supports the following browsers:

- **Chrome:** latest
- **Firefox:** latest
- **Internet Explorer:** 11, 10, and 9. Compatibility View mode is not supported.

Prerequisites

The following requirements must be met in order to install and run App Workbench:

- The machine must be running a Linux operating system.
- Docker 1.13 or 19.0.3 must be installed and running.
- You must be a non-root user.
- Your user account must be a member of the Docker group. You can add a username by executing the following commands:

```
sudo groupadd docker
$ sudo usermod -aG docker $USER
or
$ sudo usermod -aG docker <your_username>
```

  NOTE: Log out and log back in after executing these commands, so that your group membership is re-evaluated.

- You have a writable workspace directory on the host you are using to run App Workbench.
- Your firewall has opened ports to access the web interface. You will specify the port to use when launching the interface, as described in Launching App Workbench.

Installation

NOTE: Do not install App Workbench on the HPE Ezmeral Runtime Enterprise Controller host.

To install App Workbench:

last-updated: May 06, 2024

1. Download the `hpecp-workbench-5.1.tgz` file.
2. Execute the following commands:

```
$ sudo tar xf hpecp-workbench-5.1.tgz
        $ ls
```

3. Load the Docker image by executing the following command:

```
$ docker load -i hpecp-workbench-img-5.1.tgz
```

4. Copy the `bdwb` file to any directory in your `PATH`. For example:

```
$ sudo cp bdwb /usr/local/bin/
```

5. Make `bdwb` executable by executing the following command:

```
$ sudo chmod +x /usr/local/bin/bdwb
```

This process will appear similar to the following:

```
                        [root@prod21 AWB5.1]# curl -JOL http://
10.0.1.107:8001/hpecp-workbench-5.1.tgz
                          % Total    % Received % Xferd  Average Speed
Time    Time     Time  Current
                                                         Dload  Upload
Total   Spent    Left  Speed
                        100  241M  100  241M    0     0   536M
0 --:--:--:--:--:--:--:--:--  538M
                        [root@prod21 AWB5.1]# tar xf hpecp-workbench-5.1.tgz
                        [root@prod21 AWB5.1]# ls
                        bdwb  hpecp-workbench-5.1.tgz
hpecp-workbench-img-5.1.tgz
                        [root@prod21 AWB5.1]# docker load -i
hpecp-workbench-img-5.1.tgz
                        e2f033e3824d: Loading layer
[=====================================================>] 23.64MB/23.64MB
                        ccbbd4cb477b: Loading layer
[=====================================================>] 23.64MB/23.64MB
                        9fe4f417cc70: Loading layer
[=====================================================>] 23.64MB/23.64MB
                        f069e0dd89b9: Loading layer
[=====================================================>] 23.64MB/23.64MB
                        Loaded image: hpecp/workbench:5.1

                        [root@prod21 AWB5.1]#
```

Docker Registries

When specifying an image to use for a role, you can either:

- Choose to reference an existing image from a Docker registry.
- Build an image from a local directory containing a Dockerfile and, optionally, some scripts to include in that image.

In general:

- If you are building an EPIC app using a locally built image, that image will be embedded in the final app binary. If you are referencing an existing image, then you must provide a registry URL from which to pull the image when the app is deployed.
- You must always provide the registry if you are building a KubeDirector app, because KubeDirector must pull the image from the registry during deployment. If you choose to build the image locally, it will be pushed to the registry that you specify.
- Hewlett Packard Enterprise recommends using a Docker Hub account to pull the images. If you are using public repository and not Docker Hub account, you may face the Docker Hub Rate Limiting issue. To create Docker Hub accounts, see Docker Hub accounts.

Image Repo Tags

Repo-tags take the following form:

```
[REGISTRY_URL/]REPOSITORY/NAME:TAG
```

You must include the registry component in the repo-tag that you provide in the App Workbench Images screen when pushing or pulling an image to/from a registry, in the same manner as using the repo-tag in a `docker pull` or `docker push` command. Omitting from the repo-tag will default to the Docker Hub Registry (`hub.docker.com`). The repo-tag usually contains the repository within the registry when specifying an image. Omitting this as well as the registry assumes that the repository is an official Docker Hub Registry image. The following examples describe how repo-tags are interpreted:

- `nginx:1.19` - Refers to an official Docker Hub image.
- `bluedata/mysql:1.0` - Refers to an image within the `bluedata` repository on Docker Hub.
- `quay.io/bitnami/nginx:latest` - Refers to an image on RedHat's `quay.io` registry under the `bitnami` repository.

Repositories can be public or private. It is not necessary to supply credentials to pull from a public repository; you must, however, supply credentials in order to pull from a private repository from an account with access to that repository. This is typically done via a `docker login` command. Acount access is always necessary to push to a repository. App Workbench therefore needs account access for that repository within the registry when building KubeDirector apps where an image is pushed to a repository,

There are two ways to provide a `docker login` context to App Workbench:

- **Option 1:** Either before or after the App Workbench container has been launched, but before building and pushing images:

  ```
  $ docker login <server>[:<port>]
  ```

  After the image has been built, you can logout:

  ```
  $ docker logout <server>[:<port>]
  ```

  Repo-tags used for the role images will be of the form:

  ```
  <server>[:<port>]/<myrepo>/<image>:<tag>
  ```

NOTE: You may only use one registry per application.

- **Option 2:** Supply credentials through environment variables. Be aware that this exposes them in plain text. Environment variables must be placed in a file called .env at the top level of the workspace directory. This must be done before launching the App Workbench container.

```
$ cat .env
        AWB_REGISTRY_USERNAME=<user>
        AWB_REGISTRY_PASSWORD=<password>
```

Launching App Workbench

A *workspace* is a directory that contains all of the files relevant for application development. The App Workbench web interface is launched and managed on a per-workspace basis. All of the lifecycle commands on this page should be executed from the workspace directory.

This article contains the following sections:

- Lifecycle describes the high-level App Workbench lifecycle.
- Launching the App Workbench Interface provides the detailed procedure for launching and then accessing the web interface.
- Stopping and Relaunching describes how to stop and then relaunch the interface.

The diagram at the bottom of this article illustrates these processes as they may appear on your workstation.

Lifecycle

App Workbench runs as a container service. The basic lifecycle is:

1. Change directory into your workspace by executing the following command:

   ```
   cd <path_to_workspace>
   ```

2. Launch the App Workbench container service by executing the following command:

   ```
   bdwb --launchui --port <port_#>
   ```

   (If no port is specified, the default is `5002`.)

   An App Workbench interface sessions starts for the current workspace.

3. Launch a web browser to design or edit your application. Data will be saved to your workspace.

4. If necessary, log in to the Docker registry that you will use to pull or push images during the application build process:

   ```
   docker login <registry_url>
   ```

5. After the application has been built, stop the container or workspace session, execute the following command from the workspace from which the session was started:

   ```
   bdwb --stopui
   ```


last-updated: May 06, 2024

6. If necessary, log out of the Docker registry:

```
docker logout <registry_url>
```

The following sections of this article describe each step of the lifecycle in detail.

Launching the App Workbench Interface

To launch the App Workbench interface:

1. Create a new workspace directory:
2. `mkdir <workspace>`
3. Switch to that directory:

```
cd <workspace>
```

4. Launch the interface:

```
bdwb --launchui --port <port_#>
```

(If no port is specified, the default is `5002`.)

5. Launch a web browser, and then navigate to the URL that indicated via the line:

```
HPE Workbench WebUI is running: open your browser to http://<host>:<port>
```

The web interface **Application Status** screen appears (see The Application Status Screen). You may now begin building your application.

Stopping & Relaunching

- To stop the App Workbench interface, execute the following command in the same workspace directory from which the interface was started:

```
bdwb --stopui
```

If you are not in the correct directory when you execute the command, an error similar to the following is displayed:

```
HPE Workbench WebUI is not running for the current workspace.
```

- To relaunch App Workbench, execute the following command in the existing workspace directory:

```
bdwb --launchui -port <port_#>
```

(If no port is specified, the default is `5002`.)

Lifecycle Example

This diagram illustrates the App Workbench lifecycle described above:

```
[sampleuser@prod21 AWB5.1]$ sudo cp bdwb /usr/bin/bdwb
[sampleuser@prod21 AWB5.1]$ sudo chmod +x /usr/bin/bdwb
[sampleuser@prod21 AWB5.1]$ mkdir ../workspace
[sampleuser@prod21 AWB5.1]$ cd ../workspace
[sampleuser@prod21 AWB5.1]$ bdwb --launchui --port 8080
b6c976476a0fed2e4ad54a2b3c1d3c8b23b299f0a23ddb9cda11fbb8c0e08f5c
HPE Application Workbench WebUI is running: open your browser to http://
prod21.sds.local:8080
[sampleuser@prod21 AWB5.1]$ bdwb --stopui
b6c976476a0fed2e4ad54a2b3c1d3c8b23b299f0a23ddb9cda11fbb8c0e08f5c
HPE Application Workbench WebUI stopped.
[sampleuser@prod21 AWB5.1]$ bdwb --launchui --port 8080
d080e3ba3870a85201b3919836921dd1faf8ab7090f46d0e0876dd443af25ba2
HPE Application Workbench WebUI is running: open your browser to http://
prod21.sds.local:8080
[sampleuser@prod21 AWB5.1]$
```

The Application Status Screen

Launching the App Workbench web interface (see Launching App Workbench) opens the **Application Status** screen. The appearance of this screen varies, based on the workspace you are using.

- If this is the first time you are launching the web interface in a new workspace (or if you are launching the web interface in a blank workspace), then see First Launch.
- If you are launching the web interface in an existing workspace that already has one or more image(s), then see Existing Workspace.

First Launch

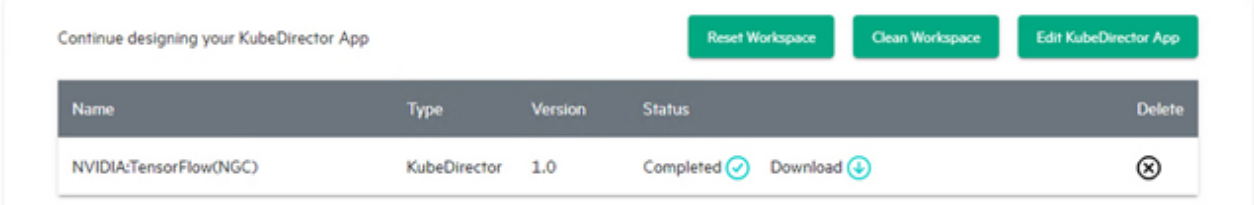
The **Application Status** screen appears as follows when you launch the web interface in a new or blank workspace.

This screen is blank because there is no application image data to show. You may begin creating a new application image by clicking the appropriate button:

- **To create a KubeDirector application:** Click the **Create a KubeDIrector App** button to open the **KubeDirector Application Details** screen. See The KubeDirector Application Details Screen.
- **To create a legacy EPIC application:** Click the **Create an EPIC App** button to open the **EPIC Application Details** screen. See The EPIC Application Details Screen.

Existing Workspace

The **Application Status** screen appears as follows when you launch the web interface in a workspace that contains an application in progress. You may only create or edit one application at a time, but you can create multiple builds or versions of that application:



The top of this screen contains the following three buttons:

- **Reset Workspace:** Clicking this button removes all application data, deliverables, and build artifacts from the workspace. See First Launch, above.
- **Clean Workspace:** Clicking this button removes application deliverables and build artifacts, but retains user-provided application data. See First Launch, above.
- **Edit App:** Clicking this button allows you to continue working on the application that appears in the table below the buttons. This button will say either:
  - **Edit KubeDirector App:** Opens the **KubeDirector Application Details** screen with the current app info loaded. See The KubeDirector Application Details Screen.
  - **Edit EPIC App:** Opens the **EPIC Application Details** screen with the current app info loaded. See The EPIC Application Details Screen.

The table on this screen displays the following information for each build deliverable of the application that you have created in the current workspace:

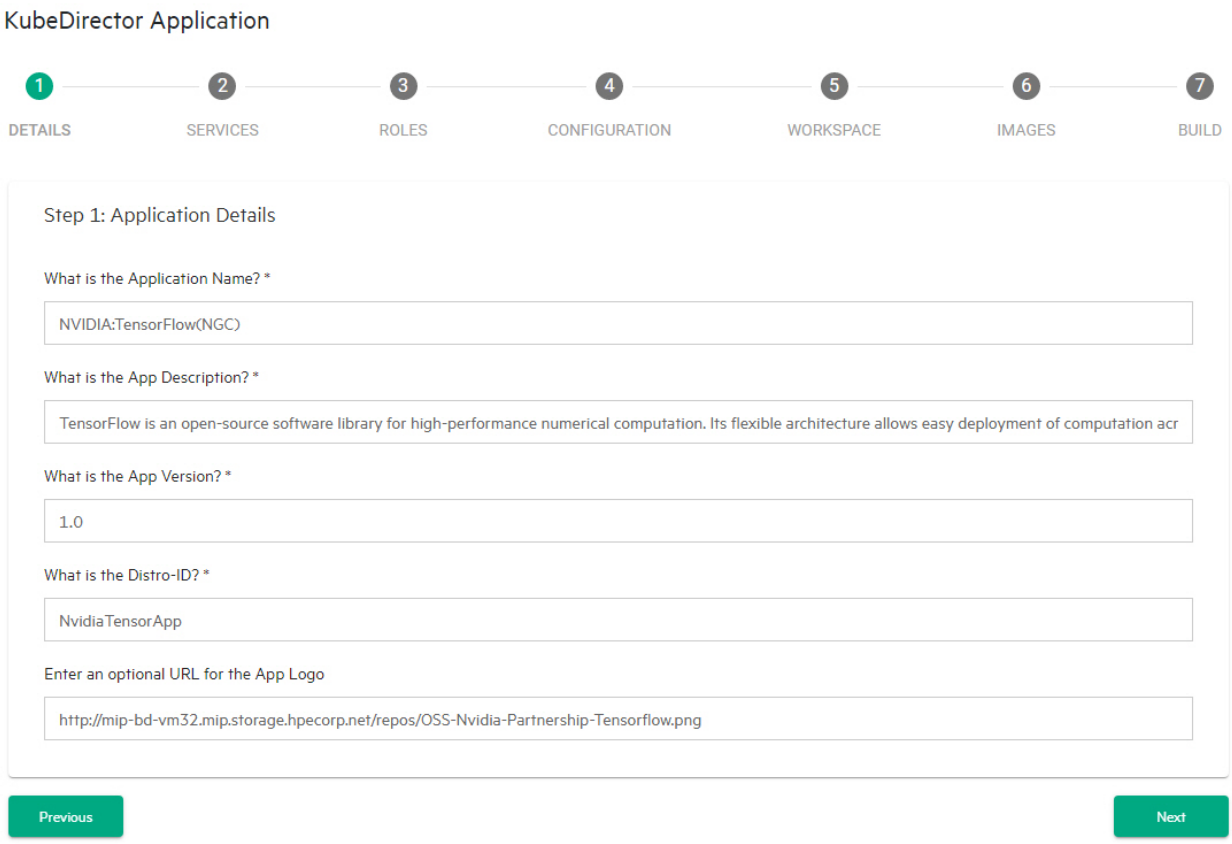
- **Name:** Name of the application build.
- **Type:** Type of application (either **KubeDirector** or **EPIC**).
- **Version:** Version of the application build deliverable.
- **Status:** Status of the application build deliverable. This will be one of the following:
  - **Failed:** The build failed. A short error message will be provided.
  - **Completed:** The build completed successfully.
- **Download:** Clicking the **Download** button (down arrow) downloads the selected application build.
- **Delete:** Clicking the **Delete** button (X) deletes the selected application build. You can still click the **Edit App** button to continue working on this application.

Building KubeDirector Apps

The KubeDirector Application Details Screen

In the **Application Status** screen (see The Application Status Screen):

- Clicking the **Create KubeDirector App** button opens a blank **KubeDirector Application Details** screen, which allows you to begin creating a new KubeDirector application. If you are creating an EPIC application, then please see The EPIC Application Details Screen.
- Clicking the **Edit KubeDirector App** buttons opens the **KubeDirector Application Details** screen, which allows you to edit the current application.

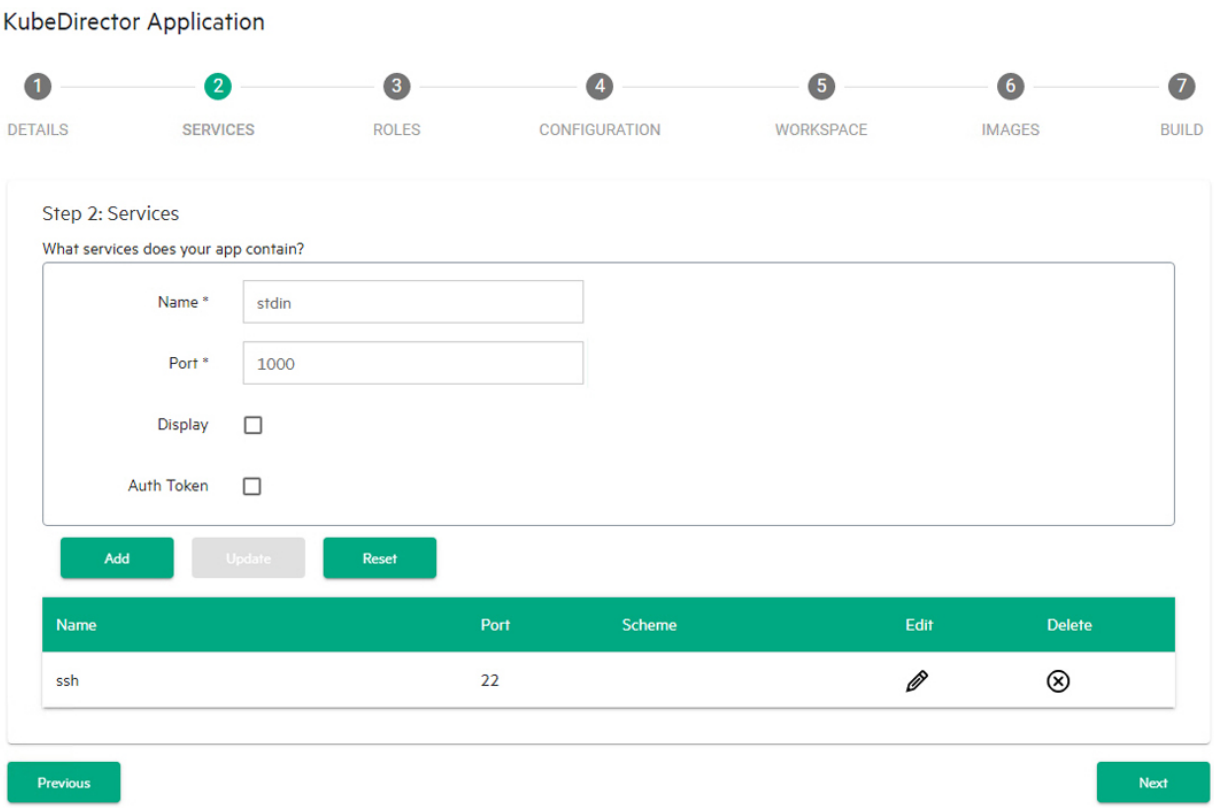


To provide application detail information:

1. Enter the name of the application that will appear in the HPE Ezmeral Runtime Enterprise **App Store** screens in the **What is the Application Name?** field.
2. Provide a short description of the application that will appear in the HPE Ezmeral Runtime Enterprise **App Store** screens in the **What is the App Description?** field.
3. Provide a unique version number for this application version in the **What is the App Version?** field.
4. If desired, provide the full path to a logo file (.jpg or .png) that will appear in the HPE Ezmeral Runtime Enterprise **App Store** screens in the **Enter an optional URL for the App Logo** field.
5. Verify this information, and then click **Next** to proceed to the **KubeDIrector Services** screen. See The KubeDIrector Services Screen.

The KubeDirector Services Screen

Clicking the **Next** button in the **KubeDirector Application Details** screen (see The KubeDirector Application Details Screen) opens the **KubeDirector Services** screen, which is where you describe the services that will be included in this application.



The **What services does your app contain**? section of the screen allows you to:

- **Add a new service:** See Adding a New Service.
- **View services:** See Viewing Services.
- **Edit an existing service:** See Editing an Existing Service.
- **Remove a service:** See Removing a Service.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT ALL LISTED SERVICES AS CONFIGURED ON THIS SCREEN.

When you have finished defining the services for your application, click **Next** to proceed to the **KubeDIrector Roles** screen. See The KubeDIrector Roles Screen.

Adding a New Service

To add a new service:

1. Enter a name for the service in the **Name** field.
2. Enter the port number this service will use in the **Port** field.
3. If the service is accessible via a web interface, then then check the **Display** checkbox to display the link to this service in the HPE Ezmeral Container Platform **Service Endpoints** tab. Otherwise, leave this checkbox blank.
4. If the service includes a web interface, then enter either `http` or `https` in the **Scheme** field, depending on whether or not the service requires secure access.

5. If the service includes a web interface, the enter the default path to the service in the **Path** field. This can be either `/` or a custom path, such as `/ui`.
6. If the service endpoint uses an authentication token, then check the **Auth Token** checkbox. Otherwise, leave it blank.
7. Click the **Add** button to add the new service.

You may now:

- View the service, as described in Viewing Services.
- Edit the service, as described in Editing an Existing Service.
- Remove the service, as described in Removing a Service.

Viewing Services

The table at the bottom of the **KubeDIrector Services** screen appears when you have defined at least one service for this application. This table displays the following information for each service:

- **Name:** Name of the service.
- **Port:** Port used by this service.
- **Scheme:** This will be either `http` or `https`, if the service has a web interface.
- **Edit:** Clicking the **Edit** icon (pencil) for a service allows you to edit that service. See Editing an Existing Service.
- **Delete:** Clicking the **Delete** icon (X) for a service removes that service. See Removing a Service.

Editing an Existing Service

To edit an existing service:

1. In the table at the bottom of the **KubeDirector Services** screen, click the **Edit** icon (pencil) for the service you want to edit.

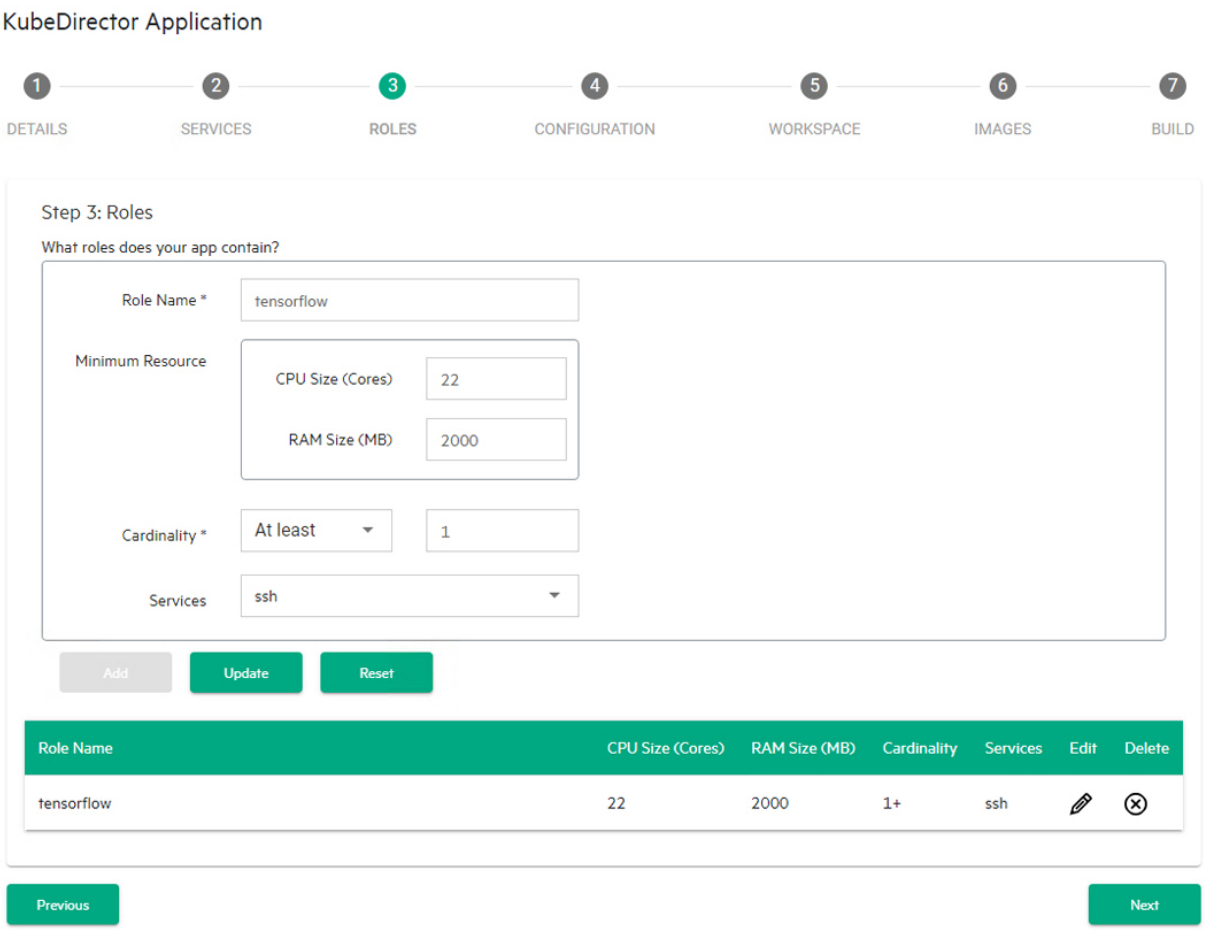
   The top of this section populates with the current information for the selected service.
2. Make your desired changes. See Adding a New Service for information on what to place in the fields.
3. Either:
   - Click **Add** to save your changes as a new service.
   - Click **Update** to save your changes to the existing service.
   - Click **Reset** to cancel your changes without modifying the service.

Removing a Service

To remove a service, click the **Delete** icon (X) for the service you want to remove in the table at the bottom of the **KubeDIrector Services** screen.

The KubeDirector Roles Screen

Clicking the **Next** button in the **KubeDirector Services** screen (see The KubeDirector Services Screen) opens the **KubeDirector Roles** screen, which is where you describe the pod roles that will be included in this application and assign services to those roles.



The **What roles does your app contain**? section of the screen allows you to:

- **Add a new role:** See Adding a New Role.
- **View roles:** See Viewing Roles.
- **Edit an existing role:** See Editing an Existing Role.
- **Remove a role:** See Removing a Role.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT ALL LISTED ROLES AS CONFIGURED ON THIS SCREEN AND THAT EACH ROLE WILL SUPPORT ALL SERVICE(S) ATTACHED TO THAT ROLE.

When you have finished defining the services for your application, click **Next** to proceed to the **KubeDIrector Configuration** screen. See The KubeDIrector Configuration Screen.

Adding a New Role

To add a new role:

1. Enter the name of the role in the **Role Name** field. This will specify the name of the virtual node/container/pod.
2. If desired, enter the minimum number of virtual CPU cores to use for this role in the **CPU Size (Cores)** field.
3. If desired, enter the minimum amount of RAM in MB to use for this role in the **RAM Size (MB)** field.

4. Enter the number of virtual nodes to be deployed for this role in the **Cardinality** field.
   - If you enter an integer, then a fixed number of virtual nodes. For example, entering 2 means that two virtual nodes of this role will be deployed.
   - If you enter an integer followed by a plus sign (+), then the integer specifies the minimum number of virtual nodes that will be deployed with this role. You may scale this out when deploying clusters/pods. For example, entering 2+ means that at least two virtual nodes of this role will be deployed; you may deploy a larger number if you choose.
5. Expand the **Services** menu to display all of the services that you configured in the **KubeDirector Roles** screen. You may now:
   - Check a checkbox for a service to assign that service to this role.
   - Clear a checkbox for a service to unassign that service from this role.
6. Click the **Add** button to add the new role.

You may now:

- View the role, as described in Viewing Roles.
- Edit the role, as described in Editing an Existing Role.
- Remove the role, as described in Removing a Role.

Viewing Roles

The table at the bottom of the **KubeDirector Roles** screen appears when you have defined at least one role for this application. This table displays the following information for each role:

- **Role Name:** Name of the role, which specifies the name of the virtual node/container/pod.
- **CPU Size (Cores):** If specified, the minimum number of virtual CPU cores to use for this role.
- **Ram Size (MB):** If specified, the minimum amount of RAM in MB to use for this role.
- **Cardinality:** Number of virtual nodes to deploy for this role. This can be either an absolute number (if the cardinality is an integer) or a minimum (if the cardinality is an integer followed by a plus sign (+)).
- **Services:** Any service(s) assigned to that role.
- **Edit:** Clicking the **Edit** icon (pencil) for a service allows you to edit that role. See Editing an Existing Role.
- **Delete:** Clicking the **Delete** icon (X) for a service removes that role. See Removing a Role.

Editing an Existing Role

To edit an existing role:

1. In the table at the bottom of the **KubeDirector Roles** screen, click the **Edit** icon (pencil) for the role you want to edit.

   The top of this section populates with the current information for the selected role.
2. Make your desired changes. See Adding a New Role for information on what to place in the fields.
3. Either:
   - Click **Add** to save your changes as a new role.

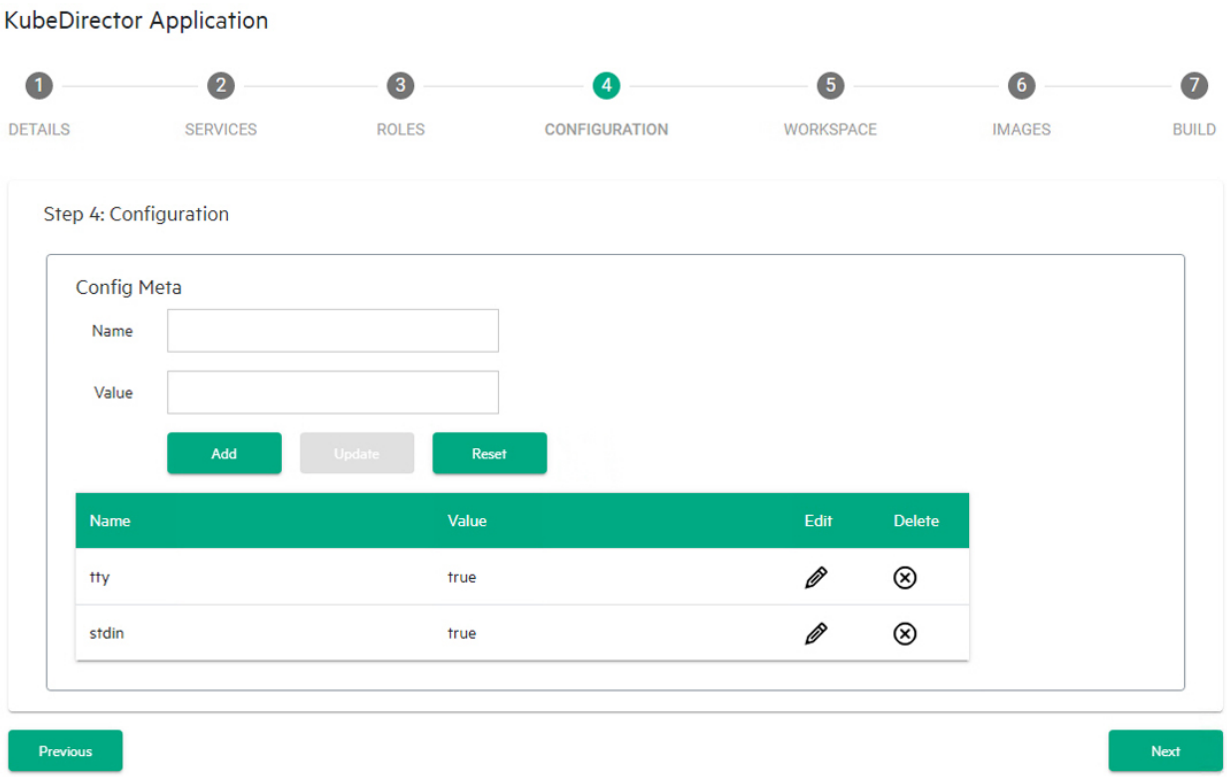
- Click **Update** to save your changes to the existing role.
- Click **Reset** to cancel your changes without modifying the role.

Removing a Role

To remove a role, click the **Delete** icon (X) for the role you want to remove in the table at the bottom of the **KubeDirector Roles** screen.

The KubeDirector Configuration Screen

Clicking the **Next** button in the **KubeDirector Roles** screen (see The KubeDirector Roles Screen) opens the **KubeDirector Configuration** screen, which is where you define key/value pairs that are used during application startup.



This screen allows you to:

- **Add a key/value pair:** See Adding a New Key.
- **View key/value pairs:** See Viewing Keys.
- **Edit an existing key/value pair:** See Editing an Existing Key.
- **Remove a key/value pair:** See Removing a Key.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT ALL LISTED KEY/VALUE PAIRS AS CONFIGURED ON THIS SCREEN.

When you have finished defining keys and values for your application, click **Next** to proceed to the **KubeDIrector Workspace** screen. See The KubeDIrector Workspace Screen.

Adding a New Key

To add a new key/value pair:

1. Enter the name of the key in the **Name** field.
2. Enter the value to assign to the key in the **Value** field.
3. Click the **Add** button to add the new key/value pair.

You may now:

- View the key/value pair, as described in Viewing Keys.
- Edit the key/value pair, as described in Editing an Existing Key.
- Remove the key/value pair, as described in Removing a Key.

Viewing Keys

The table at the bottom of the **Configuration** screen appears when you have defined at least one key/value pair for this application. This table displays the following information for each key/value pair:

- **Name:** Name of the key.
- **Value:** Value assigned to the key.
- **Edit:** Clicking the **Edit** icon (pencil) for a service allows you to edit that key/value pair. See Editing an Existing Key.
- **Delete:** Clicking the **Delete** icon (X) for a service removes that key/value pair. See Removing a Key.

Editing an Existing Key

To edit an existing key/value pair:

1. In the table at the bottom of the **Configuration** screen, click the **Edit** icon (pencil) for the key/value pair you want to edit.

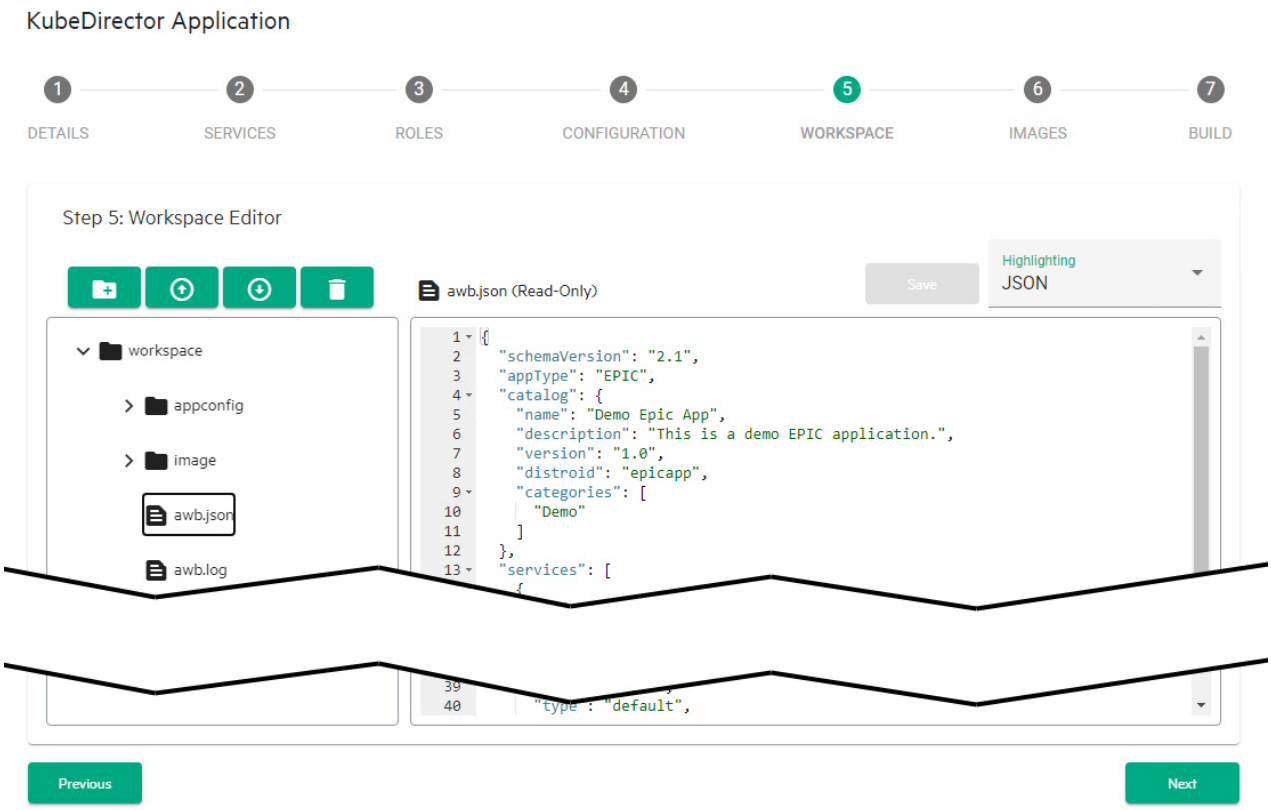
   The top of this section populates with the current information for the selected key/value pair.
2. Make your desired changes. See Adding a New Key for information on what to place in the fields.
3. Either:
   - Click **Add** to save your changes as a new key/value pair.
   - Click **Update** to save your changes to the existing key/value pair.
   - Click **Reset** to cancel your changes without modifying the key/value pair.

Removing a Key

To remove a key/value pair, click the **Delete** icon (X) for the key/value pair you want to remove in the table at the bottom of the **Configuration** screen.

The KubeDirector Workspace Screen

Clicking the **Next** button in the **KubeDirector Configuration** screen (see The KubeDirector Configuration Screen) opens the **KubeDirector Workspace** screen, which is where you can build a Docker file from scratch, and/or add or edit scripts.



The top left of this screen contains the following four buttons

- **Add a directory:** See Adding a Directory.
- **Upload a file:** See Uploading a File.
- **Download a file:** See Downloading a File.
- **Delete:** See Deleting Files and Directories.

The left side of this screen beneath the buttons displays the current directory tree and contents.

- Click a collapsed directory to expand and view its contents.
- Click an expanded directory to collapse and hide its contents.
- Click a file to view its contents on the right side of the screen.

The right side of the screen contains a text editor that populates with the contents of a file when you select that file in the directory tree.

- The **Highlighting** pull-down menu automatically selects a text highlighting schema based on the detected syntax (e.g. JSON, Markdown, or Python). You can override this setting by selecting a different schema using this menu.
- The filename appears above the file contents. If the notation **(Read-Only)** does not appear, then you may edit this file using the script editor. See Editing a File.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT THE DIRECTORIES AND FILES THAT YOU HAVE CREATED.

last-updated: May 06, 2024

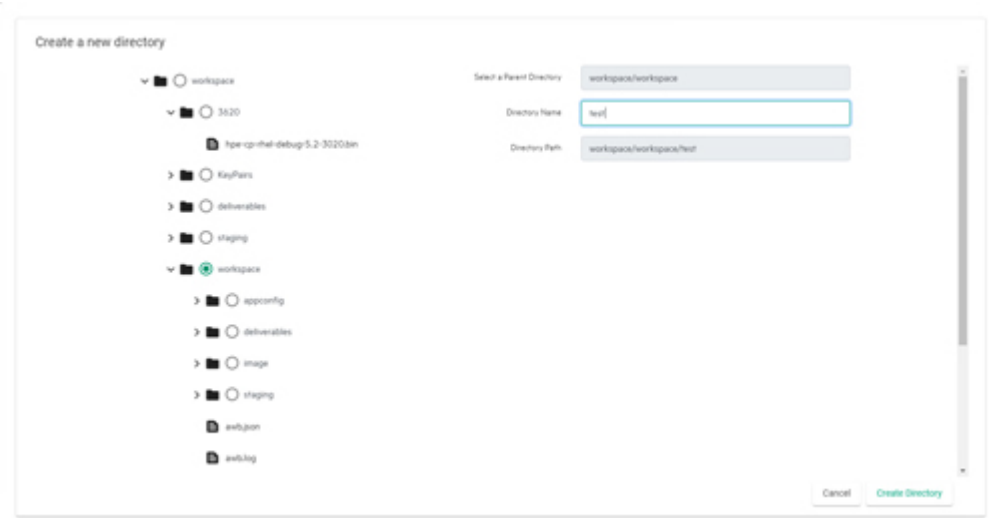
When you have finished setting up the workspace for your application, click **Next** to proceed to the **KubeDIrector Images** screen. See The KubeDIrector Images Screen.

Adding a Directory

To add a new directory:

1. Click the **Add Directory** button (folder with a + sign).

   The **Create a new directory** popup appears.



2. Check the radio button of the parent directory under which you want to create the new directory.

   The **Select a Parent Directory** and **Directory Path** fields populate. These fields are read-only.

3. Enter the name for the new directory in the **Directory Name** field.

4. Click the **Create Directory** button.

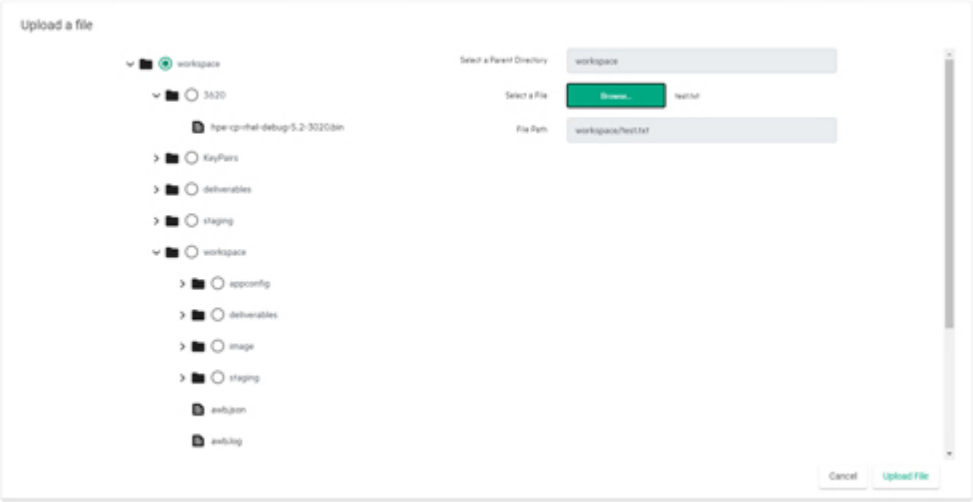
The popup closes and returns you to the **KubeDirector Workspace** screen, and the new directory appears in the directory tree on the left side of the screen.

Uploading a File

To upload a file ot the workspace:

1. Click the **Upload File** button (circle with an up arrow).

   The **Upload a file** popup appears.



2. Check the radio button that corresponds to the directory to where you want to upload the file.

   The **Select a Parent Directory** field populates. This field is read-only.

3. Click the **Browse** button to open a standard **Open** dialog, and then navigate to and select the file you want to upload.

   The **File Path** field populates. This field is read-only.

4. Click the **Upload File** button.

CAUTION: UPLOADING A DUPLICATE FILE OVERWRITES THAT FILE WITH THE NEWLY-UPLOADED VERSION. ANY EDITS YOU MADE TO THAT FILE IN THIS SCREEN WILL BE LOST.

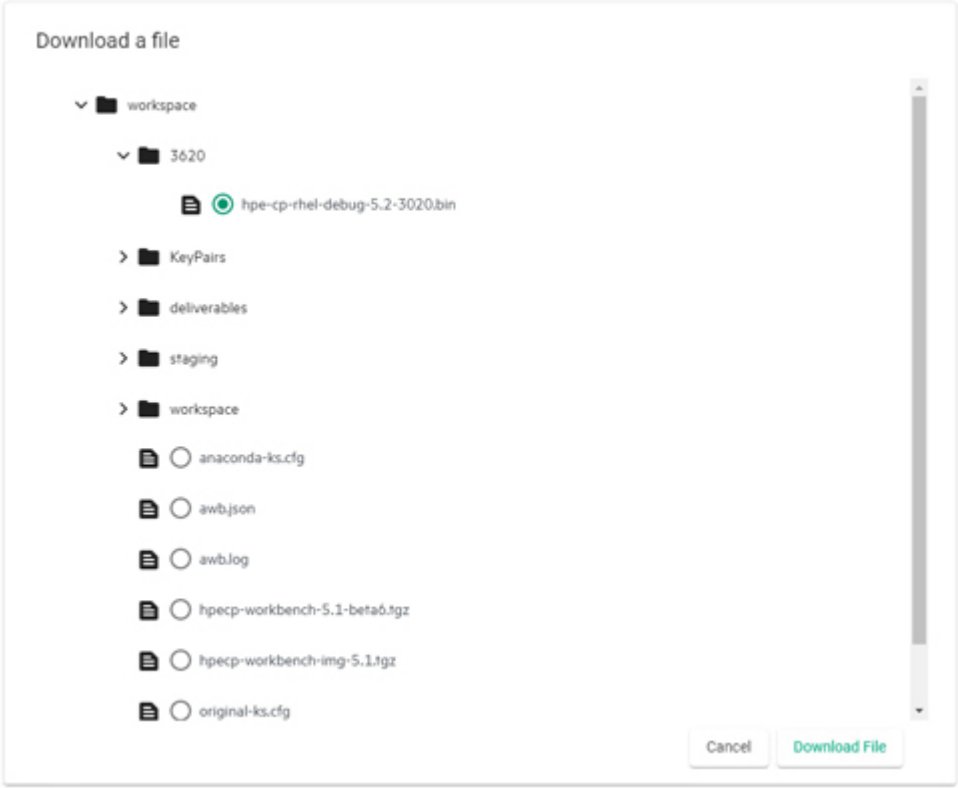
The popup closes and returns you to the **KubeDirector Workspace** screen, and the new file appears in the directory tree on the left side of the screen.

Downloading a File

To download a file:

1. Click the **Download File** button (circle with a down arrow).

   The **Download a file** pop-up appears.



2. Check the radio button that corresponds to the file you want to download.
3. Click the **Download** button.

The file downloads to your local computer. The download location and behavior will vary based on your browser configuration.

Editing Files

To edit a file:

1. Select the file you want to edit in the directory tree.

   The contents of that file automatically appear in the text editor on the right side of the screen.
2. If desired, you may use the **Highlight** pull-down menu to change the text highlighting schema. This only affects how the text is displayed; it does not modify the file in any way.
3. Make your desired edits directly in the file.

NOTE: You cannot edit a file if the notation **(Read-Only)** appears above the text editor.

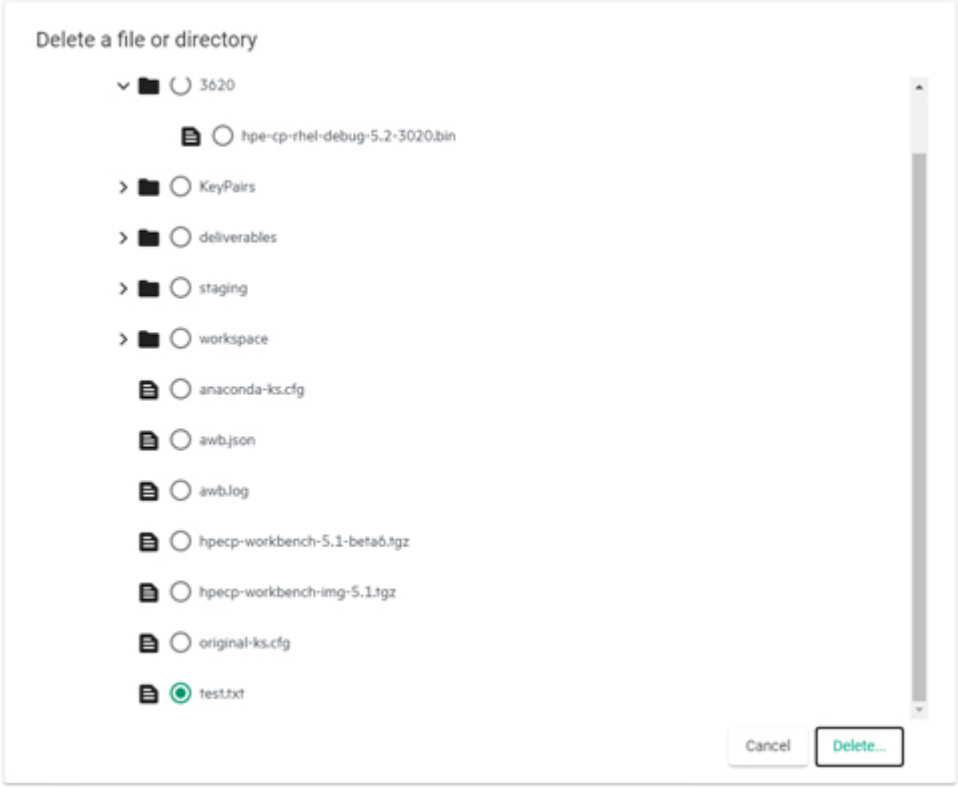
When you have completed your edits, click the **Save** button.

Deleting Files and Directories

To delete a file or directory:

1. Click the **Delete** button (trash can).

   The **Delete a file or directory** popup appears.



2. Check the radio button that corresponds to the file or directory you want to delete.

   CAUTION: DELETING A DIRECTORY REMOVES ALL OF THE CONTENTS (SUBDIRECTORIES AND FILES) OF THE DELETED DIRECTORY.

3. Click the **Delete** button.

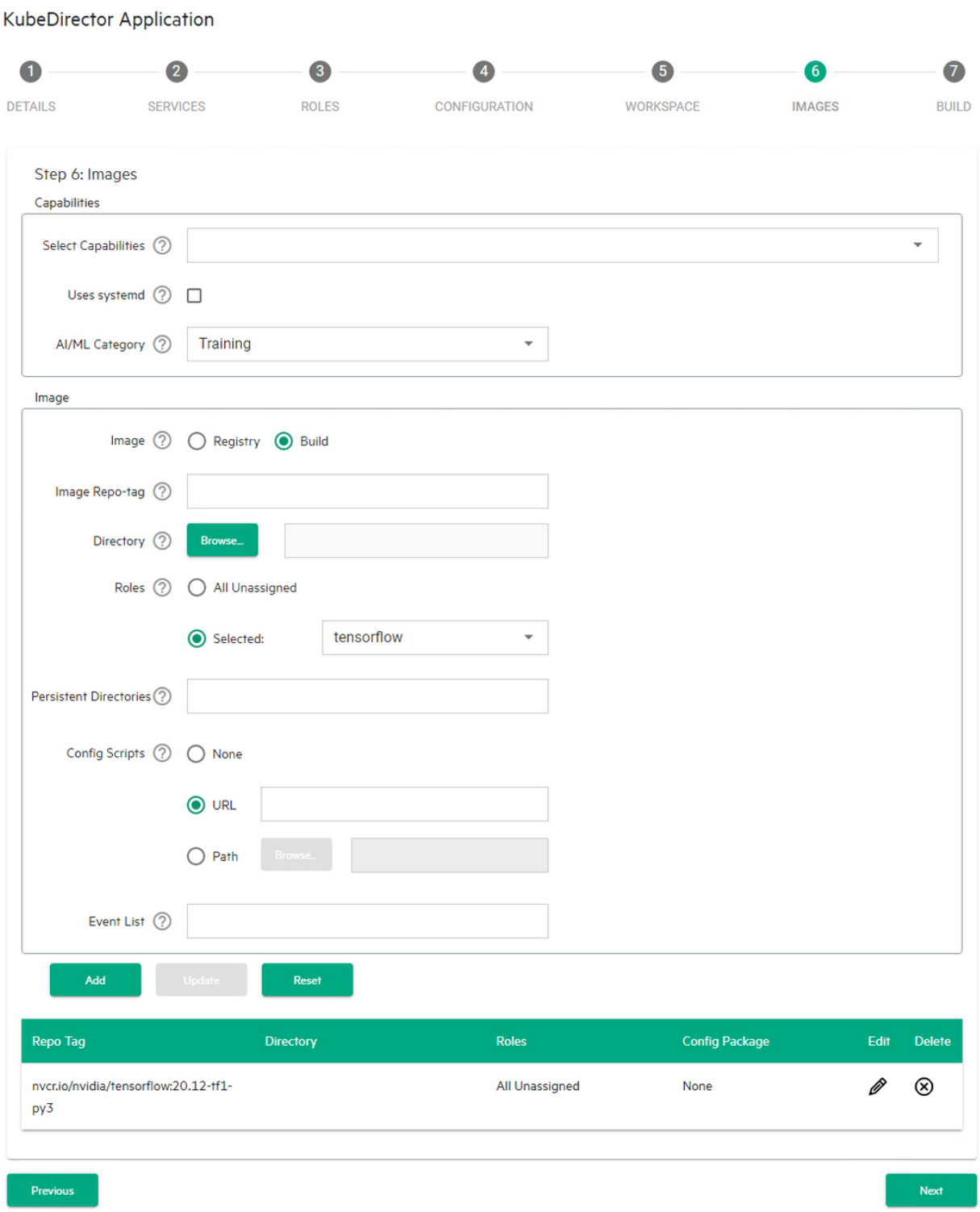
   A confirmation dialog appears.

4. Click **Confirm** to finish the deletion.

CAUTION: YOU CANNOT UNDELETE A DELETED FILE OR DIRECTORY.

The KubeDirector Images Screen

Clicking the **Next** button in the **KubeDirector Workspace** screen (see The KubeDirector Workspace Screen) opens the **KubeDirector Images** screen, which is where you assign container images to application roles.



This screen allows you to:

- **Add a new container image:** See Adding a New Image.
- **View container images:** See Viewing Images.
- **Edit an existing image:** See Editing an Existing Image.

- **Remove an image:** See Removing an Image.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT IMAGES ARE PROPERLY SOURCED AND MATCHED TO ROLES.

When you have finished mapping container images to roles, click **Next** to proceed to the **KubeDIrector Build** screen. See The KubeDIrector Build Screen.

Adding a New Image

To add a new container image and map that image to a role:

1. Use the **Select Capabilities** pull-down menu to expand a list of Linux capabilities, and then check the checkbox(es) that correspond to the Linux capability or capabilities you need this application to have.
2. If the application uses `systemd` to control its services, then check the **Uses systemd** checkbox.
3. If the application is an AI/ML application, then use the **AI/ML Category** pull-down menu to select the Kubernetes Project **App Store** screen tab where the application will appear in the HPE Ezmeral Runtime Enterprise web interface. For example, if you select **Training**, then the application will appear in the **Training** tab.
4. Check the appropriate Image radio button to determine the location of a source container image to use for the application you are creating.
   - **Registry:** Checking this radio button means that the source container image is stored in a registry, such as `docker.io`.
   - **Build:** Checking this radio button means that the source container image will be sourced locally. Selecting this option exposes the **Directory** field and **Browse** button. Either use the **Browse** button to navigate to the location of the source image, or enter the complete path in the field.
5. Provide the image repository information in the **Image Repo-tag** field, in the format:

   ```
   <repository_url>/<repository>/<name>:<tag>
   ```

   .
6. Use the **Roles** radio buttons to select the role(s) for which this container image applies.
   - **All Unassigned:** Checking this radio button assigns this container image to all roles that do not have another image specified.
   - **Selected:** Checking this radio button and then selecting one or more role(s) using the pull-down menu assigns this container image to the specified role(s). Checking a checkbox next to a role name assigns the image to that role; clearing a checkbox unassigns that image from the role.
7. Select any needed config scripts by checking the appropriate radio button:
   - **None:** No config script is needed.
   - **URL:** If the config script is available online, then check this radio button and then enter the complete URL to the application configuration script in the field.
   - **Path:** If the configuration script is available locally, then check this radio button, and then use the **Browse** button and/or field to specify the directory where the script is located.
8. Enter the lifecycle event(s) for that KubeDirector should invoke for this role in the **Event List** field.
9. Click **Add** to finish adding the image.

You may now:

- View the image/role mapping, as described in Viewing Images.
- Edit the image/role mapping, as described in Editing an Existing Image.
- Remove the image/role, as described in Removing an Image.

Viewing Images

The table at the bottom of the **KubeDirector Images** screen appears when you have defined at least one image/role mapping for this application. This table displays the following information for each image/role mapping:

- **Repo Tag:** Repo tag of the image/role mapping.
- **Directory:** If the container image is sourced locally, this is the path to that image.
- **Roles:** Role(s) to which this container image has been mapped.
- **Config Package:** Any config script defined for this image/role mapping.
- **Edit:** Clicking the **Edit** icon (pencil) for a service allows you to edit that role. See Editing an Existing Role.
- **Delete:** Clicking the **Delete** icon (X) for a service removes that role. See Removing a Role.

Editing an Existing Image

To edit an existing image/role mapping:

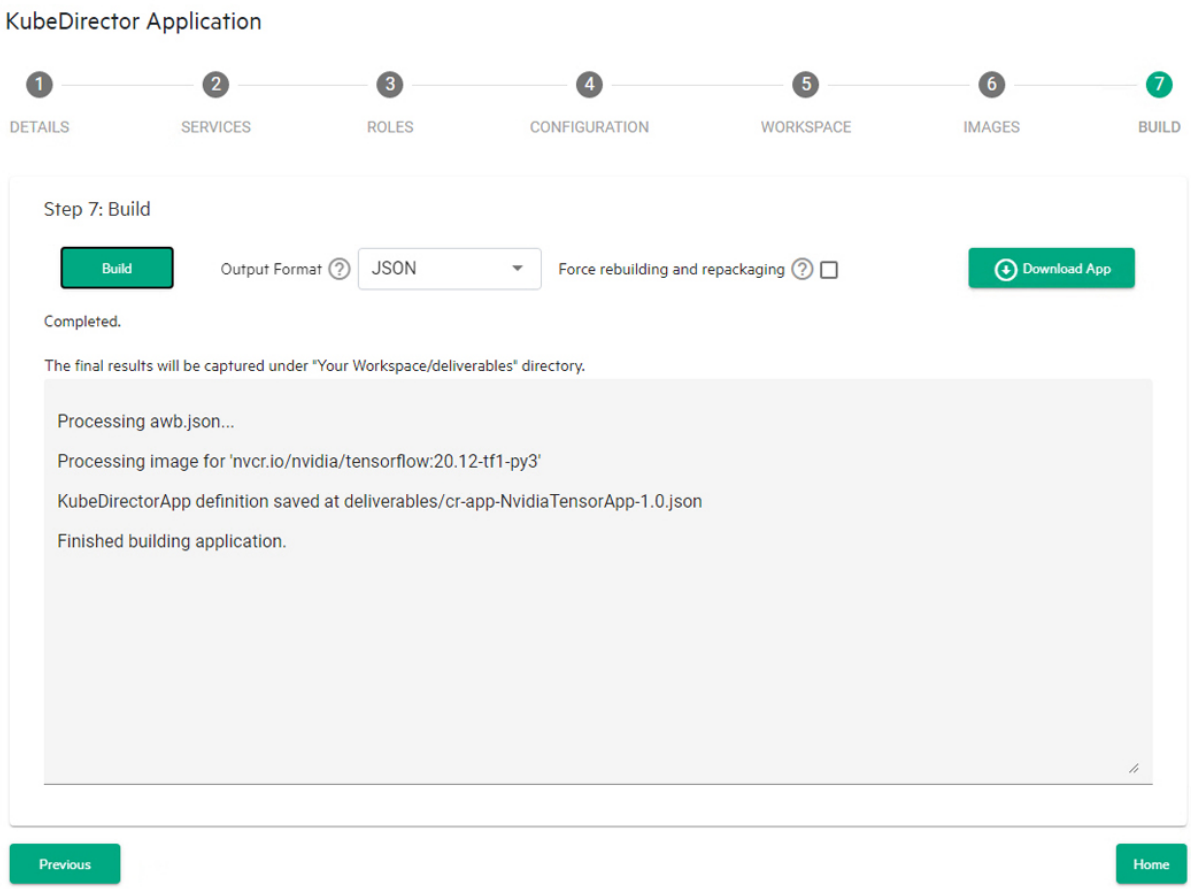
1. In the table at the bottom of the **KubeDirector Images** screen, click the **Edit** icon (pencil) for the image/role mapping you want to edit.
2. The top of this section populates with the current information for the selected image/role mapping.
3. Make your desired changes. See Adding a New Image for information on what to place in the fields.
4. Either:
   - Click **Add** to save your changes as a new image/role mapping.
   - Click **Update** to save your changes to the existing image/role mapping.
   - Click **Reset** to cancel your changes without modifying the image/role mapping.

Removing an Image

To remove an image/role mapping, click the **Delete** icon (X) for the mapping you want to remove in the table at the bottom of the **KubeDirector Images** screen.

The KubeDirector Build Screen

Clicking the **Next** button in the **KubeDirector Images** screen (see The KubeDirector Images Screen) opens the **Build** screen, which is where you can build the application.



To build your application:

1. Use the Output Format pull-down menu to specify the application build format (JSON or YAML).
2. If you want to force the build process to rebuild and repackage the application even if the image or package already exists, then check the Fore rebuilding and repacking checkbox.
3. Click the **Build** button. Clicking this button starts the application deliverable build process using the format specified in the **Output Format** pull-down menu. The text area in the center of this screen displays the application build logs. Once the application completes successfully, the deliverable file will be located in the `<workspace>/deliverables/` directory. For example:

```
                         [sampleuser@prod21 AWB5.1]# ls
                         appconfig awb20210206.log awb.json bdwb deliverables
documentation.md image logo.png staging status.json
                         [sampleuser@prod21 AWB5.1]# cd deliverables/
                         cr-app-NvidiaTensorApp-1.0.json
                         [sampleuser@prod21 AWB5.1]#
```

You may now:

- Download the application by clicking the **Download App** button.
- Manually add this application to HPE Ezmeral Runtime Enterprise. The new application appears in the **App Store** screen.

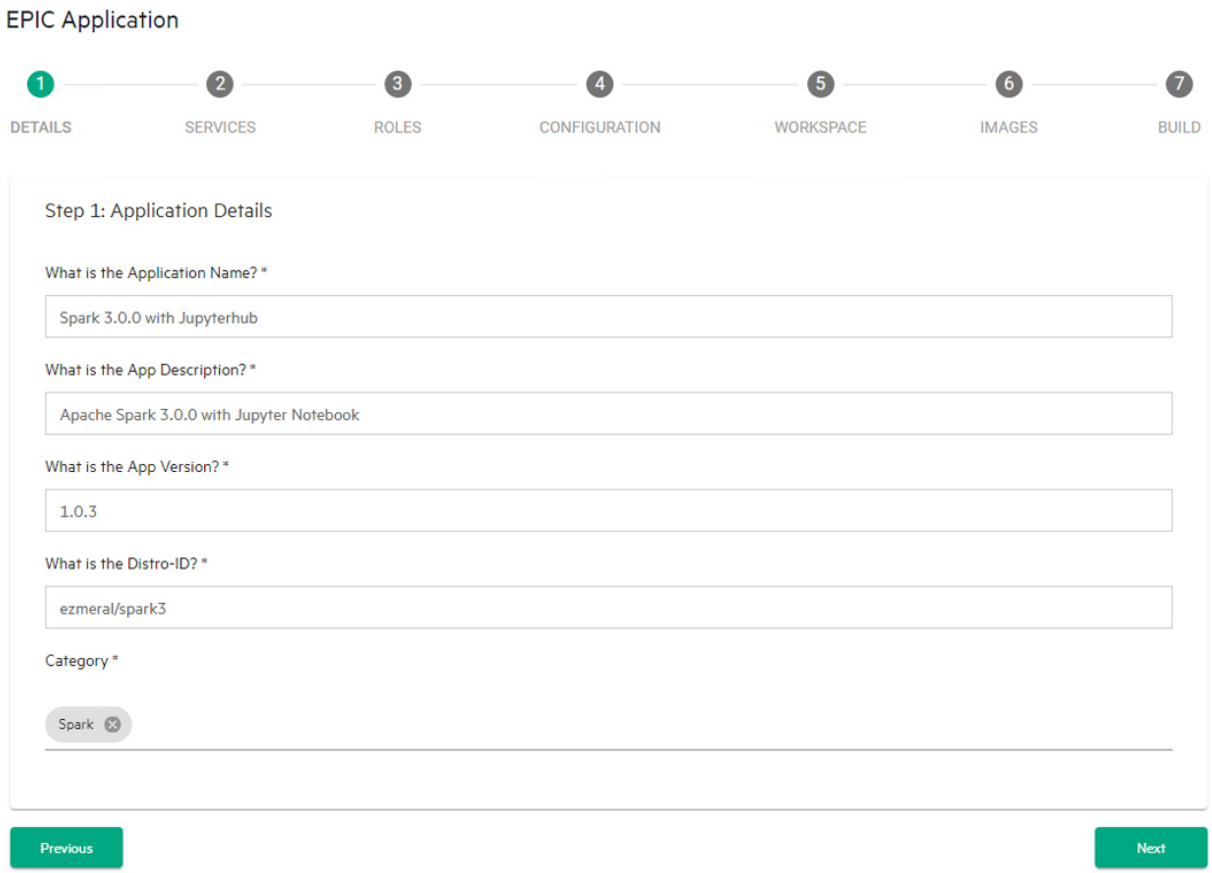
Once the build process has completed, you may click **Home** to return to the the **Application Status** screen. See The Application Status Screen.

Building EPIC Applications

The EPIC Application Details Screen

In the **Application Status** screen (see The Application Status Screen):

- Clicking the **Create EPIC App** button opens a blank **EPIC Application Details** screen, which allows you to begin creating a new EPIC application. If you are creating a KubeDIrector application, then please see The KubeDirector Application Details Screen.
- Clicking the **Edit EPIC App** buttons opens the **EPIC Application Details** screen, which allows you to edit the current application.



To provide application detail information:

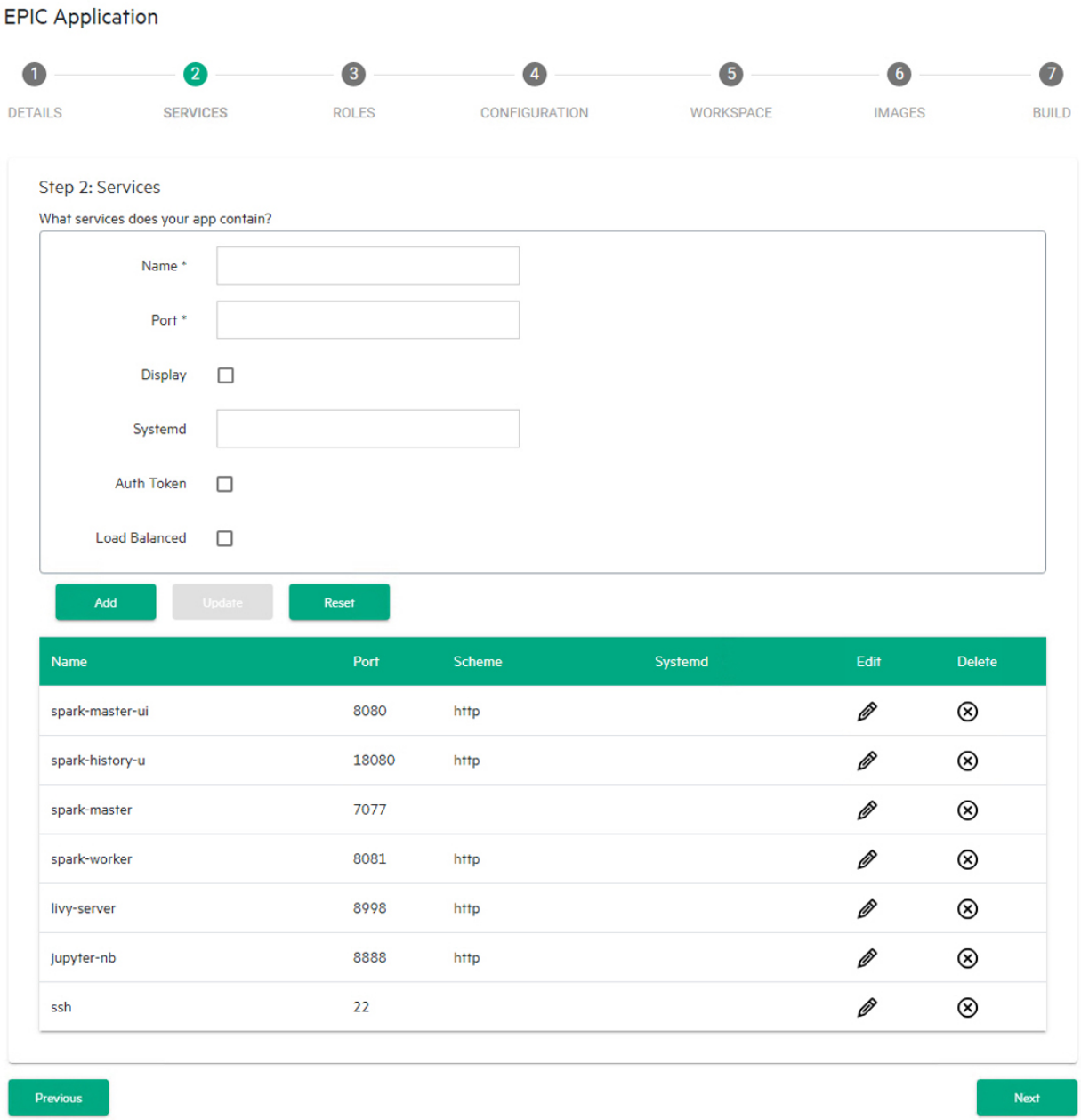
1. Enter the name of the application that will appear in the HPE Ezmeral Runtime Enterprise **App Store** screens in the **What is the Application Name?** field.
2. Provide a short description of the application that will appear in the HPE Ezmeral Runtime Enterprise **App Store** screens in the **What is the App Description?** field.
3. Provide a unique version number for this application version in the **What is the App Version?** field.
4. Select one or more category(ies) for this application:
5. To add a category, double-click anywhere in the **Category** area, and then select an available category. The added category appears in a gray bubble.

6. To remove a category, click the **Delete** icon (X) for the category you want to remove.

7. Verify this information, and then click **Next** to proceed to the **EPIC Services** screen. See The EPIC Services Screen.

The EPIC Services Screen

Clicking the **Next** button in the **EPIC Application Details** screen (see The EPIC Application Details Screen) opens the **EPIC Services** screen, which is where you describe the services that will be included in this application.



The **What services does your app contain**? section of the screen allows you to:

- **Add a new service:** See Adding a New Service.
- **View services:** See Viewing Services.

- **Edit an existing service:** See Editing an Existing Service.
- **Remove a service:** See Removing a Service.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT ALL LISTED SERVICES AS CONFIGURED ON THIS SCREEN.

When you have finished defining the services for your application, click **Next** to proceed to the **EPIC Roles** screen. See The EPIC Roles Screen.

Adding a New Service

To add a new service:

1. Enter a name for the service in the **Name** field.
2. Enter the port number this service will use in the **Port** field.
3. If the service is accessible via a web interface, then then check the **Display** checkbox to display the link to this service in the HPE Ezmeral Runtime Enterprise **Service Endpoints** tab. Otherwise, leave this checkbox blank.
4. If the service include a web interface, then enter either `http` or `https` in the **Scheme** field, depending on whether or not the service requires secure access.
5. If the service include a web interface, then enter the default path to the service in the **Path** field. This can be either `/` or a custom path, such as `/ui`.
6. If the application uses `systemd` to manage its resources, then enter the unit name of the service in the **Systemd** field.
7. If the service endpoint uses an authentication token, then check the **Auth Token** checkbox. Otherwise, leave it blank.
8. If the service needs to be load-balanced, then check the **Load Balanced** checkbox. Otherwise, leave it blank.
9. Click the **Add** button to add the new service.

You may now:

- View the service, as described in Viewing Services.
- Edit the service, as described in Editing an Existing Service.
- Remove the service, as described in Removing a Service.

Viewing Services

The table at the bottom of the **EPIC Services** screen appears when you have defined at least one service for this application. This table displays the following information for each service:

- **Name:** Name of the service.
- **Port:** Port used by this service.
- **Scheme:** This will be either `http` or `https`, if the service has a web interface.
- **Systemd:** Unit name of the `systemd` service, if the application uses `systemd` to manage its resources

- **Edit:** Clicking the **Edit** icon (pencil) for a service allows you to edit that service. See Editing an Existing Service.
- **Delete:** Clicking the **Delete** icon (X) for a service removes that service. See Removing a Service.

Editing an Existing Service

To edit an existing service:

1. In the table at the bottom of the **EPIC Services** screen, click the **Edit** icon (pencil) for the service you want to edit.

   The top of this section populates with the current information for the selected service.

2. Make your desired changes. See Adding a New Service for information on what to place in the fields.
3. Either:
   - Click **Add** to save your changes as a new service.
   - Click **Update** to save your changes to the existing service.
   - Click **Reset** to cancel your changes without modifying the service.

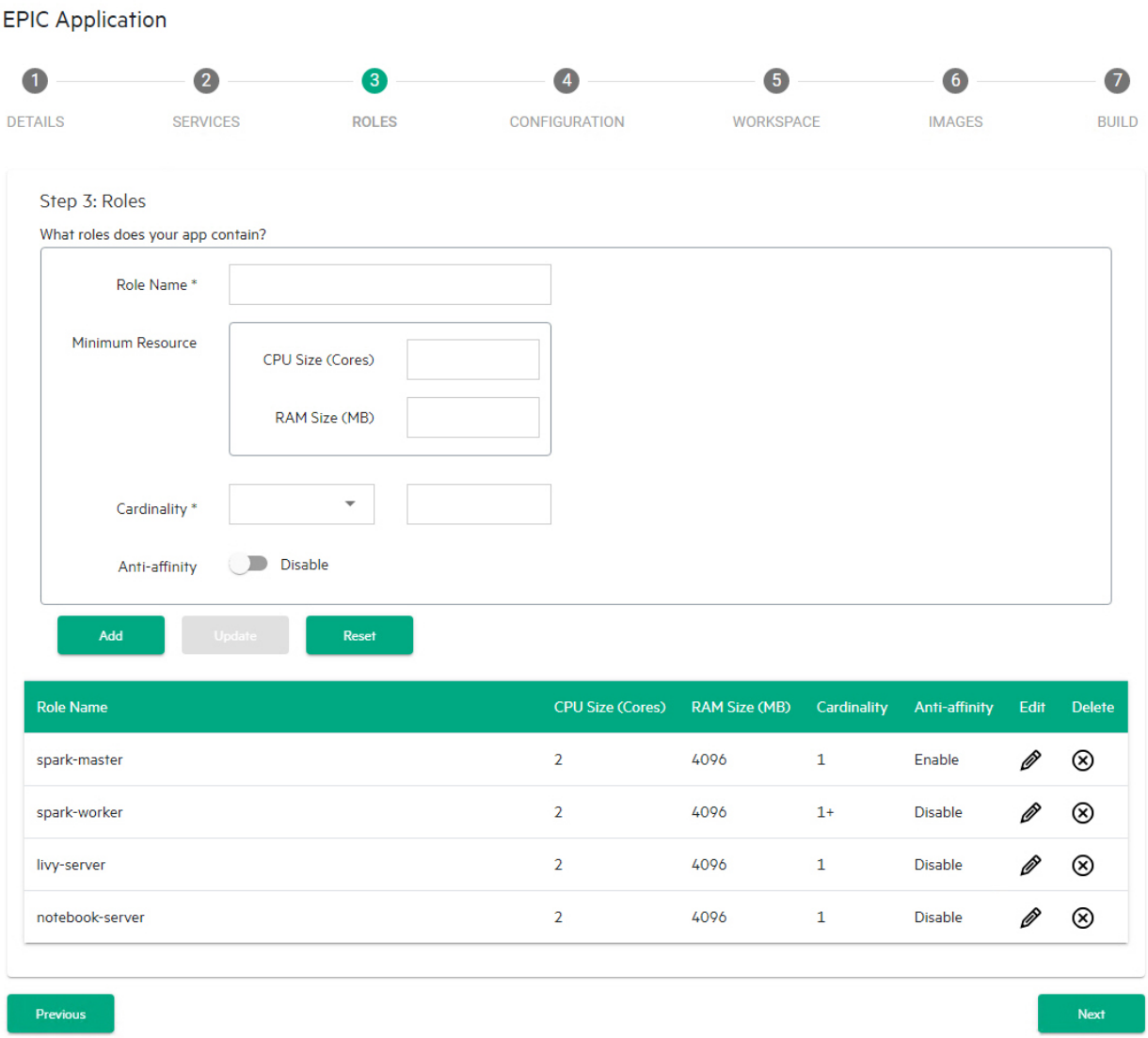
Removing a Service

To remove a service, click the **Delete** icon (X) for the service you want to remove in the table at the bottom of the **EPIC Services** screen.

The EPIC Roles Screen

Clicking the **Next** button in the **EPIC Services** screen (see The EPIC Services Screen) opens the **EPIC Roles** screen, which is where you describe the pod roles that will be included in this application and assign services to those roles.

last-updated: May 06, 2024



The **What roles does your app contain**? section of the screen allows you to:

- **Add a new role:** See Adding a New Role.
- **View roles:** See Viewing Roles.
- **Edit an existing role:** See Editing an Existing Role.
- **Remove a role:** See Removing a Role.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT ALL LISTED ROLES AS CONFIGURED ON THIS SCREEN AND THAT EACH ROLE WILL SUPPORT ALL SERVICE(S) ATTACHED TO THAT ROLE.

When you have finished defining the services for your application, click **Next** to proceed to the **EPIC Configuration** screen. See The EPIC Configuration Screen.

Adding a New Role

To add a new role:

1. Enter the name of the role in the **Role Name** field. This will specify the name of the virtual node/container/pod.
2. If desired, enter the minimum number of virtual CPU cores to use for this role in the **CPU Size (Cores)** field.
3. If desired, enter the minimum amount of RAM in MB to use for this role in the **RAM Size (MB)** field.
4. Enter the number of virtual nodes to be deployed for this role in the **Cardinality** field.
   - If you enter an integer, then a fixed number of virtual nodes. For example, entering 2 means that two virtual nodes of this role will be deployed.
   - If you enter an integer followed by a plus sign (+), then the integer specifies the minimum number of virtual nodes that will be deployed with this role. You may scale this out when deploying clusters/pods. For example, entering 2+ means that at least two virtual nodes of this role will be deployed; you may deploy a larger number if you choose.
5. By default, anti-affinity physically separates each virtual node with this role from its peers, thereby lessening the odds that a fault affecting one virtual node will affect the other virtual node(s). This feature also reduces the physical resources used by virtual nodes. To disable this feature, slide the **Anti-affinity** switch to the **Disabled** position.
6. Click the **Add** button to add the new role.

You may now:

- View the role, as described in Viewing Roles.
- Edit the role, as described in Editing an Existing Role.
- Remove the role, as described in Removing a Role.

Viewing Roles

The table at the bottom of the **EPIC Roles** screen appears when you have defined at least one role for this application. This table displays the following information for each role:

- **Role Name:** Name of the role, which specifies the name of the virtual node/container/pod.
- **CPU Size (Cores):** If specified, the minimum number of virtual CPU cores to use for this role.
- **Ram Size (MB):** If specified, the minimum amount of RAM in MB to use for this role.
- **Cardinality:** Number of virtual nodes to deploy for this role. This can be either an absolute number (if the cardinality is an integer) or a minimum (if the cardinality is an integer followed by a plus sign (+)).
- **Anti-affinitity:** Whether (**Enable**) or not (**Disable**) anti-affinity is active for this role.
- **Edit:** Clicking the **Edit** icon (pencil) for a service allows you to edit that role. See Editing an Existing Role.
- **Delete:** Clicking the **Delete** icon (X) for a service removes that role. See Removing a Role.

Editing an Existing Role

To edit an existing role:

1. In the table at the bottom of the **EPIC Roles** screen, click the **Edit** icon (pencil) for the role you want to edit.

   The top of this section populates with the current information for the selected role.

2. Make your desired changes. See Adding a New Role for information on what to place in the fields.

3. Either:

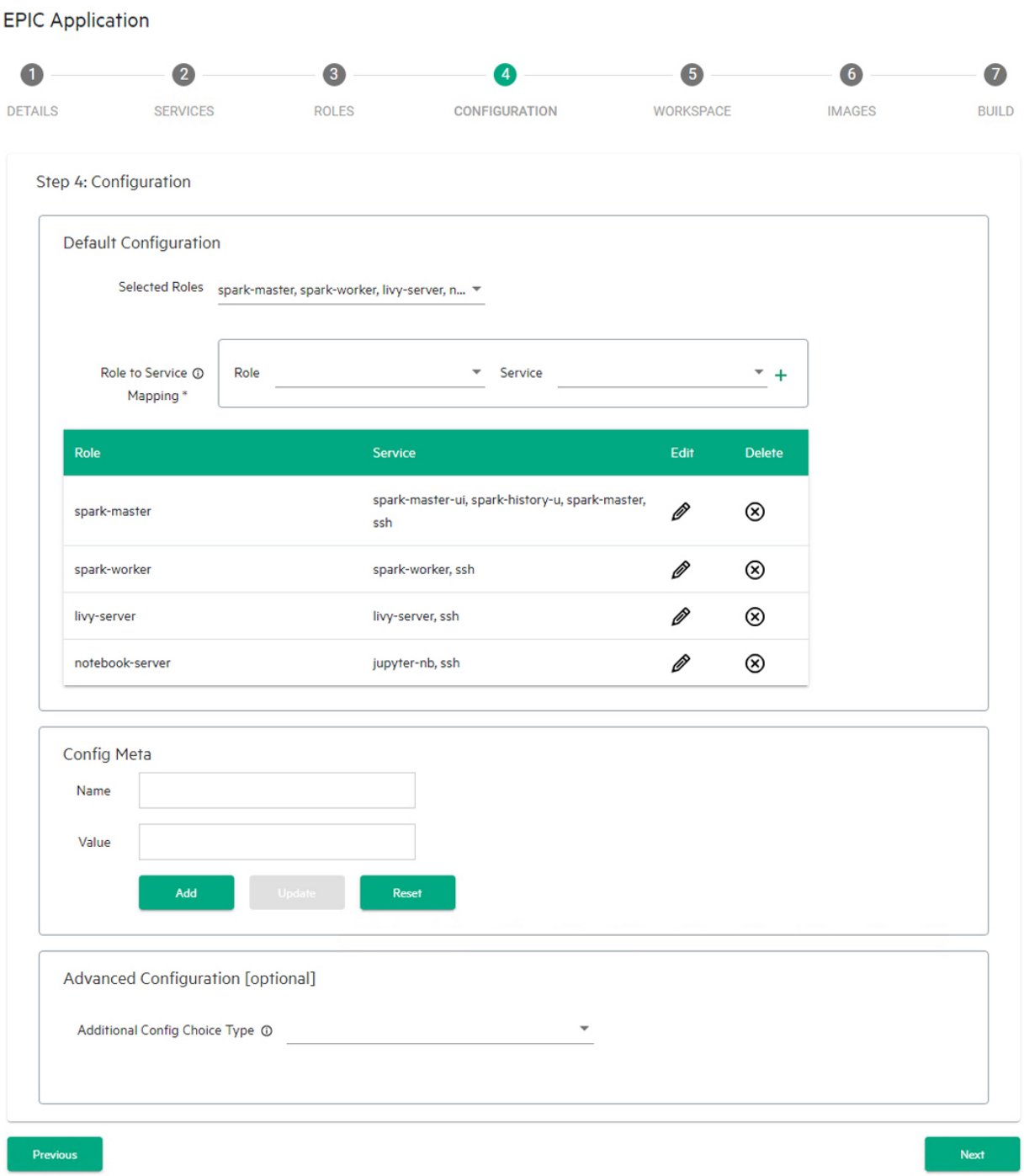
   - Click **Add** to save your changes as a new role.
   - Click **Update** to save your changes to the existing role.
   - Click **Reset** to cancel your changes without modifying the role.

Removing a Role

To remove a role, click the **Delete** icon (X) for the role you want to remove in the table at the bottom of the **EPIC Roles** screen.

The EPIC Configuration Screen

Clicking the **Next** button in the **EPIC Roles** screen (see The EPIC Roles Screen) opens the **EPIC Configuration** screen, which is where you define key/value pairs that are used during application startup.



This screen contains three sections:

- **Default Configuration**: Allows you to map services to roles. See Default Configuration.
- **Config Meta:** Allows you to specify key/value pairs that are used during application startup. See Config Meta.
- **Advanced Configuration:** Allows complex mappings and pairings. See Advanced Configuration. This optional section contains the following subsections:
  - **Boolean:** Allows conditional role-to-service mappings. See Boolean.

- **Multivalue:** Allows multiple role-to-service mappings. See Multivalue.
- **String:** Allows you to specify string inputs that will be collected when deploying the application. See String.
- **Password:** Allows you to specify a password input that will be collected when deploying the application. See Password.

To define configuration options:

1. Define role-to-service mappings, as described in Default Configuration.
2. Define key/value pairings, as described in Config Meta.
3. Optionally add any additional configurations, as described in Advanced Configurations and the applicable subsection(s) therein.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT ALL LISTED KEY/VALUE PAIRS AS CONFIGURED ON THIS SCREEN.

Verify this information, and then click **Next** to proceed to the **EPIC Workspace** screen. See The EPIC Workspace Screen.

Default Configuration

The **Default Configuration** section of the **EPIC Configuration** screen allows you to select the role(s) that are deployed with all application configurations and to map services to those roles. To do this:

1. Use the **Selected Roles** pull-down menu to select the default role(s). Press [CTRL] to make multiple selections.
2. Use the **Role** pull-down menu to select one of the default roles that you selected in Step 1.
3. Use the **Service** pull-down menu to select a service to map to role that you selected in Step 2.
4. Click the **Add** icon (+) to map another service to the selected role.

The table in this section lists each of the mappings you have configured and allows you to:

- Edit a mapping by clicking the **Edit** icon (pencil) for that mapping. You can then adjust the configuration as described above, and then click the **Add** icon to save your changes.
- Remove a mapping by clicking the **Delete** icon (X) for that mapping.

Config Meta

The **Config Meta** section of the **EPIC Configuration** screen allows you to specify key/value pairs that can be referenced by app configuration scripts. To do this:

1. Enter the name of the key as a static string in the **Name** field.
2. Enter the corresponding value as a static string in the **Value** field.
3. Click the **Add** button to add the key-to-value pairing. This information will be available for app configuration scripts to query.

The table in this section lists each of the pairings you have configured and allows you to:

- Edit a pairing by clicking the **Edit** icon (pencil) for that pairing. You can then adjust the configuration as described above, and then click the **Update** button to save your changes. If you want to cancel your edits, you may click the **Reset** button.
- Remove a pairing by clicking the **Delete** icon (X) for that mapping.

Advanced Configuration

The **Advanced Configuration** section of the **EPIC Configuration** screen allows you to optionally specify key/value pairs that can be referenced by app configuration scripts. To do this, use the **Additional Config Choice Type** pull-down menu to specify the type of configuration to add. The available options are:

- Boolean
- Multivalue
- String
- Password

Boolean

The **Boolean** subsection of the **Advanced Configuration** section allows you to map services to roles when a specified condition is met. To do this:

1. Enter a name for the condition that must be true in the **Name** field.
2. Use the **Additional Roles** pull-down menu to select the role(s) to be covered by this condition. Press [CTRL] to make multiple selections if needed.
3. Use the **Role** pull-down menu to select one of the roles that you selected in Step 2.
4. Use the **Service** pull-down menu to select a service to map to the role you selected in Step 3.
5. If desired, click the **Add** icon (+) to add map another service to this role.
6. Click the green **Add** button to add the mapping.

The table in this section lists each of the mappings you have configured and allows you to:

- Edit a mapping by clicking the **Edit** icon (pencil) for that mapping. You can then adjust the configuration as described above, and then click the **Update** button to save your changes.
- Remove a pairing by clicking the **Delete** icon (X) for that mapping.

Multivalue

The **Multivalue** subsection of the **Advanced Configuration** section allows you to map services to roles when a user specifies one or more option(s) when deploying the application. You may add a number of options, and then append an additional role to each option. These roles are added in addition to the default roles specified in the **Services and Roles** screen when the specified options are added. To do this:

1. Enter a name for the multivalue configuration in the **Name** field.
2. Add one or option(s) that will trigger this configuration in the **Options** field.
3. Use the **Additional Roles** pull-down menu to select the role(s) to be covered by this configuration. Press [CTRL] to make multiple selections if needed.
4. Use the **Role** pull-down menu to select one of the roles that you selected in Step 2.

5. Use the **Service** pull-down menu to select a service to map to the role you selected in Step 3.
6. If desired, click the **Add** icon (+) to add map another service to this role.

The first table in this section lists each of the mappings you have configured and allows you to:

- Edit a mapping by clicking the **Edit** icon (pencil) for that mapping. You can then adjust the configuration as described above, and then click the **Update** button to save your changes.
- Remove a mapping by clicking the **Delete** icon (X) for that mapping.

The second table in this section lists each of the configurations and options you have configured, along with the role(s) and service(s) configured for this option and allows you to:

- Edit a mapping by clicking the **Edit** icon (pencil) for that mapping. You can then adjust the configuration as described above, and then click the **Update** button to save your changes.
- Remove a mapping by clicking the **Delete** icon (X) for that mapping.

String

The **String** subsection of the **Advanced Configuration** section allows you to create one or more string input(s) that be collected from users when deploying the application. To do this:

1. Enter a name for the string in the **Name** field.
2. Click the **Add** button to add the string input.

The table in this section lists each of the string inputs you have configured and allows you to:

- Edit a string by clicking the **Edit** icon (pencil) for that string. You can then adjust the string as described above, and then click the **Update** button to save your changes.
- Remove a string by clicking the **Delete** icon (X) for that string.

Password

The **Password** subsection of the **Advanced Configuration** section allows you to create one or more password input(s) that be collected from users when deploying the application. To do this:

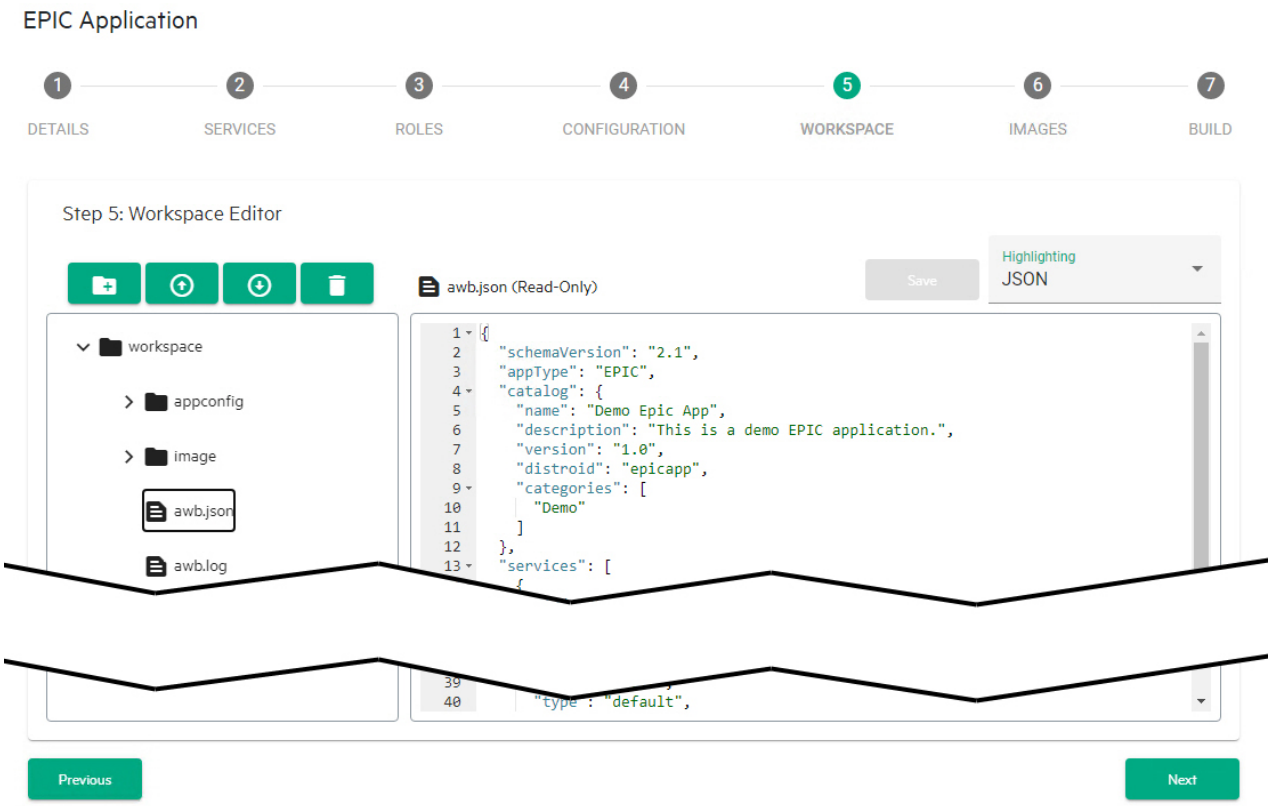
1. Enter a name for the password in the **Name** field.
2. Click the **Add** button to add the password input.

The table in this section lists each of the password inputs you have configured and allows you to:

- Edit a string by clicking the **Edit** icon (pencil) for that password. You can then adjust the password as described above, and then click the **Update** button to save your changes.
- Remove a password by clicking the **Delete** icon (X) for that password.

The EPIC Workspace Screen

Clicking the **Next** button in the **EPIC Configuration** screen (see The EPIC Configuration Screen) opens the **EPIC Workspace** screen, which is where you can build a Docker file from scratch, and/or add or edit scripts.



The top left of this screen contains the following four buttons

- **Add a directory:** See Adding a Directory.
- **Upload a file:** See Uploading a File.
- **Download a file:** See Downloading a File.
- **Delete:** See Deleting Files and Directories.

The left side of this screen beneath the buttons displays the current directory tree and contents.

- Click a collapsed directory to expand and view its contents.
- Click an expanded directory to collapse and hide its contents.
- Click a file to view its contents on the right side of the screen.

The right side of the screen contains a text editor that populates with the contents of a file when you select that file in the directory tree.

- The **Highlighting** pull-down menu automatically selects a text highlighting schema based on the detected syntax (e.g. JSON, Markdown, or Python). You can override this setting by selecting a different schema using this menu.
- The filename appears above the file contents. If the notation **(Read-Only)** does not appear, then you may edit this file using the script editor. See Editing a File.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT YOUR APPLICATION WILL SUPPORT THE DIRECTORIES AND FILES THAT YOU HAVE CREATED.

last-updated: May 06, 2024

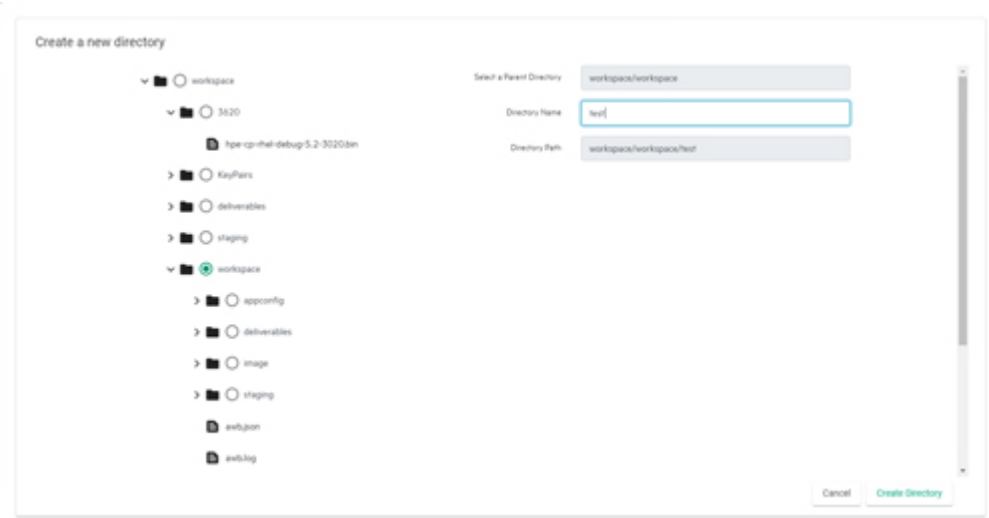
When you have finished setting up the workspace for your application, click **Next** to proceed to the **EPIC Images** screen. See The EPIC Images Screen.

Adding a Directory

To add a new directory:

1. Click the **Add Directory** button (folder with a + sign).

   The **Create a new directory** popup appears.



2. Check the radio button of the parent directory under which you want to create the new directory.

   The **Select a Parent Directory** and **Directory Path** fields populate. These fields are read-only.

3. Enter the name for the new directory in the **Directory Name** field.

4. Click the **Create Directory** button.

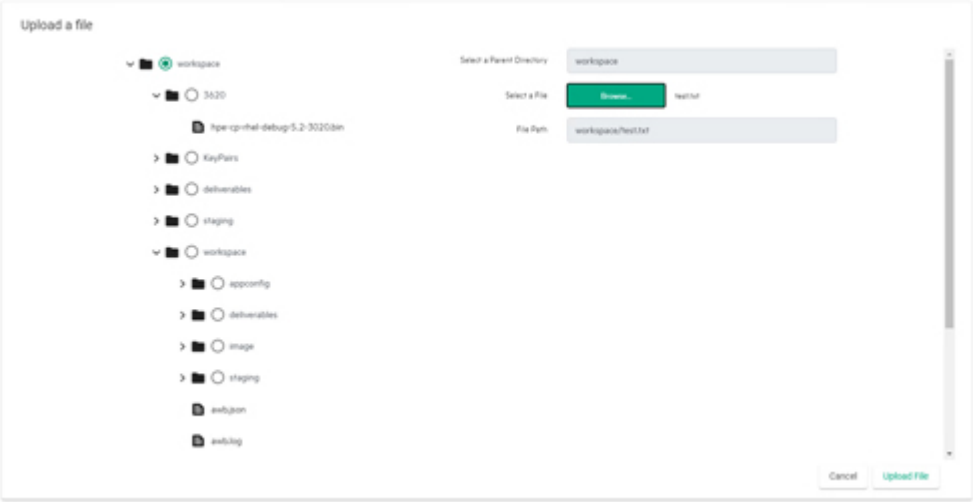
The popup closes and returns you to the **EPIC Workspace** screen, and the new directory appears in the directory tree on the left side of the screen.

Uploading a File

To upload a file ot the workspace:

1. Click the **Upload File** button (circle with an up arrow).

   The **Upload a file** popup appears.



2. Check the radio button that corresponds to the directory to where you want to upload the file.

   The **Select a Parent Directory** field populates. This field is read-only.

3. Click the **Browse** button to open a standard **Open** dialog, and then navigate to and select the file you want to upload.

   The **File Path** field populates. This field is read-only.

4. Click the **Upload File** button.

CAUTION: UPLOADING A DUPLICATE FILE OVERWRITES THAT FILE WITH THE NEWLY-UPLOADED VERSION. ANY EDITS YOU MADE TO THAT FILE IN THIS SCREEN WILL BE LOST.

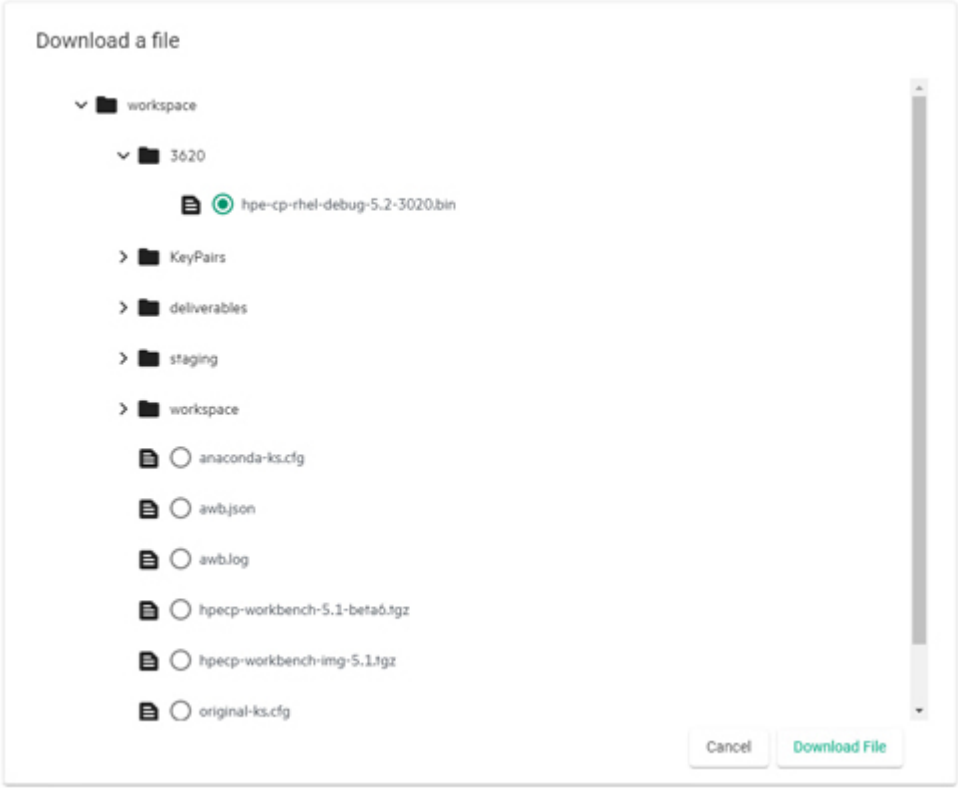
The popup closes and returns you to the **EPIC Workspace** screen, and the new file appears in the directory tree on the left side of the screen.

Downloading a File

To download a file:

1. Click the **Download File** button (circle with a down arrow).

   The **Download a file** pop-up appears.



2. Check the radio button that corresponds to the file you want to download.
3. Click the **Download** button.

The file downloads to your local computer. The download location and behavior will vary based on your browser configuration.

Editing Files

To edit a file:

1. Select the file you want to edit in the directory tree.

   The contents of that file automatically appear in the text editor on the right side of the screen.
2. If desired, you may use the **Highlight** pull-down menu to change the text highlighting schema. This only affects how the text is displayed; it does not modify the file in any way.
3. Make your desired edits directly in the file.

NOTE: You cannot edit a file if the notation **(Read-Only)** appears above the text editor.

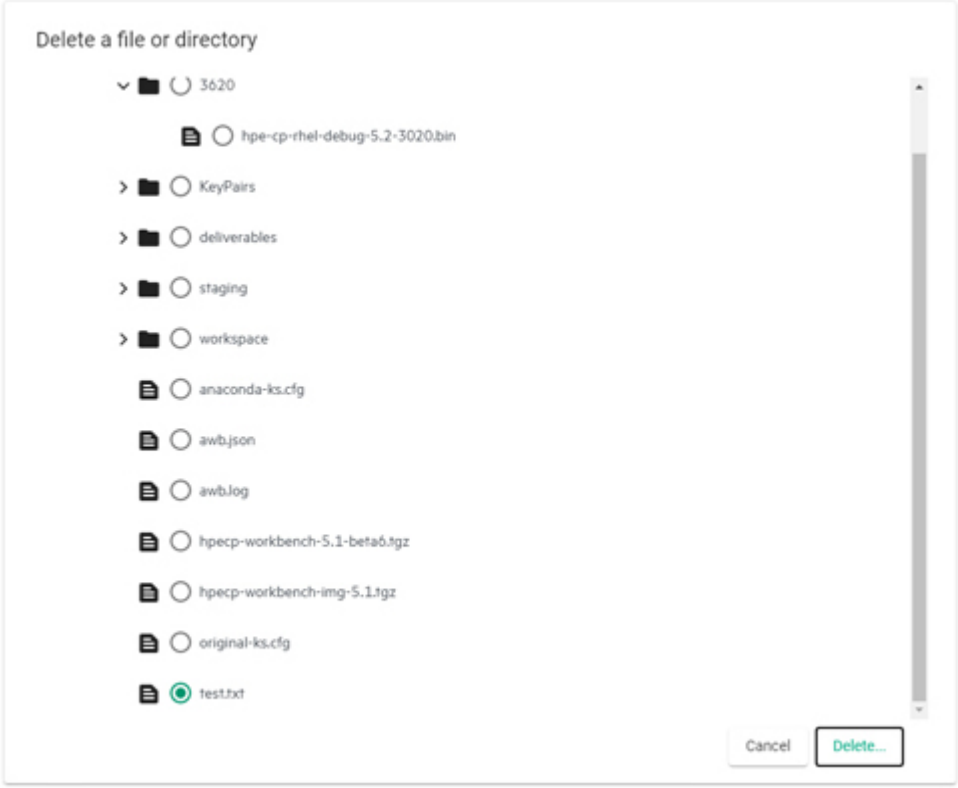
When you have completed your edits, click the **Save** button.

Deleting Files and Directories

To delete a file or directory:

1. Click the **Delete** button (trash can).

   The **Delete a file or directory** popup appears.



2. Check the radio button that corresponds to the file or directory you want to delete.

   CAUTION: DELETING A DIRECTORY REMOVES ALL OF THE CONTENTS (SUBDIRECTORIES AND FILES) OF THE DELETED DIRECTORY.

3. Click the **Delete** button.

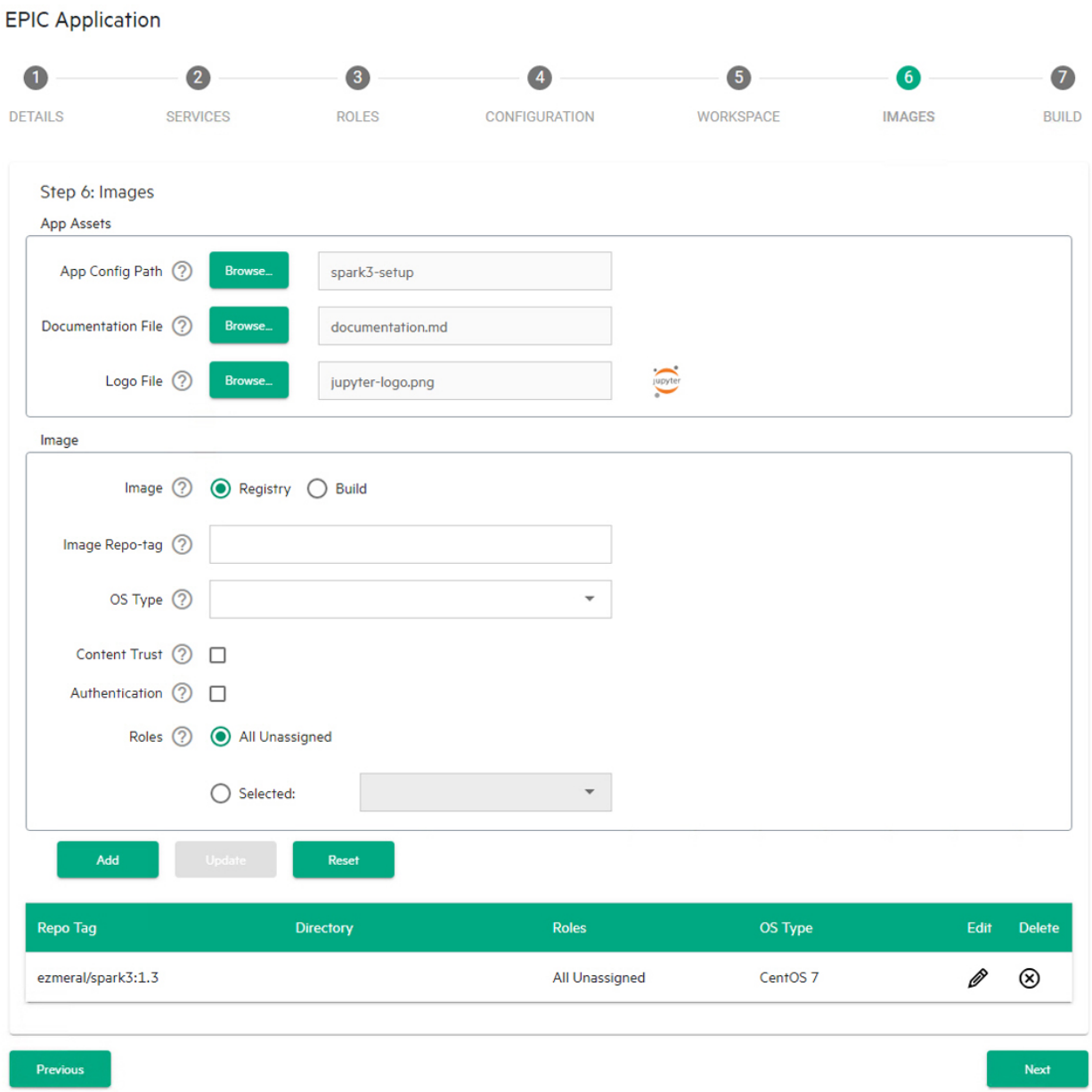
   A confirmation dialog appears.

4. Click **Confirm** to finish the deletion.

CAUTION: YOU CANNOT UNDELETE A DELETED FILE OR DIRECTORY.

The EPIC Images Screen

Clicking the **Next** button in the **EPIC Workspace** screen (see The EPIC Workspace Screen) opens the **EPIC Images** screen, which allows you to specify additional configuration options.



This screen allows you to:

- **Work with application assets:** See App Assets.
- **Add a new container image:** See Adding a New Image.
- **View container images:** See Viewing Images.
- **Edit an existing image:** See Editing an Existing Image.
- **Remove an image:** See Removing an Image.

CAUTION: APP WORKBENCH DOES NOT VALIDATE THIS INFORMATION. YOU MUST BE SURE THAT IMAGES ARE PROPERLY SOURCED AND MATCHED TO ROLES.

When you have finished mapping container images to roles, click **Next** to proceed to the **EPIC Build** screen. See The EPIC Build Screen.

App Assets

The App Assets section at the top of the screen allows you to manage the following application assets:

- **App Config Path:** Click this button to navigate to and select the directory within the workspace that contains all of the application scripts.
- **Documentation File:** Add any documentation you are including with this application in Markdown (.md) format.
- **Logo File:** You may to upload a logo image that will appear in the HPE Ezmeral Runtime Enterprise **App Store** screens. You may upload either:
  - **PNG:** 400x200 pixels.
  - **JPG:** File size must be equal to or less than 512 KB.

Adding a New Image

To add a new container image and map that image to a role:

1. Check the appropriate Image radio button to determine the location of a source container image to use for the application you are creating.
   - **Registry:** Checking this radio button means that the source container image is stored in a registry, such as `docker.io`.
   - **Build:** Checking this radio button means that the source container image will be sourced locally. Selecting this option exposes the **Directory** field and **Browse** button. Either use the **Browse** button to navigate to the location of the source image, or enter the complete path in the field.
2. Provide the image repository information in the **Image Repo-tag** field, in the format:

   ```
   <repository_url>/<repository>/<name>:<tag>
   ```

   .
3. Use the **OS Type** pull-down menu to specify the container image OS (**CentOS**, **RHEL**, or **Ubuntu**).
4. If content trust is enabled on the container image, then check the **Content Trust** checkbox.
5. If the container image registry requires authentication, then check the **Authentication** checkbox.
6. Use the **Roles** radio buttons to select the role(s) for which this container image applies.
   - **All Unassigned:** Checking this radio button assigns this container image to all roles that do not have another image specified.
   - **Selected:** Checking this radio button and then selecting one or more role(s) using the pull-down menu assigns this container image to the specified role(s). Checking a checkbox next to a role name assigns the image to that role; clearing a checkbox unassigns that image from the role.
7. Click **Add** to finish adding the image.

You may now:

- View the image/role mapping, as described in Viewing Images.
- Edit the image/role mapping, as described in Editing an Existing Image.
- Remove the image/role, as described in Removing an Image.

Viewing Images

The table at the bottom of the **EPIC Images** screen appears when you have defined at least one image/role mapping for this application. This table displays the following information for each image/role mapping:

- **Repo Tag:** Repo tag of the image/role mapping.
- **Directory:** If the container image is sourced locally, this is the path to that image.
- **Roles:** Role(s) to which this container image has been mapped.
- **OS Type:** Container image operating system.
- **Edit:** Clicking the **Edit** icon (pencil) for a service allows you to edit that role. See Editing an Existing Role.
- **Delete:** Clicking the **Delete** icon (X) for a service removes that role. See Removing a Role.

Editing an Existing Image

To edit an existing image/role mapping:

1. In the table at the bottom of the **EPIC Images** screen, click the **Edit** icon (pencil) for the image/role mapping you want to edit.
2. The top of this section populates with the current information for the selected image/role mapping.
3. Make your desired changes. See Adding a New Image for information on what to place in the fields.
4. Either:
   - Click **Add** to save your changes as a new image/role mapping.
   - Click **Update** to save your changes to the existing image/role mapping.
   - Click **Reset** to cancel your changes without modifying the image/role mapping.

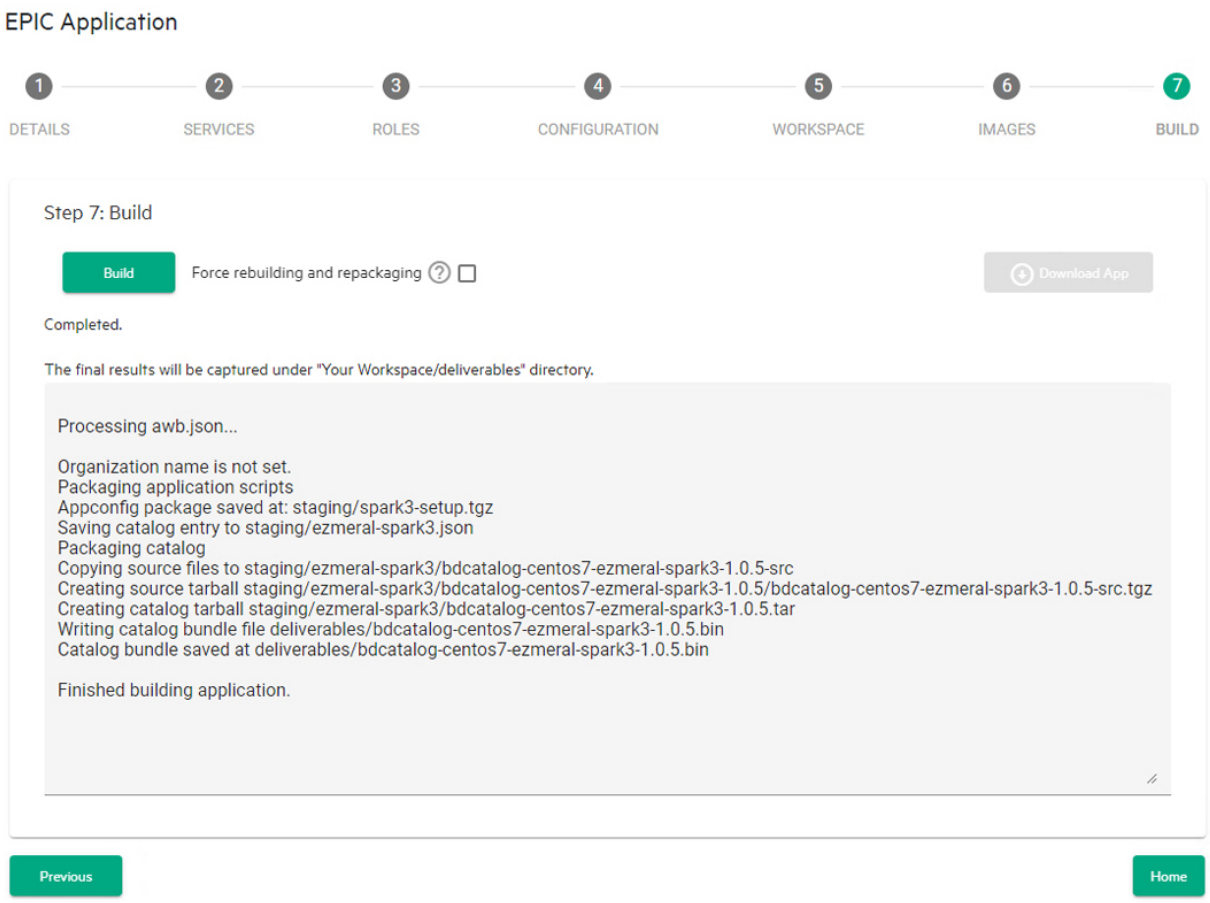
Removing an Image

To remove an image/role mapping, click the **Delete** icon (X) for the mapping you want to remove in the table at the bottom of the **EPIC Images** screen.

The EPIC Build Screen

Clicking the **Next** button in the **EPIC Images** screen (see The EPIC Images Screen) opens the **Build** screen, which is where you can view and download the JSON file and build the application.

last-updated: May 06, 2024



To build your application:

1. If you want to force the build process to rebuild and repackage the application even if the image or package already exists, then check the Fore rebuilding and repacking checkbox.

2. Click the **Build** button. Clicking this button starts the build process. The text area in the center of this screen displays the application build logs. Once the application completes successfully, the .bin file will be located in the `<workspace>/deliverables/` directory. For example:

```
[sampleuser@prod21 AWB5.1]# ls
appconfig awb20210206.log awb.json bdwb deliverables documentation.md image
logo.png staging status.json
[sampleuser@prod21 AWB5.1]# cd deliverables/
bdcatalog-centos7-ezmeral-spark3-1.o.5.bin
[sampleuser@prod21 AWB5.1]#
```

You may now:

- Download the application by clicking the **Download App** button.

- Manually add this application to HPE Ezmeral Runtime Enterprise. The new application appears in the **App Store** screen.

Once the build process has completed, you may click **Home** to return to the the **Application Status** screen. See The Application Status Screen.

Custom Base Images

About Custom Base Images

Hewlett Packard Enterprise provides publicly available base OS images for use in containerized clusters. These images extend the base OS images available from Docker hub by adding several packages that permit HPE Ezmeral Runtime Enterprise to manage container orchestration seamlessly and to improve the security of the container.

Applications that will run on EPIC deployments of HPE Ezmeral Runtime Enterprise require the use of the custom base images that are provided by Hewlett Packard Enterprise. KubeDirector applications are not required to use the custom base images. For information about container images and KubeDirector applications, see App Definition Authoring for KubeDirector (link opens an external website in a new browser tab or window).

The list of base images provided by Hewlett Packard Enterprise includes:

- `FROM bluedata/centos7:latest`
- `FROM bluedata/centos8:latest`
- `FROM bluedata/rhel7:latest`
- `FROM bluedata/rhel8:latest`
- `FROM bluedata/ubuntu18:latest`

NOTE: HPE Ezmeral Runtime Enterprise does not support CentOS 8, RHEL 7, or Ubuntu 16 container images.

If needed, you may build base images for use in HPE Ezmeral Runtime Enterprise starting from your own internal CentOS, RHEL, and/or Ubuntu images. You may also rebuild the containerized applications provided by Hewlett Packard Enterprise to meet your specific needs. Custom images can also be used to build new applications using the App Workbench. The articles in this section describe how you can build your own base image that can be used for further image development.

Please click the appropriate link for instructions on building a base image for your OS version:

- CentOS 7.x
- CentOS 8.x
- RHEL 7.x
- RHEL 8.x
- Ubuntu

CentOS 7.x

To build a custom CentOS 7.x base image:

1. SSH into the system where App Workbench is installed.
2. Create a directory by executing the `mkdir` command, such as:

```
$> mkdir -p ~/src/base_images.
```

3. Switch to the directory you just created by executing the `cd` command, such as:

```
$> cd ~/src/base_images
```

.

4. Retrieve the BlueData base image for CentOS 7 by executing the following command:

```
$> bdwb --baseimg centos7
```

.

This creates a directory called `centos7` under your current directory.

5. Switch to the `centos7` directory by executing the following command:

```
$> cd centos7
        $> ls -a
```

You should see the following:

```
                        drwxr-xr-x. 3 root root   54 Oct 10 08:59 .
                        drwxr-xr-x. 7 root root   78 Oct 15 12:32 ..
                        -rw-r--r--. 1 root root 4361 Oct  9 15:36 build.sh
                        -rw-r--r--. 1 root root 1393 Oct  9 15:36 Makefile
                        drwxr-xr-x. 3 root root  121 Oct 15 14:32 template
```

6. You may override one or more of the following parameter(s) by executing the following command(s), as appropriate:
   - `$> export BASE_IMG_ORGNAME='<orgname>'`, where `<orgname>` is the name of your organization, such as `enterprise`. The default name is `bluedata`.
   - `$> export BASE_IMG_VERSION='<version>'`, where `<version>` is the image version number, such as `1.0`. The default version is `4.1`.
   - `$> export UPSTREAM='<upstream>'`, where `<upstream>` is the name of the upstream image source, such as `artifactory.com/enterprise:centos7`. The default upstream image source is `centos:centos7`.
7. Modify the base image as needed.
8. Make the new image by executing the following command:

```
$> make centos7
```

9. Verify that the image has built successfully by executing the following command:

```
$> docker images
```

.

CentOS 8.x

To build a custom CentOS 8.x base image:

1. SSH into the system where App Workbench is installed.

2. Create a directory by executing the `mkdir` command, such as:

```
$> mkdir -p ~/src/base_images.
```

3. Switch to the directory you just created by executing the `cd` command, such as:

```
$> cd ~/src/base_images
```

.

4. Retrieve the BlueData base image for CentOS 7 by executing the following command:

```
$> bdwb --baseimg centos7
```

.

This creates a directory called `centos7` under your current directory.

5. Switch to the `centos7` directory by executing the following command:

```
$> cd centos8
        $> ls -a
```

You should see the following:

```
                        drwxr-xr-x. 3 root root   54 Oct 10 08:59 .
                        drwxr-xr-x. 7 root root   78 Oct 15 12:32 ..
                        -rw-r--r--. 1 root root 4361 Oct  9 15:36 build.sh
                        -rw-r--r--. 1 root root 1393 Oct  9 15:36 Makefile
                        drwxr-xr-x. 3 root root  121 Oct 15 14:32 template
```

6. You may override one or more of the following parameter(s) by executing the following command(s), as appropriate:
   - `$> export BASE_IMG_ORGNAME='<orgname>'`, where `<orgname>` is the name of your organization, such as `enterprise`. The default name is `bluedata`.
   - `$> export BASE_IMG_VERSION='<version>'`, where `<version>` is the image version number, such as `1.0`. The default version is `4.1`.
   - `$> export UPSTREAM='<upstream>'`, where `<upstream>` is the name of the upstream image source, such as `artifactory.com/enterprise:centos7`. The default upstream image source is `centos:centos7`.
7. Modify the base image as needed.
8. Make the new image by executing the following command:

```
$> make centos8
```

9. Verify that the image has built successfully by executing the following command:

```
$> docker images
```

.

RHEL 7.x

To build a custom RHEL 7.x base image:

1. SSH into the system where App Workbench is installed.
2. Create a directory by executing the `mkdir` command, such as:

```
$> mkdir -p ~/src/base_images.
```

3. Switch to the directory you just created by executing the `cd` command, such as:

```
$> cd ~/src/base_images
```

.

4. Retrieve the BlueData base image for RHEL 7 by executing the following command:

```
$> bdwb --baseimg rhel7
```

.

This creates a directory called `rhel7` under your current directory.

5. Switch to the `rhel7` directory by executing the following command:

```
$> cd rhel7
        $> ls -a
```

You should see the following:

```
drwxr-xr-x. 3 root root   54 Oct 10 08:59 .
drwxr-xr-x. 7 root root   78 Oct 15 12:32 ..
-rw-r--r--. 1 root root 4361 Oct  9 15:36 build.sh
-rw-r--r--. 1 root root 1393 Oct  9 15:36 Makefile
drwxr-xr-x. 3 root root   79 Oct 10 08:59 template
```

6. You may override one or more of the following parameter(s) by executing the following command(s), as appropriate:
   - `$> export BASE_IMG_ORGNAME='<orgname>'`, where `<orgname>` is the name of your organization, such as `enterprise`. The default name is `bluedata`.
   - `$> export BASE_IMG_VERSION='<version>'`, where `<version>` is the image version number, such as `1.0`. The default version is set to `4.1`.
   - `$> export UPSTREAM='<upstream>'`, where `<upstream>` is the name of the upstream image source, such as `artifactory.com/enterprise:rhel7`. The default upstream image source is `rhel:rhel7`.

   NOTE: This base image can only be built on a RHEL server.

7. Modify the base image as needed.
8. Make the new image by executing the following command:

```
$> make rhel7
```

9. Verify that the image has built successfully by executing the following command:

```
$> docker images
```

.

RHEL 8.x

To build a custom RHEL 8.x base image:

1. SSH into the system where App Workbench is installed.
2. Create a directory by executing the `mkdir` command, such as:

```
$> mkdir -p ~/src/base_images.
```

3. Switch to the directory you just created by executing the `cd` command, such as:

```
$> cd ~/src/base_images
```

.

4. Retrieve the BlueData base image for RHEL 7 by executing the following command:

```
$> bdwb --baseimg rhel7
```

.

This creates a directory called `rhel8` under your current directory.

5. Switch to the `rhel8` directory by executing the following command:

```
$> cd rhel8
        $> ls -a
```

You should see the following:

```
                        drwxr-xr-x. 3 root root   54 Oct 10 08:59 .
                        drwxr-xr-x. 7 root root   78 Oct 15 12:32 ..
                        -rw-r--r--. 1 root root 4361 Oct  9 15:36 build.sh
                        -rw-r--r--. 1 root root 1393 Oct  9 15:36 Makefile
                        drwxr-xr-x. 3 root root   79 Oct 10 08:59 template
```

6. You may override one or more of the following parameter(s) by executing the following command(s), as appropriate:
   - `$> export BASE_IMG_ORGNAME='<orgname>'`, where `<orgname>` is the name of your organization, such as `enterprise`. The default name is `bluedata`.
   - `$> export BASE_IMG_VERSION='<version>'`, where `<version>` is the image version number, such as `1.0`. The default version is set to `4.1`.
   - `$> export UPSTREAM='<upstream>'`, where `<upstream>` is the name of the upstream image source, such as `artifactory.com/enterprise:rhel8`. The default upstream image source is `rhel:rhel8`.

   NOTE: This base image can only be built on a RHEL server.

7. Modify the base image as needed.
8. Make the new image by executing the following command:

```
$> make rhel8
```

9. Verify that the image has built successfully by executing the following command:

```
$> docker images
```

.

Ubuntu

To build a custom Ubuntu 18 base image:

1. SSH into the system where App Workbench is installed.
2. Create a directory by executing the `mkdir` command, such as:

```
$> mkdir -p ~/src/base_images.
```

3. Switch to the directory you just created by executing the `cd` command, such as:

```
$> cd ~/src/base_images
```

.

4. Retrieve the HPE base image for Ubuntu 18 by executing the following command:

```
$> bdwb --baseimg ubuntu18
```

.

This creates a directory called `ubuntu18` under your current directory.

5. Switch to the `ubuntu18` directory by executing the following command:

```
$> cd ubuntu18
        $> ls -a
```

You should see the following:

```
drwxr-xr-x. 3 root root   54 Oct 10 08:59 .
drwxr-xr-x. 7 root root   78 Oct 15 12:32 ..
-rw-r--r--. 1 root root 1393 Oct  9 15:36 Makefile
drwxr-xr-x. 3 root root   79 Oct 10 08:59 ubuntu18
```

6. You may override one or more of the following parameter(s) by executing the following command(s), as appropriate:
   - `$> export BASE_IMG_ORGNAME='<orgname>'`, where `<orgname>` is the name of your organization, such as `enterprise`. The default name is `bluedata`.
   - `$> export BASE_IMG_VERSION='<version>'`, where `<version>` is the image version number, such as `1.0`. The default version is the `EPIC_BASE_IMG_VERSION`.

- `$> export UBUNTU18_UPSTREAM='<upstream>'`, where `<upstream>` is the name of the upstream image source, such as `artifactory.com/enterprise:ubuntu18`. The default upstream image source is `ubuntu:ubuntu18`.

7. Modify the base image as needed.
8. Make the new image by executing the following command:

```
$> make ubuntu18
```

9. Verify that the image has built successfully by executing the following command:

```
$> docker images
```

.

NOTE: User authentication is not automatically set up for applications that use the `bluedata/ubuntu18` base image.

Resources

BDWB Shell Commands

This article describes `bdwb`, the command line tool for the App Workbench. This tool allows you to perform various functions, such as:

- Add or modify images.
- Create and modify **App Store** entries.
- Manage roles in a multi-node, multi-service deployment.
- Register services.
- Configure clusters.
- Add `.conf` files and `init.d` scripts to an application package.

Running BDWB

To run `bdwb`: switch to the directory where you want to begin creating the new application. You can run it one of two ways:

- **Batch Mode:** You can run `bdwb` in a non-interactive mode directly from the command line, as described in Command Line Options, below.
- **Interactive:** You can run `bdwb` using the interactive `bdwb` shell, as described in Interactive Commands, below.

Command Line Options

NOTE: This section describes the options available in version 3.6 of the App Workbench.

You can run `bdwb` from the command line by executing either of the following commands:

- `bdwb -i <instruction>`, where `<instruction>` is one of the commands described in Interactive Commands, below. The specified instruction executes immediately.
- `bdwb <file_name>.wb`, where `<file_name>.wb` is the name of a Workbench file that contains instructions for non-interactive processing.

Interactive Commands

To run `bdwb` interactively, execute the command `bdwb` from the command line. You can then run the following commands:

- `appconfig`: See AppConfig, below.
- `attach`: See Attach, below.
- `baseimg`: See Base Image, below.
- `builder`: See Builder, below.
- `catalog`: See Catalog, below.
- `clusterconfig`: See Cluster Config, below.
- `define`: See Define, below.
- `document`: See Document, below.
- `EOF`: See EOF, below.
- `exit`: See Exit, below.
- `help`: See Help, below.
- `image`: See Image, below.
- `logo`: See Logo, below.
- `role`: See Role, below.
- `service`: See Service, below.
- `sources`: See Sources, below.
- `workbench`: See Workbench, below.

AppConfig

The `appconfig` command manages the packages that comprise an **App Store** bundle. This command has the following sub-commands:

- `list`: List the configuration of the given package.
- `file`: Add a local file path to the given AppConfig package. This sub-command uses the following arguments:
  - `-f <path>` or `--filepath <path>`: Full path to the AppConfig package on the local filesystem. Default is none.
  - `--md5sum`: The MD5 checksum of the AppConfig package. If not specified, the file checksum is calculated. Default is none.

last-updated: May 06, 2024

  - `--configapi <version>`: Config API version used by the AppConfig package. Default is latest version, if not specified. See API Matrices on page 1077.

- `autogen`: Auto-generates a simple AppConfig bundle. This sub-command uses the following arguments:
  - `--new <true|false>`: Starts a new auto-generation package process and overrides any previous auto-generation package progress. Default is `false`.
  - `--configapi <version>`: Config API version used by the AppConfig package. Default is latest version, if not specified. See API Matrices on page 1077.
  - `--generate <true|false>`: Auto-generate the AppConfig package that you previously created using the `--new` argument. Default is `false`.

    The following arguments are used to copy files to the deployed virtual node:

  - `--pkgfile <file_name <file_name ... >>`: Name(s) of the file(s) already in the AppConfig directory to be used as the source file(s). Default is none.
  - `--destdir <directory>`: Destination directory where the given file(s) is(are) to be copied on a deployed node. Default is none.
  - `--dest <absolute_path>`: Absolute path where the local file should be placed inside the container. Any directories necessary to put the file at the location are created. Default is none.

    The following argument appends files with cluster-specific properties:

  - `--append <file_name>`: Append files with cluster-specific properties. Absolute path of the configuration file inside the container. Default is none.

    The following arguments provide various options for handling custom scripts:

  - `--execute <file_name>`: Absolute path of the file to be executed inside the container. Default is none.
  - `--sourcefile <file_path>`: Path of the file to be sourced. Default is none.
  - `--onroles <role_1> <role_2> ... <role_n>`: Conditionally execute or source file on the specific virtual node role(s). By default, the scripts are executed or sourced on all the virtual node roles. Default is none.

    The following arguments are used to assign permissions to files or directories in the container image:

  - `--abspath <absolute_path>`: Absolute path of the file or directory whose permissions are being set. Default is none.
  - `--perms <permissions>`: RWX permissions to set for this file or directory. Default is none.
  - `--uid <uid>`: UID to set for the file or directory. Default is none.
  - `--gid <gid>`: GID to set for the file or directory. Default is none.

    The following arguments provide pattern replacement instructions for auto-generating config file customization.

  - `--replace <file_name>`: Absolute path of the configuration file inside the container. Default is none.
  - `--pattern <pattern>`: Pattern replacement instructions for auto-generating config file customization. Default is none. See Macros and Keys on page 1038.
  - `--macro ...`: A command whose output is used to replace the pattern. This could be an SDK-defined function or a simple command invocation. Default is none. See Macros and Keys on page 1038.
- `package`: packages the AppConfig directory being developed. This sub-command uses the following arguments:
  - `-d <package_directory>` or `--dir <package_directory>`: A directory where the AppConfig scripts being developed are located. The directory name is used as the package name. If this argument is not specified, then the auto-generated AppConfig will be packaged. Default is none.
  - `--configapi <version>`: Config API version used by the AppConfig package. Default is latest version, if not specified. See API Matrices on page 1077.
- `init`: When you are manually developing an AppConfig script, this sub-command copies a few useful scripts that you can use as starter code. This sub-command uses the `-d <destination_directory>` or `--dir <destination_directory>` argument, which is the directory where the starter code is to be copied. This directory and all its parents will be created if they do not already exist. Default is none.
- `download`: Downloads the AppConfig package from an HTTP url and adds it to the **App Store** entry. This sub-command uses the following arguments:
  - `-u <setup_package_url>` or `--url <setup_package_url>`: HTTP URL for downloading the AppConfig package. The file is downloaded to the staging directory. Default is none.
  - `--md5sum <md5_sum>`: The MD5 checksum of the AppConfig package, which is used to verify the checksum immediately after downloading. Default is none.
  - `--configapi <api_version>`: Config API version used by the AppConfig package. Default is latest version, if not specified. See API Matrices on page 1077.

Attach

The `attach` command defines what other application can be attached. This command has the following sub-commands:

- `distro`: Attach applications based on distro IDs. This sub-command uses the following argument:
  - `-d <distro_id_1>...<distro_id_n>` or `--distroid <distro_id_1>...<distro_id_n>`: A space-separated list of unique distro IDs that this application can attach to. Default is none.
- `service`: Attach applications based on exported services and qualifiers. This sub-command uses the following arguments:
  - `-s <service>` or `--service <service>`: An application with the given service name is attachable to this cluster. Default is none.
  - `-q <qualifier>` or `--qualifier <qualifier>`: An optional service qualifier that further narrows down the attachable applications. Default is none.
- `category`: Attach applications based on categories. This sub-command uses the following argument:

- `-c <category_1>...<category_n>` or `--category <category_1>...<category_n>`: A space-separated list of categories that this entry will be available under during cluster creation. Any existing categories may be used, or new ones may also be defined here. Default is none.

Base Image

The `baseimg` command defines the base OS image to use for the application. This command uses the following sub-command:

- `init`: When manually developing appconfig script, this copies a few useful scripts that you can use as starter code. This sub-command uses the following argument:
  - `--os {centos7,rhel7,centos8,rhel8,ubuntu18}`: Copies all the files related to building a Docker image that can be used as a base for apps on HPE Ezmeral Runtime Enterprise. These files are copied to the current directory. Default is none.

Builder

The `builder` command sets the organization name for the **App Store** entry. The organization name is used to disambiguate the distro ID of the entry as well as the name of the Docker image imported. This command uses the syntax `builder -n <organization>` or `builder --name <organization>`, where `<organization>` is the organization name to use for the entry. This must be a single word with no spaces. The input will be converted to all-lowercase if any mixed or uppercase characters are used. Default is none.

Catalog

The `catalog` command manages the **App Store** entry. This command has the following sub-commands:

- `new`: Starts a session for creating a new **App Store** entry. Any previously-started sessions will be lost unless they were saved. This sub-command can use the following optional arguments:
  - `--distroid <distro_id>`: A distro ID that is unique across the entire HPE Ezmeral Runtime Enterprise Catalog. Default is none.
  - `--name <name>`: The name of the App Store entry. If the name includes spaces, then enclose it in double quotes, for example `"Application Name"`. Default is none.
  - `--depends_on <distro_id>`: The distro ID of another **App Store** entry that this entry depends on. When used, this sub-command indicates that this is an add-on image. Default is none.
  - `--desc <description>`: The description for the App Store entry. Use double quotes to enclose the description. For example, `"This is an application description."`. Default is none.
  - `-v <version>` or `--version <version>`: Version of the **App Store** image formatted as `x.y`, where `x` is the major version and `y` is the minor version. Default is `1.0`.
  - `-c <categories <categories... >` or `--categories <categories <categories... >`: Space-separated list of categories that this **App Store** entry will be available under during cluster creation. You may use any existing category or create a new one. Default is `Hadoop`.
  - `--catalogapi <api_version>`: Catalog API version used by the AppConfig package. Default is latest version, if not specified. See API Matrices on page 1077.

  - `--epic <target_epic_version>`: The target HPE Ezmeral Container Platform version where the application package will be installed, such as

    ```
    --epic 5.2
    ```

    . See API Matrices on page 1077.

- `save`: Saves the current in-memory state of the **App Store** entry to a file. This sub-command can use the following optional arguments:
  - `-f <path>` or `--filepath <path>`: File path where to save the **App Store** entry JSON file. If not specified, the JSON file will be saved in the `'staging_dir'` defined in `bench.conf`. Default is none.
  - `--force <true|false>`: Overwrites an existing catalog entry JSON file, if any. Default is `false`.
- `load`: Loads an existing **App Store** entry. This sub-command uses the syntax `load -f <:file_path>` or `load --filepath <:file_path>`, which is the file path to an existing **App Store** entry JSON file.
- `modify`: Allows selected fields in the **App Store** entry to be updated. This sub-command uses the following optional arguments:
  - `--distroid <distro_id>`: A distro ID that is unique across the entire HPE Ezmeral Runtime Enterprise Catalog. Default is none.
  - `--name <name>`: Catalog name for the end user. If the name includes spaces, then enclose it in double quotes, for example `"Application Name"`. Default is none.
  - `--depends_on <distro_id>`: Distro ID of another **App Store** entry that this entry depends on. When used, this sub-command indicates that this is an add-on image. Default is none.
  - `--desc <description>`: Description for the end user. Use double quotes to enclose the description. For example, `"This is an application description."`. Default is none.
  - `-v <version>` or `--version <version>`: Version of the **App Store** formatted as `x.y`, where `x` is the major version and `y` is the minor version. Default is none.
  - `-r or -recommend_distro`: When set allows HPE Ezmeral Runtime Enterprise to recommend this distro when creating clusters in AI/ML tenants.
  - `-c <categories <categories... >` or `--categories <categories <categories... >`: Space-separated list of categories that this **App Store** entry will be available under during cluster creation. You may use any existing category or create a new one. Default is `Hadoop`.
  - `-a <aiml_categories> or -aiml-categories <aiml_categories>`: A space separated list of AIML categories. The available options are: `AIML/Notebook`, `AIML/Training`, and/or `AIML/Deployment`.
  - `--catalogapi <api_version>`: Catalog API version used by the AppConfig package. Default is latest version, if not specified. See API Matrices on page 1077.
  - `-epic <epic_version>`: HPE Ezmeral Runtime Enterprise version the current app is being built for. The available options are: 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 5.0, 5.1, 5.2, and 5.3.
- `package`: Package all components of the **App Store** entry into a bundle (.bin) file. This sub-command uses the following optional arguments:

- `-v <true|false>` or `--verbose <true|false>`: Show details of the packing process. Default is `false`.
- `-o <operating_system>` or `--os <operating_system>`: Determines which type of HPE Ezmeral Runtime Enterprise installation this **App Store** entry should be available for (`any`, `centos`, or `rhel`). For example, an RHEL entry will not show on an HPE Ezmeral Runtime Enterprise deployment that is installed on CentOS. Default is `centos`.

Cluster Config

The `clusterconfig` command manages the cluster configuration for an **App Store** entry. This command has the following sub-commands:

- `new`: Creates a new configuration that maps various services and roles with each other. This command uses the syntax `new -c <config_id>` or `new --configid <config_id>` to specify a unique **App Store** configuration ID.
- `list`: Lists details about the various cluster configurations for the current **App Store**. When one or more `<config_id>` are specified, only details for those configurations will be displayed. This command uses the syntax `list <config_id <config_id ... >>` to provide one or more space-separated config ID(s) for which to show details. If no IDs are provided, then the details of all currently-configured config IDs will be shown. Default is `all`.
- `assign`: Defines a cluster configuration which essentially links various roles with the expected services. Multiple calls of this command with the same `<config_id>` can be used to associate different roles and service(s). This sub-command uses the following optional arguments:
  - `-c <config_id>` or `--configid <config_id>`: A unique **App; Store** configuration ID.
  - `-r <role_id>` or `--roleid <role_id>`: A unique **App; Store**-wide role identifier.
  - `-s <service_id <service_id ...>` or `--srvcids <service_id <service_id ... >`: Service ID(s) to be assigned to the role when this configuration is enabled. Default is none.

Define

The `define` command defines variables and/or constraints. This command has the following sub-command:

`var <key_1=value_1> <key_2=value2> ... <key_n=value_n>`: Defines one or more key(s) and assigns a value to each key. All occurrences of %KEY% in any subsequent App Workbench command will be replaced by the defined value. Neither the key nor the value may include a percent symbol (%), equal sign (=) or spaces.

Document

The document command provides container documentation management for the catalog entry. This command uses the following sub-commands:

- file: Adds a local file path to the documentation for the catalog entry. This sub-command uses the following arguments:
  - `-f <file_path>` or `--filepath <file_path>`: File path to the documentation file on the local filesystem. Default is none.
  - `-m <md5_sum>` or `--md5sum <md5_sum>`: MD5 checksum of the documentation file. If this is not specified, then the checksum for the file is calculated. Default is none.

  - `-t <mime_type> or --mimetype <mime_type>`: Overrides the MIME type of the document file (such as text/markdown). If not specified, the MIME type is guessed from the file extension. Default is none.

- `list`: Lists the configured document. This sub-command does not use any arguments.
- `download`: Downloads the document file from an HTTP URL and adds it to the catalog feed entry. This sub-command uses the following arguments:
  - `-l <document_url>` or `--url <document_url>`: HTTP URL for downloading the document file. The file is downloaded to the staging directory. Default is none.
  - `--md5sum <md5_checksum>`: MD5 checksum of the document file that is used to verify the checksum immediately after downloading. Default is none.
- `init`: Creates a template documentation markdown file. This sub-command uses the following argument:
  - `--force {true,false}`: Overwrite the documentation template file if it already exists. Default is `false`.

EOF

The `EOF` command exits the interactive `bdwb` shell.

Exit

The `exit` command exits the interactive `bdwb` shell.

Help

The `help` command provides help with the `bdwb` commands. It can be used as follows:

- `help`: Lists the top-level `bdwb` commands.
- `help <command>`: Describes the selected command. For example, `help autogen` provides help about the `autogen` command.
- `help <command> <option>`: Describes the selected option. For example, `help autogen destdir` provides help about the `destdir` sub-command within the `autogen` command.

Image

The `image` command manages container images for the **App Store** entry. It has the following sub-commands:

- `build`: Build an **App Store** image from a Dockerfile. Additional arguments can be passed to the `docker build` command by setting the environment variable `AWB_DOCKER_BUILD_OPTS`. This sub-command uses the following optional arguments:
  - `-b <base_directory>` or `--basedir <base_directory>`: Directory path where the Dockerfile and related files are located. Default is none.
  - `-i <repotag>` or `--image-repotag <repotag>`: Container name and tag for the newly-built image. This usually takes the form `REGISTRY_HOST[:REGISTRY_PORT]/]REPOSITORY[:TAG]`. Default is none.
  - `-t <tags>` or `--additional-tags <tags>`: Create additional tag(s) for the image that is built.
- `download`: Download the image file from an HTTP URL and add it to the **App Store** entry.

  - `--md5sum`: The MD5 checksum of the image file. Used to verify the checksum immediately after downloading. Default is none.
  - `--os <os>`: OS distribution of the container image. Default is none.
  - `--roles <role_1> <role_2> ... <role_n>`: Assign the image to one or more specific virtual node role(s). If specified, the image is used for the role(s), such as `master` or `worker`, when that image is deployed on an HPE Ezmeral Runtime Enterprise cluster. This sub-command is only supported when the Catalog API version is 4 or higher. Also, the role must be previously defined in the metadata JSON file, such as by using the `role add` command. Default is `all_roles`.
  - `-u <image_url>` or `--url <image_url>`: HTTP URL for downloading the image. The file is downloaded to the staging directory. Default is none.
- `list`: Lists the configured container image.
- `load`: Loads an image. This sub-command uses the following optional arguments:
  - `-f <file_path>` or `--filepath <file_path>`: File path to the container image on the local filesystem. Default is none.
  - `-i <repotag>` or `--image-repotag <repotag>`: Container name and tag to save in the metadata. This usually takes the form `REGISTRY_HOST[:REGISTRY_PORT]/]REPOSITORY[:TAG]`. Default is none.
  - `--md5sum <sum>`: MD5 checksum of the `appconfig` package. If this is not specified, then the system calculates the checksum for the file. Default is none.
  - `--os <os>`: OS distribution of the container image. Default is none.
  - `--roles <role_1> <role_2> ... <role_n>`: Assign the image to one or more specific virtual node role(s). If specified, the image is used for the role(s), such as `master` or `worker`, when that image is deployed on an HPE Ezmeral Container Platform cluster. This sub-command is only supported when the Catalog API version is 4 or higher. Default is `all_roles`.
- `package`: Pulls a Docker image and then packages it into the Catalog entry as a file. This sub-command uses the following optional arguments:
  - `-i <repotag>` or `--image-repotag <repotag>`: Container name and tag to save in the metadata. This usually takes the form `REGISTRY_HOST[:REGISTRY_PORT]/]REPOSITORY[:TAG]`. Default is none.
  - `--os <os>`: OS distribution of the container image. Default is none.
  - `--roles <role_1> <role_2> ... <role_n>`: Assign the image to one or more specific virtual node role(s). If specified, the image is used for the role(s), such as `master` or `worker`, when that image is deployed on an HPE Ezmeral Container Platform cluster. This sub-command is only supported when the Catalog API version is 4 or higher. Default is `all_roles`.

    The following arguments are available when the Docker registry requires authentication:
  - `-u <username>` or `--username <username>`: Username to be used when pushing the Docker image to a registry that requires authentication. You may also set the environment variable `AWB_REGISTRY_USERNAME`. Default is none.

- `-p <password>` or `--password <password>`: Password to be used when pushing the Docker image to a registry that requires authentication. You may also set the environment variable `AWB_REGISTRY_PASSWORD`. Default is none.

  The following arguments are available when the Docker registry has content trust enabled:

- `-o <passphrase>` or `--ct-root-passphrase <passphrase>`: Specifies the content trust root passphrase, if content trust is enabled for the Docker registry. You may also set the environment variable `DOCKER_CONTENT_TRUST_ROOT_PASSPHRASE`. Default is none.

- `-r <passphrase>` or `--ct-registry-passphrase <passphrase>`: Specifies the content trust repository passphrase, if content trust is enabled for the Docker registry. You may also set the environment variable `DOCKER_CONTENT_TRUST_REPOSITORY_PASSPHRASE`. Default is none.

- `push`: Pushes a Docker image to a registry and refers to it in the Catalog entry metadata. This sub-command uses the following optional arguments:

  - `-i <repotag>` or `--image-repotag <repotag>`: Container name and tag to save in the metadata. This usually takes the form `REGISTRY_HOST[:REGISTRY_PORT]/]REPOSITORY[:TAG]`. Default is none.

  - `--os <os>`: OS distribution of the container image. Default is none.

  - `--roles <role_1> <role_2> ... <role_n>`: Assign the image to one or more specific virtual node role(s). If specified, the image is used for the role(s), such as `master` or `worker`, when that image is deployed on an HPE Ezmeral Container Platform cluster. This sub-command is only supported when the Catalog API version is 4 or higher. Default is `all_roles`.

    The following arguments are available when the Docker registry requires authentication:

  - `-u <username>` or `--username <username>`: Username to be used when pushing the Docker image to a registry that requires authentication. You may also set the environment variable `AWB_REGISTRY_USERNAME`. Default is none.

  - `-p <password>` or `--password <password>`: Password to be used when pushing the Docker image to a registry that requires authentication. You may also set the environment variable `AWB_REGISTRY_PASSWORD`. Default is none.

    The following arguments are available when the Docker registry has content trust enabled:

  - `-o <passphrase>` or `--ct-root-passphrase <passphrase>`: Specifies the content trust root passphrase, if content trust is enabled for the Docker registry. You may also set the environment variable `DOCKER_CONTENT_TRUST_ROOT_PASSPHRASE`. Default is none.

  - `-r <passphrase>` or `--ct-registry-passphrase <passphrase>`: Specifies the content trust repository passphrase, if content trust is enabled for the Docker registry. You may also set the environment variable `DOCKER_CONTENT_TRUST_REPOSITORY_PASSPHRASE`. Default is none.

- `registry`: Docker registry information. This sub-command uses the following arguments:

  - `--auth-enabled`: Specifies that the registry requires authentication. Please set the environment variables `AWB_REGISTRY_USERNAME` and `AWB_REGISTRY_PASSWORD` before invoking App Workbench. Default is `False`.

  - `--trust`: Specifies that content trust is enabled for the Docker images.
    Please set the environment variables `DOCKER_CONTENT_TRUST_ROOT_PASSPHRASE` and `DOCKER_CONTENT_TRUST_REPOSITORY_PASSPHRASE` before invoking App Workbench. Default is `False`.
  - `--url <registry_host:port>`: Registry URL and port specification. Default is none.
- `pull`: Pulls an image from the repository. This sub-command uses the following arguments:
  - `-i <image_reportag>` or `--image-repotag <image_repotag>`: Container name and tag for the newly-built image. This usually takes the form `REGISTRY_HOST[:REGISTRY_PORT]/]REPOSITORY[:TAG]`. See Macros and Keys on page 1038 for more details. Default is none.
  - `-t <new_repotag>` or `--retag <new_repotag>`: Re-tag the image after pulling it from the remote registry. Default is none.
  - `-u <username>` or `--username <username>`: For an authentication-enabled registry, specifies the username for pulling the image from that registry. You may also set the environment variable `AWB_REGISTRY_USERNAME`. Default is none.
  - `-p <password>` or `--password <password<>`: For an authentication-enabled registry, specifies the password for pulling the image from that registry. You may also set the environment variable `AWB_REGISTRY_PASSWORD`. Default is none.
  - `-o <passphrase>` or `--ct-root-passphrase <passphrase>`: For a content-trust-enabled registry, specifies the content trust root passphrase. You may also set the environment variable `DOCKER_CONTENT_TRUST_ROOT_PASSPHRASE`. Default is none.
  - `-r <passphrase>` or `--ct-registry-passphrase <passphrase>`: For a content-trust-enabled registry, specifies the content trust repository passphrase. You may also set the environment variable `DOCKER_CONTENT_TRUST_REPOSITORY_PASSPHRASE`. Default is none.

Logo

The `logo` command manages the container logo for the **App Store** entry. This command has the following sub-commands:

- `file`: Add a local file path for a logo to the **App Store** entry. This sub-command has the following optional arguments:
  - `-f <path>` or `--filepath <path>`: Full path to the image file on the local filesystem. Default is none.
  - `--md5sum <md5_checksum>`: The MD5 checksum of the logo image file. If not specified, the file checksum is calculated. Default is none.
- `list`: Lists the logo image file, if any.
- `download`: Download the logo file from a HTTP URL and add it to the **App Store** entry.
  - `-l <logo_url>` or `-url <logo_url>`: HTTP URL for downloading the logo. The file is downloaded to the staging directory. Default is none.
  - `--md5sum <md5_checksum>`: TheMD5 checksum of the logo image file. Used to verify the checksum immediately after downloading. Default is none.

Role

The `role` command manages the roles for the **App Store** entry. This command has the following sub-commands:

- `add`: Add a new role to the current **App Store** entry. This sub-command uses the syntax `add <role_id> <cardinality>`, where `<role_id>` is a role ID that is unique to the **App Store** entry and `<cardinality>` is the cardinality for the role. Cardinality is a defined as a number followed by an optional plus sign (+). The number indicates the minimum number of nodes of that particular role that HPE Ezmeral Container Platform will force user to deploy, and the optional plus sign (+) indicates to HPE Ezmeral Runtime Enterprise that the user must be allowed to choose any number above the absolute minimum. For example:
  - `0+` means "zero nodes or more," which is generally used for Worker roles.
  - `1+` means "one node or more."
  - `2` means "two and only two nodes" of this role are allowed.
- `list`: List details of role(s) defined in the current **App Store** entry. This sub-command uses the syntax `list <role_id <role_id... >>`, where you can add one or more space-separated role IDs. Default is `all`.
- `remove`: Remove role(s) from the current **App Store** entry. This sub-command uses the syntax `remove <role_id> <role_id... >>`, where you can remove one or more specific space-separated role IDs. Default is `all`.

Service

The `service` command manages the services that are part of **App Store** entry. This command has the syntax `service <option>>`, where `<option>` is one of the following:

- `add`: Adds a service to the **App Store** entry. This sub-command has the following arguments:
  - `--srvcid <service_id> -n <service_name>`, where:
    - `<service_id>` is the service ID, which is unique to the entire **App Store** entry. Default is none.
    - `<service_name>` is the name of the service to be displayed in the HPE Ezmeral Container Platform interface. Default is none.
  - `--export_as <export_name> -s <scheme> --port <port> --path <path> --display <true|false>`, where:
    - `<export_name>` is the name this service is exported as in the HPE Ezmeral Runtime Enterprise interface. Default is none.
    - `<scheme>` is the URI scheme for the service, if any. Default is none.
    - `<port>` is the URI port number, if any. Default is none.
    - `<path>` is the URI path for the service, if any. Default is none.
    - `--display` Displays the service to the user in the **Cluster Details** page. Adding this option sets it to `true`; default is `false`.
  - `--sysv <service_name>`: SystemV service name for managing the life cycle of the service. Default is none.

- `--sysctl <unit_name>`: SystemD unit name for managing the life cycle of the service. Default is none.
- `--onroles <role_1> <role_2> ... <role_n>`: Virtual node role(s) to which the service should be assigned by default. Default is none.

Sources

The `sources` command delivers source files as part of **App Store** bundles. This command has the syntax `sources <package>`, where `<package>` lists the package sources in the application bundle. The following files and directories are automatically packaged if they exist:

- `/appconfig`
- `/images`
- logo file (if specified using the `logo` command, as described in Logo, above). For example, `app_logo.png`
- instruction (if being used). For example, `app_package.wb`.
- metadata JSON file

Workbench

The `workbench` command manages a workspace for developing an **App Store** entry. This command has the syntax `workbench <option>>`, where `<option>` is one of the following:

- `initapp`: Initializes a workspace for developing a new **App Store** entry. This sub-command has the following arguments:
  - `-apptype <type_of_app>`: Type of application (either `EPIC` or `KubeDirector`).
  - `-f` or `--force`: Forces App Workbench initialization. Default is `false`.
- `clean`: Cleans up all temporary artifacts and log files that may have been generated during the package process.
- `version`: Displays the App Workbench version.

Macros and Keys

Application configuration scripts that need to be populated with dynamic values (such as an IP address or FQDN) can do so by specifying a pattern in that config file by including the following command in the .wb file:

```
appconfig autogen --replace <filename> --pattern <pattern> --macro <macro>
```

This command replaces the `<pattern>` with the output of the `<macro>` in the `<filename>`.

There are three types of macro:

- **Inline:** Simple commands like `echo $VARIABLE` may be defined directly when specifying the pattern replace instruction in the .wb file. Any valid bash statement can be used as an inline macro.
- **Macros** defined by App Workbench: This article describes those in detail.

- **User-defined macros:** Users can define their own macros in a file and use the command `appconfig autogen --sourcefile <file>` to make those macros available to the pattern replace command. The `sourcefile` command must be specified before any pattern replace command(s) that may use the custom macros.

The following macros are available:

NOTE: This list is presented in the same order in which it appears in the macro definition file `macros.sh`.

- **Node Details:** Gets details about a virtual node. See Node Details.
- **Get Cluster Configuration Choice:** Returns the value of the specified key. See Get Cluster Configuration Choice.
- **Get Cluster Configuration Metadata:** Returns the value of the specified metadata key. See Get Cluster Configuration.
- **Get Specific Configuration Choice Key:** Returns the value of a specific configuration choice key. See Get Specific Configuration Choice Key.
- **Get Specific Metadata Choice Key:** Returns the value of a specific metadata choice key. See Get Specific Metadata Choice Key.
- **Get a Unique Integer for This Host:** Returns an integer for the current host. See Get This Host Number.
- **Get a Unique Integer for Another Host:** Returns an integer for another host. See Get Other Host Number.
- **Get Unique Integer by Service:** Returns a unique integer for a service. See Get Unique Integer.
- **Get Total Available VRAM:** Returns the amount of available virtual RAM in MB. See Get Total Available VRAM.
- **Get Total Available VCPU Cores:** Returns the number of virtual CPU cores available to Spark. See Get Total Available VCPU Cores.
- **Get FQDN List:** Returns the FQDNs of the virtual nodes in the cluster. See Get FQDN List.
- **Get IP Address List:** Returns the IP addresses of the virtual nodes in the cluster. See Get IP Address List.
- **Get Virtual Node FQDN:** Returns the FQDN of a specific virtual node. See Get Virtual Node FQDN.
- **Get Virtual Node IP Address:** Returns the IP address of a specific virtual node. See Get Virtual Node IP Address.
- **Generate Application-Specific URL:** Generates a URL for the specified application. See Generate Application-Specific URL.
- **Get Tenant Information:** Gets the value of the specified tenant namespace key. See Get Tenant Information.

This article presents the following information for each macro:

- **Description:** Function of the macro.
- **Input(s):** Information that must be supplied to the macro.

last-updated: May 06, 2024

- **Output:** Output of the macro for both success and failure outcomes.
- **Usage Example:** Generic usage example for the macro.

Node Details

The following node details can be obtained for a Docker container (virtual node) in a cluster:

- FQDN
- Role
- Domain
- Distribution
- Hostname
- Nodegroup ID
- Dependencies
- DataTap.jar
- Cluster Name
- Total Virtual CPU Cores
- Total Virtual RAM

FQDN

Name

```
FQDN
```

Description

The fully qualified domain name of the Controller.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro FQDN
```

Role

Name

```
ROLE
```

Description

Role of the Container (such as Master, Worker, or Edge).

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro ROLE
```

Domain

Name

```
DOMAIN
```

Description

Domain to which the virtual node belongs.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro DOMAIN
```

Distribution

Name

```
DISTRO
```

Description

Distribution (App Store application) being run on the virtual node.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro DISTRO
```

Hostname

Name

```
HOSTNAME
```

Description

Hostname of the virtual node.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro HOSTNAME
```

Nodegroup ID

Name

```
NODEGROUP
```

Description

ID number of the nodegroup to which the virtual node belongs.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro NODEGROUP
```

Dependencies

Name

```
DEPENDS_ON
```

Description

The `distro_id` of the primary nodegroup that this node depends on.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro DEPENDS_ON
```

DataTap.jar

Name

```
DTAP_JAR
```

Description

The published location of the DataTap jar, which should be copied appropriately by the application configuration scripts.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro DTAP_JAR
```

Cluster Name

Name

```
CLUSTER_NAME
```

Description

Name of the cluster to which the virtual node belongs.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro CLUSTER_NAME
```

Total Virtual CPU Cores

Name

```
TOTAL_VCPU
```

Description

Total number of virtual CPU cores assigned to the virtual node.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro TOTAL_VCPU
```

Total Virtual RAM

Name

TOTAL_VMEM

Description

Total amount of virtual RAM assigned to the virtual node, in MB.

Usage Example

appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro TOTAL_VMEM

Get Cluster Configuration Choice

Name

```
CLUSTER_CONFIG_CHOICE
```

Description

Returns the value of a specific configuration choice key for any nodegroup in the virtual cluster. Each cluster can have multiple nodegroups.

Inputs

- **NGID:** Nodegroup ID. This is usually the nodegroup to which the current virtual node belongs, so `\$NODEGROUP` will be used.
- **KEY:** Configuration choice key for which to return the value. This is application-specific.

Outputs

This macro returns the following information:

- **Success:** Returns the value for the requested key.
- **Failure:** Nothing; exits the script with a non-zero status.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro CLUSTER_CONFIG_CHOICE
\$NODEGROUP <key>
```

Get Cluster Configuration Metadata

Name

```
CLUSTER_CONFIG_METADATA
```

Description

Returns the value of a specific configuration metadata key for any nodegroup within the virtual cluster.

Inputs

- **NGID:** Nodegroup ID. This is usually the nodegroup to which the current virtual node belongs, so `\$NODEGROUP` will be used
- **KEY:** Configuration choice key for which to return the value. This is application-specific.

Outputs

- **Success:** Returns the value for the requested key.
- **Failure:** Nothing; exits the script with a non-zero status.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro
CLUSTER_CONFIG_METADATA \$NODEGROUP <key>
```

Get Specific Configuration Choice Key

Name

```
NODEGROUP_CONFIG_CHOICE
```

Description

Returns the value of a specific configuration choice key for the nodegroup that the current virtual node belongs to.

Input

- **KEY:** Configuration choice key for which to return the value. This is application-specific.

Output

- Same as CLUSTER_CONFIG_CHOICE(). See Get Cluster Configuration Choice.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro
NODEGROUP_CONFIG_CHOICE <key>
```

Get Specific Metadata Choice Key

Name

```
NODEGROUP_CONFIG_METADATA
```

Description

Returns the value of a specific configuration metadata key for the nodegroup that the current virtual node belongs to.

Input

- **KEY:** Configuration choice key for which to return the value. This is application-specific

Output

- Same as `CLUSTER_CONFIG_METADATA()`. See Get Cluster Configuration Metadata.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro
NODEGROUP_CONFIG_METADATA <key>
```

Get a Unique Integer for This Host

Name

```
UNIQUE_SELF_NODE_INT
```

Description

Returns an integer for the current host. This number is unique across the nodegroup and is guaranteed to be between 1 and the number of hosts in the nodegroup. This function returns the same unique integer on a given virtual node across multiple invocations.

Input

nothing

Output

- **Success:** An integer is echoed as an output.
- **Failure:** Nothing is echoed from the function.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro UNIQUE_SELF_NODE_INT
```

Get a Unique Integer for Another Host

Name

```
UNIQUE_ANOTHER_NODE_INT
```

Description

Returns an integer for a specified remote host in the current nodegroup. This number is unique across the nodegroup and is guaranteed to be between 1 and the number of hosts in the nodegroup. This function returns the same unique integer on a given virtual node across multiple invocations.

Input

Hostname of the remote host.

Output

- **Success:** An integer is echoed as an output.
- **Failure:** Nothing is echoed from the function.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro
UNIQUE_ANOTHER_NODE_INT
```

Get Unique Integer by Service

Name

```
UNIQUE_INT_ID_BY_SRVC
```

Description

Returns a unique integer for a Catalog service based on an unspecified criterion. The integer generated is guaranteed to be between 1 and the number of hosts in the nodegroup that run the specified service. This function returns the same unique integer on a given virtual node across multiple invocations.

Input

- **SRVCID:** ID number of the Catalog service.

Output

- **Success:** An integer is echoed as an output.
- **Failure:** Nothing is echoed from the function.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro UNIQUE_INT_ID_BY_SRVC
```

Get Total Available VRAM

Name

```
GET_TOTAL_VMEMORY_MB
```

Description

Get the total amount of available virtual memory, in MB

Input

nothing

Output

- **Success:** Total amount of available VRAM, in MB.
- **Failure:** Nothing; exits when the return status of `bd_vcli` is non-zero.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro GET_TOTAL_VMEMORY_MB
```

Get Total Available VCPU Cores

Name

```
GET_TOTAL_VCORES
```

Description

Get the total number of virtual CPU cores available for Spark.

Input

nothing

Output

- **Success:** Total number of virtual CPU cores available to Spark.
- **Failure:** Nothing; exits when the return status of `bd_vcli` is non-zero.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro GET_TOTAL_VCORES
```

Get FQDN List

Name

```
GET_FQDN_LIST
```

Description

Get a list of Fully Qualified Domain Name(s) (FQDN) for the virtual node(s) with the specified role (e.g. Master, Worker, Edge).

Input

- Role (such as `master`, `worker`, etc.)

Output

- **Success:** FQDNs of the virtual node(s) in the cluster that have the specified role.
- **Failure:** Nothing; exits when the return status of `bd_vcli` is non-zero.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro GET_FQDN_LIST \$ROLE
<role>
```

Get IP Address List

Name

```
GET_IPADDR_LIST
```

Description

Get a list of IP addresses for the virtual node(s) with the specified role (e.g. Master, Worker, Edge).

Input

- Role (such as `master`, `worker`, etc.)

Output

- **Success:** IP addresses of the virtual node(s) in the cluster that have the specified role.
- **Failure:** Nothing; exits when the return status of `bd_vcli` is non-zero.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro GET_IPADDR_LIST \
$ROLE <role>
```

Get Virtual Node FQDN

Name

```
GET_NODE_FQDN
```

Description

Get the Fully Qualified Domain Name (FQDN) for the current virtual node.

Input

nothing

Output

- **Success:** FQDN of the current node.
- **Failure:** Exits when the return status of `bd_vcli` is non-zero.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro GET_NODE_FQDN
```

Get Virtual Node IP Address

Name

```
GET_NODE_IPADDR
```

Description

Get the IP address for the current virtual node.

Input

nothing

Output

- **Success:** IP address of the current virtual node.
- **Failure:** Returns a non-zero status.

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro GET_NODE_IPADDR
```

Generate Application-Specific URL

Name

```
GET_SERVICE_URL
```

Description

Generate a URL for a specific application.

Inputs

- **SRVC_ID:** ID number of the service for which to generate the URL.

- **ROLE:** Virtual node role on which the specified service runs (such as master or worker).
- **NODEGRP:** ID number of the nodegroup to which this URL applies. This disambiguates situations when the specified service is running in multiple nodegroups. If this argument is not specified and the same service name exists in multiple nodegroups, then the first nodegroup ID will be automatically used for this disambiguation.

Output

- **Success:** The application URL.
- **Failure:** Nothing; exits when the return status of `bd_vcli` is non-zero

Usage Example

```
appconfig autogen --pattern @@@@MYPATTERN@@@@ --macro GET_SERVICE_URL \
$SRV_ID <service_id> \$ROLE <role> \$NODEGRP <nodegroup_id>
```

Get Tenant Information

Name

```
TENANT_INFO
```

Description

Returns the value of the specified tenant namespace key.

Input

- **KEY:** Such as `aws_access_key` or `aws_secret_key`. See the list of valid tenant namespace keys in Application Configuration API on page 1050.

Output

- **Success:** Value of the specified tenant namespace key.
- **Failure:** Nothing; exits when the return status of `bd_vcli` is non-zero.

Usage Example

```
appconfig autogen --pattern @@@MYPATTERN@@@ --macro TENANT_INFO <key>
```

Sample Docker Files

This article presents the following sample Dockerfiles:

- Spark
- Datameer

These samples are for demonstration purposes only; the contents of a Dockerfile will vary greatly depending on the specific application and associated configuration requirements.

Sample 1: Spark

This example shows a Spark Dockerfile:

```
# Spark-1. docker image for RHEL/CentOS 6.x
FROM bluedata/centos7:latest
```

```
                        # Download and extract spark
                        RUN mkdir /usr/lib/spark; curl -s http://
archive.apache.org/dist/spark/spark-2.0.1/spark-2.0.1- bin-hadoop2.6.tgz |
tar xz -C /usr/lib/spark/

                        ## Install zeppelin
                        RUN mkdir /usr/lib/zeppelin;
curl -s https://s3.amazonaws.com/bluedata-catalog/thirdparty/ zeppelin/
zeppelin-0.7.0-SNAPSHOT.tar.gz | tar xz -C /usr/lib/zeppelin

                        ADD configure_spark_services.sh /root/
configure_spark_services.sh
                        RUN chmod -x /root/configure_spark_services.sh && /
root/configure_spark_services.sh
```

Sample 2: Datameer

This is a sample Dockerfile for the Datameer analytics platform.

```
                        # Datameer docker image
                        FROM bluedata/centos7:latest

                        RUN yum install -y http://
download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
                        RUN yum install -y expect mysql-server
mysql-connector-java

                        COPY cloudera-manager.repo /etc/yum.repos.d/
                        RUN yum -y install hadoop-client

                        RUN groupadd --system datameer
                        RUN useradd --system --create-home --gid datameer
datameer

                        ADD datameer-user.sh /root/
                        RUN /root/datameer-user.sh && rm /root/
datameer-user.sh

                        #COPY Datameer-6.1.14-cdh-5.9.0.zip /opt/datameer/
                        #RUN su - datameer && cd /opt/datameer && unzip -q
Datameer-6.1.14-cdh-5.9.0.zip && rm Datameer-6.1.14-cdh-5.9.0.zip

                        ## Install Datameer
                        RUN wget -q https://s3.amazonaws.com/bluedata-catalog/
thirdparty/datameer/Datameer-6.1.14-cdh-5.9.0.zip -P /opt/datameer/ &&
unzip -q /opt/datameer/Datameer-6.1.14-cdh-5.9.0.zip -d /opt/datameer/ &&
rm -rf /opt/datameer/Datameer-6.1.14-cdh-5.9.0.zip

                        RUN su - datameer && \
                            cd /opt/datameer && \
                            ln -s Datameer-6.1.14-cdh-5.9.0 current && \
                            cd current && \
                            mv logs/.donotdelete /var/log/datameer && \
                            rm -rf logs && \
                            ln -s /var/log/datameer logs
                        RUN chown -R datameer:datameer /opt/datameer/
```

Application Configuration API

This article describes the HPE Ezmeral Runtime Enterprise API that is used for automated application configuration. An application can use this API to query various configuration parameters and then include the responses as part of the configuration process that occurs when the application starts. The API framework consists of three primary components:

- **BlueData Agent:** This agent is installed in each of the container nodes in a cluster. The agent is configured when the cluster starts up and is completely transparent to the application and related configuration scripts.
- **Python API:** The `BD_VLIB` API includes the `BD_VCLI` command line utility provides a friendly shell script interface.
- **Application configuration bundle:** This is provided by a third-party developer who is developing a Catalog for HPE Ezmeral Runtime Enterprise.

This article describes the following topics:

- Application Configuration Bundle
- The BD_VLIB/BD_VCLI API

Application Configuration Bundle

An application configuration bundle may be either an uncompressed .tar file (.tar) or a gzipped .tar file (.tar.gz or .tgz). The bundle must contain two entry points implemented in either Python or bash, and must be named as follows:

- `startscript`: Must accept the following command line options:
  - `--configure`: Indicates that the application is being configured for the first time on this cluster. This option is invoked on all nodes that are deployed for the virtual cluster.
  - `--addnodes`: Indicates that new nodes were added to the virtual cluster. All nodes that existed before the new additions receive this notification, even if they belong to a different nodegroup. The following additional arguments are always specified to describe the new additions:
    - `--nodegroup`: Nodegroup ID of the added nodes.
    - `--role`: Role of the newly-added nodes in the above nodegroup.
    - `--fqdns`: Comma-separated list of FQDNs of the new nodes for quick identification.
  - `--delnodes`: Indicates that the nodes have been deleted from the virtual cluster. All remaining nodes receive this notification, even if they belong to a different nodegroup. The following additional arguments are always specified:
    - `--nodegroup`: Nodegroup ID of the deleted nodes.
    - `--role`: Role of the deleted nodes in the above nodegroup.
    - `--fqdns`: Comma-separated list of FQDNs of the deleted nodes for quick identification.
  - `--reattach`: Indicates that the cluster is being (re)attached to another cluster.
  - `--delete`: Invoked on all of the virtual nodes (containers) in a virtual cluster before cluster deletion. Also invoked when shrinking a virtual cluster on the virtual node(s) that are being deleted, as shown in the following table:

API calls for various cluster lifecycle events		
Virtual Cluster Lifecycle Event	**All Existing or Remaining Containers**	**New or Containers Being Deleted**
Create	N/A	All containers are new: startscript --configure

last-updated: May 06, 2024

Expand	startscript `--addnodes --nodegroup <NGID> --role <RID> --fqdns <NEW_FQDNS>`	On new containers: startscript `--configure`
Shrink	startscript `--delnodes --nodegroup <NGID> --role <RID> --fqdns <NEW_FQDNS>`	On containers being deleted: startscript `--configure`
Delete	startscript `-delete` *Only applies for* `config_api` *version 9 or higher.*	N/A

The BD_VLIB/BD_VCLI API

When using the `bd_vlib` library, developers may opt to use either a list of tokenized keys or a custom letter-delimited key. For simplicity, this article only describes the `bd_vcli` command line options.

- `--get=KEY`: Returns the value defined for the provided `KEY`. The available keys are divided into various namespaces, as described below.
  - **Platform namespace:** The keyword `platform` at the beginning of the key indicates the Platform namespace to the API. This namespace can only be accessed by a super user. It includes the following keys:
    - `platform.version`: Application configuration API version. Default is `8`.
    - `platform.plha`: Whether (`true`) or not (`false`) platform High Availability protection has been enabled.
    - `platform.clusterip`: IP address of the primary Controller host, regardless of the platform.plha value. The `platform.clustername` key will be empty if platform High Availability is not enabled, or if a DNS name was not specified when platform High Availability was enabled.
    - `platform.controllerip`: IP address of the original Controller host.
    - `platform.shadowip`: IP address of the Shadow Controller host, if platform High Availability is enabled.
    - `platform.arbiterip`: IP address of the Arbiter host, if platform High Availability is enabled.
    - `platform.controllerfqdn`: FQDN of the Controller host.
    - `platform.shadownfqdn`: FQDN of the Shadow Controller host, if platform High Availability is enabled.
    - `platform.aribterfqdn`: FQDN of the Arbiterhost, if platform High Availability is enabled.
    - `platform.clustername`: Name of the cluster name that was defined when enabling platform High Availability; otherwise, none.
    - `platform.isskudocker`: Whether (`true`) or not (`false`) HPE Ezmeral Runtime Enterprise is installed on-premises or hybrid.
    - `platform.isskuec2`: true/false: Whether (`true`) or not (`false`) HPE Ezmeral Runtime Enterprise is installed on EC2 only (not hybrid).
    - `platform.isskuk8s`: true/false: - Whether (`true`) or not (`false`) HPE Ezmeral Runtime Enterprise is installed on Kubernetes.

- **Tenant namespace:** The keyword `tenant` at the beginning of the key indicates the Tenant namespace to the API. This namespace can only be accessed by a super user. It includes the following keys:
  - `tenant.id`: ID of the tenant.
  - `tenant.name`: Name of the tenant.
  - `tenant.key_visibility`: Visibility of the tenant keypair. This will be either `all` (visible to everyone), `all_admins` (visible to the Tenant Administrator only), or `site_admin_only` (visible to the Platform Administrator only).
  - `tenant.user_groups`: Comma-separated list of user role names. Standard HPE Ezmeral Runtime Enterprise user roles are Admin and Member.
  - `tenant.user_groups.<role_name>`: Comma-separated list of indexes in to a list of DNs that can be used to log in to the role described by the `<role_name>`. The list is 0-indexed.
  - `tenant.user_groups.<role_name>.<list_index>`: An LDAP or Active Directory DN that defines a group. Users who are assigned to that group will be permitted to log in to the virtual node.
  - `tenant.kdc_type`: Kerberos protection configured for the tenant. This will be either `none` (if Kerberos is not enabled), or `MIT KDC` or `Active Directory` (if Kerberos is enabled).
- **Tenant namespace (with MIT KDC):** The following APIs apply if MIT KDC is configured for the tenant:
  - `tenant.kdc_host`: Name or IP address of the Kerberos host(s).
  - `tenant.kdc_realm:` Namespace that helps define access permissions.
  - `tenant.krb_enc_types`: Type of Kerberos encryption specified during tenant configuration.
  - `tenant.kdc_admin_user`: Username of the Kerberos administrator.
  - `tenant.kdc_admin_password`: Password of the Kerberos administrator.
- **Tenant namespace (with Active Directory KDC):** The following APIs apply if AD KDC is configured for the tenant:
  - `tenant.kdc_host`: Name or IP address of the Kerberos host(s).
  - `tenant.kdc_realm`: Namespace that helps define access permissions.
  - `tenant.krb_enc_types`: Type of Kerberos encryption specified during tenant configuration.
  - `tenant.kdc_admin_user`: Username of the Kerberos administrator.
  - `tenant.kdc_admin_password`: Password of the Kerberos administrator.
  - `tenant.kdc_ad_prefix`: Optional prefix to be added to all newly-created accounts.
  - `tenant.kdc_ad_suffix`: Active Directory suffix where all of the accounts used inside virtual clusters will be created.
  - `tenant.kdc_ad_ldaps_port`: Active Directory port for LDAPS.

- **Node namespace:** The keyword `node` at the beginning of the key indicates the Node namespace to the API. This namespace includes the following keys:
  - `node.hostname`: Hostname assigned to the node.
  - `node.domain`: Domain name of the node.
  - `node.fqdn`: Fully Qualified Domain Name (FQDN) of the host.
  - `node.role_id`: Role identifier of the node, as specified in the Catalog.
  - `node.distro_id`: Distro identification of the node as specified by the Catalog. This value may be empty.
  - `node.nodegroup_id`: Cluster-wide unique nodegroup identifier this node belongs to.
  - `node.depends_on`: Cluster-wide unique service IDs that this node depends on.
- **Cluster namespace:** The keyword `cluster` at the beginning of the key indicates the Cluster namespace to the API. This namespace includes the following keys:
  - `cluster.name`: Name assigned to the cluster by the user who created it.
  - `cluster.created_by_user_name`: The username associated with the user who created the cluster.
  - `cluster.service_tokens`: List of nodegroups ids where the services with auth token are running.
  - `cluster.service_tokens.<ng_id>`: List of service IDs in this namespace with authorization tokens.
  - `cluster.service_tokens.<ng_id>.<srvc_id>.auth_token`: The authorization token associated with the service.
  - `cluster.config_choice_selections`: List of all nodegroup IDs that are part of the virtual cluster. This information is based on what is specified in the Catalog and specific selections made by the user at cluster creation time.
  - `cluster.config_choice_selections.<ng_id>`: Keys that may be available for the specific node group.
  - `cluster.config_choice_selections.<ng_id>.<key>`: Value assigned to the specific key.
  - `cluster.config_metadata`: A list of all nodegroup IDs that are part of the virtual cluster.
  - `cluster.config_metadata.<ng_id>`: List of all configuration metadata keys available for the particular nodegroup. The available keys are gathered from the information specified in the Catalog of this distribution.
  - `cluster.config_metadata.<ng_id>.<key>`: Value assigned to the specific key.
- **Distros namespace:** The keyword `distros` at the beginning of the key indicates the Distros namespace to the API. This namespace includes the following keys:
  - `distros`: List of Catalog IDs of the various distributions deployed in the virtual cluster.
  - `distros.<d_id>`: List of all nodegroup IDs that use the specific distribution.

  - `distros.<d_id>.<ng_id>`: List of keys available for this specific distribution deployed in the given nodegroup.
  - `distros.<d_id>.<ng_id>.distro_id`: The Catalog distribution ID for this distribution.
  - `distros.<d_id>.<ng_id>.catalog_entry_version`: Version defined in the Catalog describing the distribution.
  - `distros.<d_id>.<ng_id>.config_metadata`: Configuration metadata declared in the Catalog.
  - `distros.<d_id>.<ng_id>.config_choice_selections`: Configuration choices made by the user when creating the cluster.
  - `distro.<d_id>.<ng_id>.roles`: List of all the roles deployed in the virtual cluster that use this distribution ID and belong to the specified nodegroup.
- **Services namespace:** The keyword `services` at the beginning of the key indicates the Services namespace to the API. This namespace includes the following keys:
  - `services`: List of all cluster-wide services. Services are only listed here once, even if a particular service will be configured on multiple nodes.
  - `services.<s_id>`: List of all nodegroup IDs on which this particular service is expected to be available.
  - `services.<s_id>.<ng_id>`: List of all role IDs in the specified `<ng_id>` that is/are expected to run the service specified by `<s_id>`.
  - `services.<s_id>.<ng_id>.<r_id>`: List of keys available for this particular service on the specified nodegroup and role.
  - `services.<s_id>.<ng_id>.<r_id>.fqdns`: FQDNs of all nodes that will be running the service.
  - `services.<s_id>.<ng_id>.<r_id>.hostnames`: Hostnames of all nodes that will be running the service.
  - `services.<s_id>.<ng_id>.<r_id>.endpoints`: List of URIs to the selected service.
  - `services.<s_id>.<ng_id>.<r_id>.qualifiers`: List of all qualifiers associated with this service. This information is gathered from the Catalog.
- **Attachments namespace:** This namespace is available when an AI/ML cluster is attached to another cluster. Otherwise, it will be empty.
  - `attachments`: `clusters` and `models` are the only available sub-keys.
  - `attachments.clusters`: List of all cluster IDs attached to the current cluster.
  - `attachments.clusters.<cluster_id>`: List of all keys describing the cluster.
  - `attachments.clusters.<cluster_id>.id`: The attached cluster's ID.
  - `attachments.clusters.<cluster_id>.name`: The attached cluster's name.
  - `attachments.clusters.<cluster_id>.config_metadata`: The attached cluster's config metadata when it was deployed. Please refer to `cluster.config_metadata` for sub-keys.

    - `attachments.clusters.<cluster_id>.config_choice_selections`: The attached cluster's selected configuration choices when it was deployed. Please refer to `cluster.config_choice_selections` for sub-keys.
    - `attachments.clusters.<cluster_id>.isolated`: Boolean describing whether the attached cluster is in Isolated mode.
    - `attachments.clusters.<cluster_id>.services`: The Services namespace of the attached cluster. Please refer to the **Services namespace** for sub-keys.
    - `attachments.clusters.<cluster_id>.distros`: The Distro namespace of the attached cluster. Please refer to the **Distros namespace** for sub-keys.
    - `attachments.models`: Comma separated list of IDs of all models attached to this cluster.
    - `attachments.models.<model_id>`: Comma-separated list of keys available to query.
    - `attachments.models.<model_id>.name`: The name given to the model.
    - `attachments.models.<model_id>.version`: The model's versions.
    - `attachments.models.<model_id>.status`: The current status of the model.
    - `attachments.models.<model_id>.created_by_user_id`: The ID of the user who created the model.
    - `attachments.models.<model_id>.create_by_user_name`: The username associated with the creator of the model.
    - `attachments.models.<model_id>.model_location`: The model's location in the project repository.
    - `attachments.models.<model_id>.scoring_script`: The path to the model's scoring script in the project repository.
    - `attachments.models.<model_id>. input_parameter_file`: Input parameter specification file for the model.
    - `attachments.models.<model_id>. inference_bootstrap_script`: Bootstrap script specified when the model was created.
    - `attachments.models.<model_id>. training_data`: Pointer to the training data.
    - `attachments.models.<model_id>. training_cluster`: Training cluster used when creating the model.

- **Distributed synchronization:** Application configuration on different nodes with different roles may depend on each other and may have to wait for one to come up before the other. This API provides the ability to wait for a service to be registered and running before proceeding with the configuration.
    - `--wait=<KEY(S)>`: Comma-separated list of key(s) specifying which services to wait for. The key(s) specified could be either the end services or a single key representing a group of services, but not both at the same time.
    - `--timeout=SEC`: Maximum time to wait for a response from the remote node running the service specified to `--wait` options. If not specified, this timeout defaults to one hour.

- **Service registration:** All services defined in the Catalog must be registered with the application configuration framework at the time of initial configuration. The registration process also starts the service at both create time and after a node reboot. The registration also provides service lifecycle management facilities. The following API is available for registering catalog services:
  - `--service_key=KEY`: The key from the virtual cluster metadata querying API that uniquely identifies the specific service being registered.
  - `--sysv=SERVICE`: Node-wide unique system service name to register, such as `sssd`.
  - `--desc=DESCRIPTION`: Description for the system service being registered.
- **Miscellaneous:** The API also includes the following miscellaneous functions:
  - `--version`: Current version of the metadata representation. Configuration scripts may use this information as deemed necessary; however, HPE Ezmeral Runtime Enterprise does not currently support upgrading guest configuration bundles on existing virtual clusters.
  - `--get_local_group_fqdns`: Returns a comma-separated list of FQDNs of all the nodes deployed in the same nodegroup as the requesting node.
  - `--get_all_fqdns`: Returns a comma-separated list of FQDNs of all the nodes deployed in the cluster.
  - `--restart_all_services`: Restarts all services registered with the application configuration framework. This may be used to restart all registered services when the parameters were modified after successful cluster creation. Alternatively, the application configuration scripts may choose to restart individual services as required.
  - `--cp --node <fqdn> --src <local_path> --dest <remote_path> --perms <remote_permissions>`: Copies a file from the node invoking this API to another specified by the FQDN. For security, the destination file will be owned by the same user that initiated the API on the source node. In this command:
    - `<fqdn>` is the fully qualified domain name of the source container.
    - `<local_path>` is the full path to the file to be copied.
    - `<remote_path>` is the full path to the location where the file will be copied.
    - `<remote_permissions>` are the permissions to assign to the file on the destination, in absolute (octal) notation. If needed, you may access a permissions calculator that will help you determine the proper permissions to assign to the copied file (link opens an external website in a new browser tab/window).
  - `--execute --remote_node <fqdn> --script <remote_path>`: Executes a script on a remote Docker container (virtual node), where:
    - `<fqdn>` is the fully qualified domain name of the source container.
    - `<remote_path>` is the full path to the script that will be executed. This script must already exist in the specified path before executing this command.

Metadata JSON

Each **App Store** image includes metadata contained inside a JSON file that specifies various interface and configuration options. Some of this metadata is visible in the **App Store** screen and/or the **Create Cluster** and **Create New Job** screens, and is described in the Interface Metadata section of this article. The

interface metadata, along with other application configuration metadata, is contained inside the Catalog JSON file that is described in the Catalog JSON File section of this article.

Interface Metadata

The interface-related metadata included for use by the **App Store** interface consists of the following information:

- **Basic image information:** This information is visible in the **App Store** screen and includes:
  - **App name:** Name of the application.
  - **Description:** Short description of the application.
  - **Logo:** Image file that displays in the **App Store** screen.
- Hovering the mouse over an application tile expands the tile to display the following additional information:
  - **Long Description:** Longer description of the application.
  - **Version:** Image version and optional build number.
  - **Root disk size (Local):** Root disk size required for running the image on-premises.
  - **Root disk size (EC2):** Root disk size required for running the image on an EC2 instance.
  - **Distro ID:** Unique identifier for the image.
  - **Category:** Category of Big Data application provided by the image.
- Additional metadata determines the options that will be available in the **Create New Cluster** screen when a new cluster is created. This includes:
  - **Cluster group name:** Type(s) of cluster (such as Hadoop, Spark, and/or Kafka) on which the image can run.
  - **Node flavor limits:** Role type specific node flavor(s) required to run the application, which will be based on the CPU, RAM, and storage requirements of the application.
  - **Node count limits:** Number of role-specific and/or Edge nodes required to run the application.

Catalog JSON File

This article uses the **CDH 5.4.3 with Cloudera Manager** Catalog entry as an example for explaining the HPE Ezmeral Runtime Enterprise Catalog (**App Store**) entry JSON properties. The `cdh54CM.json` file is located in the `/opt/bluedata/catalog/entries/system` directory.

NOTE: This article describes Version 1 of the catalog JSON. This version is still supported; however, you may want to use later versions for authoring new Catalog entries. Version 2 (or later) will be required for any entry that makes use of a later version of the vAgent config API, and Version 3 (or later) will be required if you are supplying a custom logo.

Catalog entry properties can be broadly segregated into the following purposes:

- Identification
- Components
- Services
- Node Roles

- Configuration
  - Selected Roles
  - Node Services
  - Config Metadata
  - Config Choices

Identification

The `identification` blob appears as follows:

```
                          "distro_id": "cdh54CM",
                          "label": {
                            "name": "CDH 5.4.3 with Cloudera Manager",
                            "description": "CDH 5.4.3 with MRv1/YARN and HBase
support. Includes Pig, Hive, Hue and Spark."
                            },
                          "version": "2.0.1",
                          "epic_compatible_versions": ["3.4"],
                          "categories": [ "Hadoop", "HBase" ],
```

In this blob:

- `distro_id` is unique identifier for either a Catalog entry or a versioned set of Catalog entries. It represents a particular application or application-framework setup as created and maintained by a particular author or organization. The HPE Ezmeral Runtime Enterprise interface and API currently only allow only one Catalog entry with a given distro ID to be installed for use at any given time. Each distro ID corresponds to one "tile" in the **Images** tab of the **App Store** screen. HPE Ezmeral Runtime Enterprise may also reference the distro ID when determining appropriate Add-On image entries that can be added to a cluster, because an add-on may have a distro ID requirement.
- The `label` property contains the following parameters:
  - `name`, which is the "short name" of the Catalog entry. The Catalog API does not allow entries with different distro IDs to share the same name.
  - `description`, which is a longer, more detailed blurb about the entry.
- `version` is a discriminator between multiple Catalog entries that share the same distro ID. It is expected to adhere to a simple pattern of digits separated by dots in the format version a.b.c, where:
  - `a.b` is the version number, such as the `2.0` in `"version": "2.0.1"`. You may assign any version you want to the Catalog entry, and each Catalog entry will have its own unique distro ID . This version represents iterations of this Catalog entry; it does not necessarily represent the version of any software deployed in a cluster. For example, you may have a CDH 5.4 Catalog entry that you deploy as Version 1.0 followed by 1.1, 2.0, etc. HPE Ezmeral Runtime Enterprise installs the newest available version of a given distro ID when instructed to install or upgrade that distro ID.

  - c is the optional build number, such as the `1` in `"version": "2.0.1"`. App Workbench stores the first value used for c when the distro ID is created. Future versions of the same distro ID will automatically increment the build number based on the last value stored in the system, provided that you do not change the c value in the JSON file. In this example, the first-ever build of the same distro ID will be version 2.0.1, the next version will be 2.0.2, and so forth. Manually entering a new build number that is equal to or less than the stored build value will not have any effect until you change the version number by modifying the a and/or b values, such as by moving from version 2.0.1 to version 2.1.1 or 3.0.1. Manually entering a new build number that is higher than the stored build value will increment the build number to the new value. For example, if the stored build value is 5 and you enter a build number that is less than or equal to 5, then the next build number will be 6; however, if the stored build value is 5 and you enter a build number of 10, then the next build number will be 10 and will increment from there.

- `epic_compatible_versions` lists the HPE Ezmeral Runtime Enterprise versions where this Catalog entry may be used. An asterisk (*) may be used in a version string as a wildcard.

- `categories` is a list of strings used by the HPE Ezmeral Runtime Enterprise interface to group Catalog entries during cluster creation. These values appear in the **Select Cluster Type** pull-down menu.

Components

The `components` blob appears as follows:

```
                          "image": {
                            "checksum": "b07e8cfea8a9c1a6cdc6990b1da29b9f",
                            "import_url": "http://s3.amazonaws.com/
bluedata-vmimages/Cloudera-CDH-CM-5.4.3-v2.tgz"
                          },
                          "setup_package": {
                            "checksum": "7560c8841c1400e0e4a4ba3dac1ba8d7",
                            "import_url": "http://s3.amazonaws.com/
bluedata-vmimages/cdh5-cm-setup.tgz"
                          },
```

In this blob:

- `image` is a property that identifies the location for the image used to launch virtual nodes for this Catalog entry. In HPE Ezmeral Runtime Enterprise (EPIC) versions 2.0 and above, this will be an image for launching a Docker container. This location can be specified in either of two ways:

  - `import_url`, which is the http (not https) URL from which the image can be downloaded. This must be accompanied by the `checksum`, which is the MD5 checksum of the image. This method is used for normal Catalog entry distribution. The image will be downloaded into the images download cache directory when the entry is installed, and the downloaded image may be automatically deleted in certain garbage-collection situations when the Catalog entry is not in use and not present in any Catalog feed.

  - `source_file/opt/bluedata/catalog/images/`). Only the file system is necessary, not the complete path. No checksum is provided in this case. This method is used for either development or site-local entries. In this case, HPE Ezmeral Runtime Enterprise will never automatically download the designated image file.

- `setup_package` is similar to the image property except for the configuration scripts package that runs inside the launched virtual node. In this case, the download cache directory is `/opt/bluedata/catalog/guestconfig`.

Services

The `services` blob appears as follows:

```
"services": [
  {
    "id": "hbase_master",
    "exported_service": "hbase",
    "label": {
      "name": "HMaster"
      },
    "endpoint" : {
      "url_scheme" : "http",
      "port" : "60010",
      "path" : "/",
      "is_dashboard" : true
      }
    },
    {
      "id": "hbase_worker",
      "label": {
        "name": "HRegionServer"
        },
      "endpoint" : {
        "url_scheme" : "http",
        "port" : "60030",
        "path" : "/",
        "is_dashboard" : true
      }
      },
    {
      "id": "hbase_thrift",
      "label": {
        "name": "HBase Thrift service."
        }
      },
    ...
],
```

In this example, `services` is a list of service objects. The defined services will be referenced by other elements of this JSON file to determine which services are active on which nodes within the cluster. That information will then be used to:

- Present clickable **Dashboard** links in the HPE Ezmeral Runtime Enterprise interface.
- Determine which dependent nodegroups (Add-On Images) can be attached to the cluster.
- Trigger NAT port mapping for the service, if appropriate.
- Optionally be referenced by the setup scripts that run within the virtual node.

Setup scripts also use service identifiers to register those services with vAgent, so that necessary services can be properly started and restarted along with the virtual node. Setup scripts can also choose to wait for a vAgent-registered service to be active on a node in order to coordinate multi-node setup across the cluster.

NOTE: The "service" terminology does not correspond to a definition of "service" that is specific to some particular application or application framework. A "service" is any entity that can be used for any of the purposes described above. For example, a YARN resource manager is a service, as is `sshd`.

In this blob:

- `id` is an identifier that must be unique within the scope of this JSON file. It is used by other objects in this file to reference this service. It is also used in the setup scripts when composing a key for registering a service with vAgent, or when waiting on a registered service to start.
- `exported_service` is an optional property that has an agreed-by-convention value for a service that is referenced from outside the cluster. This property can have an optional qualifiers list of descriptive qualifiers for that exported service, again with agreed-by-convention values. qualifiers may only be defined if `exported_service` is defined.

NOTE: The above values are currently only used when determining appropriate Add-On Image entries that can be added to a cluster, because those entries may have a requirement that the cluster provides specific exported services, or even exported services with specific qualifiers. For example, an add-on may have a dependence on the Hadoop exported service, or a more specific dependence on Hadoop with the YARN qualifier.

- `label` uses the same format as the entry's label:
  - `name`, which briefly describes the service. This property is currently used only when composing clickable service-dashboard links in the HPE Ezmeral Runtime Enterprise interface; however, it is required for all services.
  - `description`, which is an optional description property with more details.
- `endpoint` describes the network endpoint of the service.
  - "`auth_token": true|false`: Whether (true) or not (false) the endpoint requires an authentication token.
  - `is_dashboard` is a Boolean property of the endpoint that indicates whether this is a URL that can (and should) be viewed from a web browser, such as in the HPE Ezmeral Runtime Enterprise interface.
  - The `url_scheme`, `port`, and `path` properties of this object are used to compose a service URL. These properties have the following constraints:
    - `url_scheme` must be defined if `is_dashboard` is `true`.
    - `port` must be defined.
    - `path` is optional.

NOTE: The presence of an `endpoint` object triggers the creation of a NAT port mapping for this service, if HPE Ezmeral Runtime Enterprise is running inside an EC2 instance.

Node Roles

The `node_roles` blob appears as follows:

```
"node_roles": [
  {
    "id": "controller",
    "cardinality": "1",
    "anti_affinity_group_id": "CM",
    "min_cores": "4",
    "min_memory": "12288"
  },
  {
    "id": "standby",
    "cardinality": "1",
```

```
                    "anti_affinity_group_id": "CM"
                  },
                  {
                    "id": "arbiter",
                    "cardinality": "1",
                    "anti_affinity_group_id": "CM"
                  },
                  {
                    "id": "worker",
                    "cardinality": "1+"
                  }
                ],
```

In this example, `node_roles` is a list of objects describing roles that may be deployed for this Catalog entry. Each role is a particular configuration instantiated from the entry's virtual node image and configured by the setup scripts. The configuration associated with a particular role is broadly left up to the setup scripts, and thus varies widely from entry to entry; however, there are certain constraints and semantics associated with specific roles in the current HPE Ezmeral Runtime Enterprise release (for non-Add-On entries):

- The allowed roles are `controller`, `worker`, `standby`, and `arbiter`. If applicable, these roles will be created using the **Master Node Flavor** specified in the HPE Ezmeral Runtime Enterprise interface when you create the cluster.
- To support job submission to a cluster from the HPE Ezmeral Runtime Enterprise interface, the cluster must include a `controller-role` node. If the cluster also includes a `standby-role` node, then that standby will be tried as an alternate target for job submission if the Controller node is unresponsive.
- Worker role nodes (if applicable) will be created using the **Worker Node Flavor** specified in the HPE Ezmeral Runtime Enterprise interface when you create the cluster.
- Only the `worker` role is allowed to have scale-out cardinality (see below); the worker role MUST have scale-out cardinality.
- The **Worker Count** in the cluster creation interface covers the total number of `worker`, `standby`, and `arbiter` nodes. Cluster expansion will increase the number of `worker` nodes.

The properties of each role object are:

- `id` is an identifier that must be unique within the scope of this JSON file. It is used by other objects in this file to reference this role. It is also used by HPE Ezmeral Runtime Enterprise as described above, and may also be referenced by the setup scripts.
- `cardinality` describes the number of nodes in this role that will be deployed, if/when this role is selected to be used in a cluster. If the `cardinality` string just consists of an integer, then a fixed number of nodes will be deployed for this role. If the `cardinality` string is an integer followed by `+`, then a variable number of nodes may be deployed in this role. The integer is the minimum number. This kind of value is referred to as a "scale-out" cardinality.
- `anti_affinity_group_id`, if it has a specified value, causes nodes deployed from this role and/or from any other role with the same `anti_affinity_group_id` to be placed on different physical hosts. If this constraint cannot be satisfied, then the cluster creation/expansion will be rejected.

Anti-affinity is typically used to reduce the physical resources shared by a set of nodes, to make it less likely for a single physical fault to affect them all. This constraint only applies to nodes within a given cluster; anti-affinity is not enforced among nodes from different clusters.

- `min_cores` is an optional property that specifies a minimum number of virtual cores that must be provided in the flavor used to deploy this role.

- `min_memory` is an optional property that specifies a minimum memory size that must be met by the flavor used to deploy this role.

Configuration

The `configuration` blob appears as follows:

```
"config": {
  "selected_roles": [
     ...
    ],
  "node_services": [
     ...
    ],
  "config_meta": [
     ...
    ],
  "config_choices": [
     ...
    ],
```

The remainder of the JSON file describes which node roles will be deployed into the cluster, and which services will be present on any node with a given role. This information may depend on choices provided by the UI/API user when they are creating the cluster.

- `selected_roles` lists IDs of roles that will be deployed.
- `node_services` lists IDs of services that will be present on nodes of a given role, if that role is deployed.
- `config_meta` lists of string key/value pairs that can be referenced by the setup scripts.
- `config_choices` lists both the choices available to the UI/API user and the possible selections for each choice. This is a potentially recursive data structure in that a selection may include another config object, which in turn may contain `selected_roles`/`node_services`/`config_meta`/`config_choices` properties.

This structure means that the top-level `selected_roles`, `node_services`, and `config_meta` `property` values will apply regardless of any user-provided input about choice selections. User-provided input may then have consequences such as activating additional roles and/or services in the cluster, and/or adding more elements to the `config_meta`

For example, in the CDH 5.4.3 JSON:

- There is a top-level `mrtypemrv1` and `yarn`.
- If `yarn` is selected for the `mrtype` choice, then:
  - The `controller` and `worker,` roles are selected for deployment.
  - The `yarn_rm` and `job_history_server` services are selected to be present on the `controller` role node.
  - The `yarn_nm` service is selected to be present on the `worker` role nodes.
  - The `yarn_nm` service is also selected to be present on the `standby` and `arbiter`
  - The `yarn_ha` options is enabled, with valid selections true or false. If `true` is selected for `yarn_ha`, then:
    - The `controller`, `standby`, `arbiter`, and `worker` roles must be defined.

- The `zookeeper` service is selected to be present on the `controller`, `standby`, and `arbiter` role nodes.
- The `yarn_rm` and `hdfs_rm` services are selected to be present on the `standby` role node.

Selected Roles

The `selected_role` blob appears as follows:

```
                          "selected_roles": [
                            "controller",
                            "standby",
                            "arbiter",
                            "worker"
                          ],
```

The value of the `selected_roles` property is a list of role IDs. The example shown above is taken from the choice selection that activates HBase support.

NOTE: In this particular Catalog entry, the top-level `selected_roles` property is an empty list; no roles at all will be activated unless the user provides some input (choice selections). This is a valid arrangement and reflects the fact that, for this Catalog entry, some choices must be made before any usable application framework can be provided in this cluster. By contrast, some other Catalog entries have roles and services that are always selected.

Node Services

The `node_services` blob appears as follows:

```
                          "node_services": [
                            {
                              "role_id": "controller",
                              "service_ids": [ "ganglia", "ganglia_api", "ssh",
"gmetad", "gmond", "httpd" ]
                            },
                            {
                              "role_id": "standby",
                              "service_ids": [ "ssh", "gmond" ]
                            },
                            {
                              "role_id": "arbiter",
                              "service_ids": [ "ssh", "gmond" ]
                            },
                            {
                              "role_id": "worker",
                              "service_ids": [ "ssh", "gmond" ]
                            }
                          ],
```

Each element of this list is a `node_services` object that describes the services available on a given role. The role may or may not be selected; this data structure simply indicates that if a certain role is selected (according to choice selections), then these are the services a node with that role will provide. The top-level `node_services` in this example Catalog entry are all of the ancillary services that don't depend on choices like HBase support or MR type.

The properties of each `node_services` object are:

- `role_id` references the value of the `id` property of a `node_role` object defined within this same catalog entry JSON.
- `service_ids` is a list of id values of service objects defined within this same Catalog entry JSON.

Config Metadata

The `config_metadata` appears as follows:

```
                        "config_meta": {
                          "streaming_jar": "/opt/cloudera/parcels/CDH/lib/
hadoop-mapreduce/hadoop-streaming.jar",
                          "impala_jar_version": "0.1-SNAPSHOT",
                          "cdh_major_version": "CDH5",
                          "cdh_full_version": "5.4.3",
                          "cdh_parcel_version": "5.4.3-1.cdh5.4.3.p0.6",
                          "cdh_parcel_repo": "http://archive.cloudera.com/
cdh5/parcels/5.4.3"
                        },
```

In this example, `config_meta` is a key-value store. These values are only used by the scripts in the guest package and are thus completely opaque to HPE Ezmeral Runtime Enterprise. These values may be referenced during node setup. For example, the `streaming_jar` value is conventionally referenced by the script that runs Hadoop Streaming jobs.

Choice selections may cause the definition of multiple `config_meta` lists that together form the KV store visible to the in-guest scripts. To avoid confusion, key conflicts are not allowed. For example, it is legal for mutually exclusive choice selections to define different values for a key, but it is not legal for the same key to be defined more than once when composing the KV store that results from a particular set of choice selections.

Config Choices

This `config_choices` blob appears as follows:

```
                        "config_choices": [
                          {
                            "id": "hbase",
                            "type": "boolean",
                            "label": {
                              "name": "HBase"
                            },
                            "selections": [
                              {
                                "id": false
                              },
                              {
                                "id": true,
                                "config": {
                                  ...
                                }
                              }
                            ]
                          },
                          {
                            "id": "mrtype",
                            "type": "multi",
                            "label": {
                              "name": "MR Type"
                            },
                            "selections": [
                              {
                                "id": "mrv1",
                                "label": {
                                  "name": "MRv1"
                                },
                                "config": {
```

```
                                    ...
                                  }
                                },
                                {
                                  "id": "yarn",
                                  "label": {
                                    "name": "YARN"
                                  },
                                  "preferred": true,
                                  "config": {
                                    "selected_roles": [
                                      "controller",
                                      "worker"
                                    ],
                                    "node_services": [
                                      ...
                                    ],
                                    "config_choices": [
                                      {
                                        "id": "yarn_ha",
                                        "type": "boolean",
                                        "label": {
                                          "name": "YARN and HDFS High
Availability"
                                        },

                                        "selections": [
                                          {
                                            "id": false
                                          },
                                          {
                                            "id": true,
                                            "config": {
                                              ...
                                            }
                                          }
                                        ]
                                      }
                                    ],
                                    "config_choices":[
                                      {
                                        "label": {
                                          "name": "CLouderaManagerServer"
                                        },
                                        "type": "string",
                                        "id": "clouderamanager-server"
                                      }
                                    ]
                                  }
                                }
                              ]
                            },
                          }
```

This blob lists the choices available to the API/UI user when creating a cluster. Each choice has some number of valid selections (either Boolean or multiple-choice) that can be provided to satisfy that choice. A given selection can then contain a nested `config`, as described previously.

In this example, one choice describes whether or not to activate HBase support. Another describes the choice between using MRv1 or YARN. If YARN is selected, then there is a further choice as to whether to activate cluster High A.

Each of these choices activates certain roles for deployment and selects certain services to be present on nodes of given roles.

This structure is fairly generic; however, HPE Ezmeral Runtime Enterprise constrains the choices to those currently defined among the various Catalog entries provided as part of the HPE Ezmeral Runtime Enterprise release. Please contact Hewlett Packard Enterprise support if you wish to define choices in a Catalog entry that you are authoring.

The properties of each choice object are:

- `id` is a choice identifier. It can be referenced by the setup scripts (which can see all choice selections made for cluster creation). Each selection object must contain an `id` property that is the selection value. The possible values for this property are limited to the set of choices present in the Catalog provided with the HPE Ezmeral Runtime Enterprise release.
- `type` describes the selection value type. This property may have one of the following values:
  - `boolean`: Selection values are either `true` or `false`. This selection type does not require a label.
  - `multi`: Selection values are a defined set of strings. This selection type must have a `label` object that describes the selection. This object includes a required `name` and an optional `description`, which will be used by future HPE Ezmeral Runtime Enterprise versions to drive various interface behaviors.
  - string: Alphanumeric characters.
- `selections` lists the valid selections for this choice. A selection may include an optional `preferred` property. If this is set to `true`, the HPE Ezmeral Runtime Enterprise interface will default to this selection value when presenting the choice. A selection may contain an optional nested `config` object that describes the configuration activated by the selection.

Upgrading an Existing Image

This article describes how to update an existing application `.bin` file. This example upgrades an existing CDH 5.7 image to CDH 5.9. This upgrade process does not cover any additional modifications to the image, such as adding services or modifying startscripts. To upgrade an existing image:

1. Create a new directory to house the image you are going to create, such as `/source/<image name>`, where `/source/<image name>` is the name of the new image you are going to create. This example uses `/source/cdh59`.
2. Copy the `.bin` for the existing image to the directory that you created in Step 1.
3. Navigate to the `/opt/bluedata/catalog/bundles` directory. This directory contains two subdirectories:
   - `/download`
   - `/install`.

4. If the image you want to modify has already been installed in your **App Store**, then go to the `/install` directory. If the image is available in your **App Store** but has not been installed, then go to the `/download` directory.

```
                        [root@yav-100 ~]# cd opt/bluedata/catalog/bundles
                        [root@yav-100 bundles]# ls
                        download install
                        [root@yav-100 bundles]
                        [root@yav-100 downloads]# ls
                        bdcatalog-centos-bluedata-cdh551-1.6.bin
bdcatalog-centos-bluedata-hdp25-ambari-2.0.bin
                        bdcatalog-centos-bluedata-cdh551-edge-1.3.bin
bdcatalog-centos-bluedata-hdp-edge25-2.3.bin
                        bdcatalog-centos-bluedata-cdh57-2.1.bin
bdcatalog-centos-bluedata-mapr510-2.3.bin
                        bdcatalog-centos-bluedata-cdh57-edge-1.1.bin
bdcatalog-centos-bluedata-spark15-2.1.bin
                        bdcatalog-centos-bluedata-hdp23-ambari-1.4.bin
bdcatalog-centos-bluedata-spark16-1.8.bin
                        bdcatalog-centos-bluedata-hdp24-edge-1.0.bin
bdcatalog-centos-bluedata-spark201-1.2.bin
                        bdcatalog-centos-bluedata-hdp24-ambari-1.7.bin
bdcatalog-centos-bluedata-spark201-edge-2.3.bin
```

5. Identify the `.bin` file for your image, and then copy it to the parent `source` directory that you created in Step 1. This may take several minutes if the image was not installed in your **App Store**, depending on the file size and transfer speed.

6. Make the `.bin` file executable by executing the command `chmod +x <bin_name>`, where `<bin_name>` is the name of the `.bin` file, such as `chmod +x bdcatalog-centos-bluedata-cdh57-2.1.bin`.

7. Unpack the `.bin` file by executing the command `./<bin-name> --payload`. This generates two files in the `source` directory:

   - `payload.tar`
   - `decompress.sh`.

```
                        [root@yav-100 source]# ./
bdcatalog-centos-bluedata-cdh57-2.1.bin --payload
                        [root@yav-100 source]# ls
                        bdcatalog-centos-bluedata-cdh57-2.1.bin   cdh59
decompress.sh   payload.tar
```

8. Untar the `payload.tar` file by executing the command `tar xvf payload.tar`. This creates a new directory, such as `bdcatalog-centos-bluedata-cdh57-2.1`.

```
                        [root@yav-100 source]# tar xvf payload.tar
                        bdcatalog-centos-bluedata-cdh57-2.1/
                        bdcatalog-centos-bluedata-cdh57-2.1/
bdcatalog-centos-bluedata-cdh57-2.1-src.tgz
                        bdcatalog-centos-bluedata-cdh57-2.1/
bluedata-cdh57-centos-2.0.tar.gz
                        bdcatalog-centos-bluedata-cdh57-2.1/
cdh5-cm-setup.tgz
                        bdcatalog-centos-bluedata-cdh57-2.1/
cdh57CM-centos.json
                        [root@yav-100 source]# ls
                        bdcatalog-centos-bluedata-cdh57-2.1
bdcatalog-centos-bluedata-cdh57-2.1.bin
                        cdh59   decompress.sh   payload.tar
```

9. Navigate to the new directory, and untar the `<bin name>-src.tgz` file by executing the command `tar xvzf bdcatalog-centos-bluedata-cdh57-2.1-src.tgz` to access the source directory of the `.bin` file.

```
                        [root@yav-100 source]# cd
bdcatalog-centos-bluedata-cdh57-2.1
                        [root@yav-100
bdcatalog-centos-bluedata-cdh57-2.1]# ls
                        bdcatalog-centos-bluedata-cdh57-2.1-src
bdcatalog-centos-bluedata-cdh57-2.1-src.tgz
                        bdcatalog-cdh57-centos-2.0.tar.gz
cdh57CM-centos.json   cdh5-cm-setup.tgz
                        [root@yav-100
bdcatalog-centos-bluedata-cdh57-2.1]# cd
bdcatalog-centos-bluedata-cdh57-2.1-src [root@yav-100
bdcatalog-centos-bluedata-cdh57-2.1-src]# ls
                        cdh57CM-centos.json   cdh57cm.wb   cdh5-cm-setup
image   Logo_Cloudera.png
```

10. Copy the contents of the source directory to the new image directory that you created in Step 1. The directory should appear as shown below after the copy is complete:

```
                        [root@yav-100 source]# cd cdh59
                        [root@yav-100 cdh59]# ls
                        cdh57CM-centos.json   cdh57cm.wb   cdh5-cm-setup
image   Logo_Cloudera.png
```

11. If desired, you may clean up your source directory by removing the following items so that only the image directory remains:
    - `payload.tar`
    - `decompress.sh`
    - The original `.bin` file, such as `bdcatalog-centos-bluedata-cdh57-2.1.bin`.
    - The unpacked `.bin` file, such as `bdcatalog-centos-bluedata-cdh57-2.1`.

12. Change the JSON and `.wb` file names to reflect the new image version. You do not need to change the `logo.png` file, nor the image-setup directory (such as `version.cdh5-cm-setup`) in order to upgrade the image.

```
                        [root@yav-100 cdh59]# ls
                        cdh59CM-centos.json   cdh59cm.wb   cdh5-cm-setup
image   Logo_Cloudera.png
```

13. Modify the `.wb` file,as needed. This example requires changing the following three items:
    - The `catalog load --filepath` needs to be updated to reflect the JSON file of the new image, such as changing `cdh57CM-centos.json` to `cdh59CM-centos.json`.
    - In the CentOS catalog bundle section, the `imgversion` needs to be updated. This example demonstrates building a new CDH 5.9 image, so the image version will be `1.0`.
    - In the CentOS catalog bundle section, update the `catalog save --filepath` to also reflect the JSON file of the new image.
14. Verify that the lines in the `RHEL catalog bundle` section are commented out, because this example is for a CentOS image.

    The following image shows the changes made to the Dockerfile (changes appear in red text).

```
                        #
                        # YOUR_ORGANIZATION_NAME must be replaced with a
valid organization name. Please
                        # refer to 'help builder organization' for details.
                        #
                        builder organization --name BlueData

                        catalog load --filepath cdh59CM-centos.json
                        appconfig package --dir cdh5-cm-setup
                        logo file --filepath Logo_Cloudera.png

                        #@# Centos catalog bundle
                        image build --basedir image/centos --imgversion
1.0 --os centos
                        catalog save --filepath staging/
cdh59CM-centos.json --force
                        sources package --additional cdh5-cm-setup
                        catalogpackage --os=centos

                        ##RHEL catalog bundle
                        #imagebuild --basedir image/rhel --imgversion
2.0 --os rhel
                        #catalogsave --filepath staging/
cdh57CM-rhel.json --force
                        #sources package --additional cdh5-cm-setup
                        #catalog package --os=rhel

                        workbench clean
                        ~
                        ~
                        "cdh59cm.wb" 24L, 769C
```

15. Navigate to the `image/<os>` directory (such as `centos`), which contains the `cloudera-manager.repo` and Dockerfile. Dockerfile contents will vary by application and configuration. See Sample Docker Files on page 1049 for two sample Dockerfiles.

16. Open the `cloudera-manager.repo` file, and then verify that the base URL path is valid for your desired image before you update the `repo` file. In this example, update the `baseurl` path to `5.9.0`.

```
                          baseurl=http://archive.coudera.com/cm5/redhat/6/
x86_64/cm/5.9.0
                          gpgkey = http://archive.cloudera.com/cm5/redhat/6/
x86_64/cm/RPM-GPG-KEY-cloudera
```

17. Navigate to the `image/<os>` directory, and then open the Dockerfile. The contents of this file will vary depending on the application and its unique configurations. In this example, a total of 15 changes are required. Most of these changes involve simply changing `5.7.0` to `5.9.0`). It is recommended that you verify the parcel and rpm names with the distribution (Hadoop, Spark, Cloudera) in order to confirm that all changes required for this upgrade have been made. The following text shows a sample Dockerfile before and after making these changes, where:

    - The **before** example uses a gray background.
    - The **after** example uses a black background. Changes are shown in red text.

```
# CDH 5.7.0 docker image

                         FROM bluedata/centos6:latest

                         #LABEL Description="This is an image for CDH5.7.0"
Vendor="BlueData, Inc"

                         # remove openjdk we installed in base image

                        RUN yum -y erase java-1.7.0-openjdk

                        ENV JAVA_HOME ''

                         # Install cloudera manager and parcels

                        ADD cloudera-manager.repo /etc/yum.repos.d/

                        RUN yum -y install oracle-j2sdk1.7.x86_64
cloudera-manager-server-db-2-5.7.0   \

                                            cloudera-manager-server-5.7.0
cloudera-manager-daemons-5.7.0\

                                            cloudera-manager-agent-5.7.0 krb5*\

                         #Clean up the yum repository

                        RUN yum clean all; rm -rf /tmp/* /var/tmp/* /var/cache/jum/*

                         # Install python php and install cm-api through pip

                        RUN wget https://bootstrap.pypa.io/get-pip.py -P /tmp

                        RUN python /tmp/get-pip.py

                        RUN pip install cm-api==12.0.0

                         # download parcels for CDH

                        RUN wget http://archive.cloudera.com/cdhg5/parcels/5.7.0/
CDH-5.7.0-1.cdh5.7.0.p0.45-el6.parcel -P /opt/cloudera/parcel-repo
```

```
                        # CDH 5.9.0 docker image

                        FROM bluedata/centos6:latest

                        #LABEL Description="This is an image for CDH5.9.0"
Vendor="BlueData, Inc"

                        # remove openjdk we installed in base image
                        RUN yum -y erase java-1.7.0-openjdk
                        ENV JAVA_HOME ''

                        # Install cloudera manager and parcels
                        ADD cloudera-manager.repo /etc/yum.repos.d/
                        RUN yum -y install oracle-j2sdk1.7.x86_64
cloudera-manager-server-db-2-5.9.0   \
                                          cloudera-manager-server-5.9.0
cloudera-manager-daemons-5.9.0\
                                          cloudera-manager-agent-5.9.0
krb5*\

                        #Clean up the yum repository
                        RUN yum clean all; rm -rf /tmp/* /var/tmp/* /var/
cache/jum/*

                        # Install python php and install cm-api through pip
                        RUN wget https://bootstrap.pypa.io/
get-pip.py -P /tmp
                        RUN python /tmp/get-pip.py
                        RUN pip install cm-api==12.0.0

                        # download parcels for CDH
                        RUN wget http://archive.cloudera.com/cdhg5/parcels/
5.9.0/CDH-5.9.0-1.cdh5.9.0.p0.23-e16.parcel -P /opt/cloudera/parcel-repo
                        RUN
wget http://archive.cloudera.com/cdhg5/parcels/5.9.0/
CDH-5.9.0-1.cdh5.9.0.p0.23-e16.parcel.sha1 -P /opt/cloudera/parcel-repo
                        RUN mv /opt/cloudera/parcel-repo/
CDH-5.9.0-1.cdh5.9.0.p0.23-e16.parcel.sha1 /opt/cloudera/parcel-repo/
CDH-5.9.0-1.cdh5.9.0.p0.23-e16.parcel.sha
                        RUN chown cloudera-scm:cloudera-scm /opt/cloudera/
parcel-repo/*

                        ## Download Unlimited JCE policy
zip file and copy to the parent directory of
                        ## this Dockerfile before uncommenting this section

                        #Add UnlimitedJCEPolicyJDK7.zip .
                        #RUN wzip UnlimtedJCEPolicyJDK7.zip &&
rm UnlimtedJCEPolicyJDK7.zip
                        #RUN
mv UnlimtedJCEPolicy/US_export_Policy.jar /usr/java/
jdk1.7.*-cloudera/jre/lib/security/local_policy.jar
                        #RUN rm -rf UnlimtedJCEPolicy*
```

18. Update the JSON file. To upgrade an image, you will need to make changes to various parameters, as appropriate to the application. This example requires the following changes (In the following examples, the **before** examples use a gray background, and the **after** examples use a black background with changes shown in red text):

    - `cdh_parcel_repo`
    - `cdh_full_version`

- `cdh_parcel_version`
- `source_file`
- label name
- label description
- `distro_id`
- `version`

```
"catalog_api_version": 2,

                        "config": {

                          "config_meta": {     "streaming_jar": "/opt/cloudera/
parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar",

                            "cdh_major_version": "CDH5",

                            "cdh_parcel_repo": "http://archive.cloudera.com/cdh5/
parcels/5.7.0",

                            "cdh_full_version": "5.7.0",

                            "cdh_parcel_version": "5.7.0-1.cdh5.7.0.p0.45",

                            "impala_jar_version": "0.1-SNAPSHOT"
```

```
                           "catalog_api_version": 2,
                           "config": {
                             "config_meta": {     "streaming_jar": "/opt/
cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar",
                               "cdh_major_version": "CDH5",
                               "cdh_parcel_repo": "http://
archive.cloudera.com/cdh5/parcels/5.9.1",
                               "cdh_full_version": "5.9.1",
                               "cdh_parcel_version": "5.9.1-1.cdh5.7.1.p0.4",
                               "impala_jar_version": "0.1-SNAPSHOT"
```

```
},

                        "image": {

                          "checksum": "31bb8c37ecb491b8a42660a06428eace",

                          "source_file": "bluedata-cdh57-centos-1.0.tar.gz"

                        },

                        "label": {

                          "name": "CDH 5.7.0 with Cloudera Manager",

                          "description": "CDH 5.7.0 with MRv1/YARN and HBase
support. Includes Pig, Hive, Hue and Spark."

                        },

                        "distro_id": "bluedata/cdh57",

                        "version": "2.1",

                        "services": [

                          {
```

```
                          },
                          "image": {
                            "checksum": "31bb8c37ecb491b8a42660a06428eace",
                            "source_file": "bluedata-cdh59-centos-1.0.tar.gz"
                          },
                          "label": {
                            "name": "CDH 5.9.1 with Cloudera Manager",
                            "description": "CDH 5.9.1 with MRv1/YARN and
HBase support. Includes Pig, Hive, Hue and Spark."
                          },
                          "distro_id": "bluedata/cdh59",
                          "version": "1.0",
                          "services": [
                            {
```

19. Execute the `.wb` by executing the command `./<wb name>.wb`, such as `./cdh59cm.wb`. Image creation can take several minutes depending on the size and complexity of your image. You will be alerted if there are any errors. You will also be alerted once the build is successful.

20. Once the bin build process is completed, copy your new image `.bin` from the `/deliverables` directory that is generated during the build.

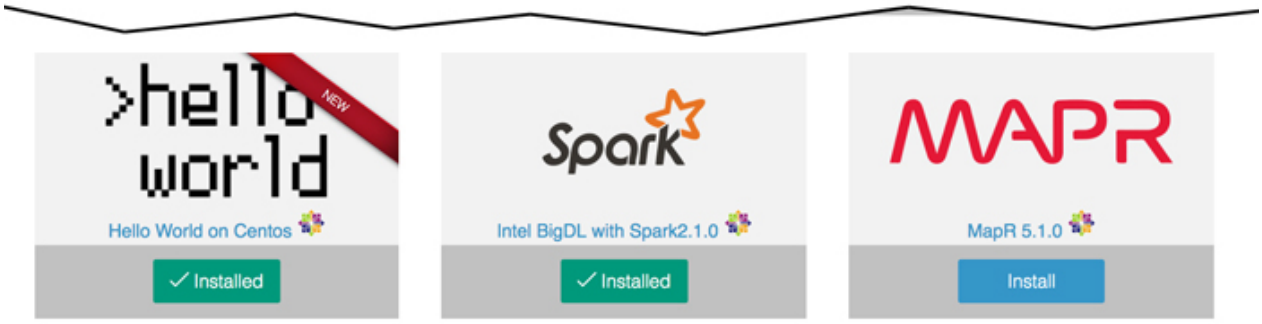
```
                          Successfully built bluedata/cdhg59-centos:1.0.

                          Saving bluedata/cd59-centos:1.0 as /root/source/
cdh59/staging/bluedata-cdh-59-centos-1.0.tar.gz
                          Packaging the entry for centos.
                          Catalog bundle is saved at /root/source/cdh59/
deliverables/bdcatalog-centos-bluedata-cdh-59-2.1.bin
```

21. Copy the new `.bin` to `</srv/bluedata/catalog`, and then make it executable by executing the command `chmod +x <bin name>`.

22. Log into the HPE Ezmeral Runtime Enterprise web interface as a Platform Administrator, navigate to the **App Store** screen, and then click **Refresh**.

    The new image will appear in the **App Store** with a red **New** banner across the tile.



23. Click the **Install** button to install the image, and then create a test cluster as a final validation step.

API Matrices

The App Workbench includes the following APIs:

- **Catalog API:** Refers to HPE Ezmeral Runtime Enterprise management capabilities, such as support for custom Worker roles. See Catalog API.
- **Configuration API:** Handles metadata and JSON structure definitions. See Configuration API.

The tables in these sections list the API versions from newest to oldest, and includes a list of the features added to each version. New features are cumulative, meaning that each new version of an API includes all of the features introduced in previous versions. For example, if you need AWS support, then you may use Version 4 or higher of the Configuration API.

The API versions are specified in the .wb file. If you do not specify a version for one or both of these APIs, then App Workbench will automatically use the latest version(s) of the API(s). For example, if you specify `--configapi 4` and do not specify a Catalog API version, then AppWorkbench will use the latest version of the Catalog API and Version 4 of the Configuration API.

NOTE: It is best to use the latest API versions unless you are creating an **App Store** entry for an earlier version of EPIC.

Catalog API

The Catalog API versions, corresponding EPIC version, and features are:

API Version	EPIC Version	Notes

6	4.0	Adds support for AI/ML projects and additional enhancements.
5	3.5	Adds a "string" type to the metadata JSON.
4	3.4	Support for separate Docker images per virtual node role.
3	3.1	Custom Worker role definitions
2	3.0.5 and previous	

Configuration API

The Configuration API versions, corresponding EPIC version, and features are:

API Version	EPIC Version	Notes
11	5.0	Added support for Ubuntu 18.04.
10	4.0	Added attachments and namespaces, and the `--reattach` API call to the startscript.
9	3.6	Added the `--delete` API call to the startscript when deleting the virtual nodes (containers).
8	3.4	Remote copy and execute, plus platform information in namespaces.
7	3.2	Support for Isolated mode, bootstrap actions, and two-phase cluster deletion.
6	3.0.5	Support for CentOS/RHEL 7.x container images and `systemctl` service registration support.
5	2.6	Added node ID to FQDN maps and introduced `bdmacros` Python library.
4	2.5	First version with AWS support. This version is not recommended for use.
3	2.4	Tenant information in `bdvcli` namespaces.
2	2.3	Introduced Catalog bundles.

HPE Ezmeral Runtime Enterprise Documentation Home

Get started with HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise is software platform designed to deploy cloud-native and non-cloud-native applications using 100% open-source Kubernetes—running on bare-metal or virtualized infrastructure, on-premises, on any cloud, and at the edge. Get started with HPE Ezmeral Runtime Enterprise:

HPE Ezmeral ML Ops

Supports the entire machine learning lifecycle and implement DevOps-like processes to standardize machine learning workflows.

last-updated: May 06, 2024

HPE Ezmeral Data Fabric

An exabyte-scale, edge-to-cloud distributed file system and data platform for the diverse data needs of modern enterprise applications.

HPE Ezmeral Runtime Analytics for Apache Spark

A hybrid analytics platform that spans edge to cloud, with enterprise-grade Apache Spark and Delta Lake integration and support for external business intelligence apps.

App Workbench

An SDK to help you rebuild or redesign application architecture for deployment on HPE Ezmeral Runtime Enterprise.

HPE Ezmeral Runtime Enterprise Administration

Get started with administration tasks on HPE Ezmeral Runtime Enterprise:

HPE Ezmeral Runtime Enterprise Release Information

Information about the current release:

last-updated: May 06, 2024