HPE Ezmeral Unified Analytics Software 1.5 Documentation

HPE Ezmeral Software Unified Analytics

EZUA 1.5.x

Contents

Get Started

Describes how to get started with HPE Ezmeral Unified Analytics Software.

The following sections provide links to topics for administrators and members to get started with HPE Ezmeral Unified Analytics Software.

Administrator

The Administrator may be interested in the following topics:

- Installation on page 63
- AD/LDAP Servers on page 227
- Identity and Access Management on page 220
- Adding and Removing Users on page 233
- Importing Frameworks and Managing the Application Lifecycle on page 117
- Connecting to HPE Ezmeral Data Fabric on page 133
- Configuring Endpoints on page 135

Member

The Member (non-administrative users) may be interested in the following topics:

- Tutorials on page 8
- Data Engineering on page 291
- Data Analytics on page 368
- Data Science on page 408
- Notebooks on page 434

About

Provides an overview of HPE Ezmeral Unified Analytics Software.

HPE Ezmeral Unified Analytics Software is usage-based Software-as-a-Service (SaaS) that fully manages, supports, and maintains hybrid and multi-cloud modern analytical workloads through open-source tools. HPE Ezmeral Unified Analytics Software separates compute and storage for flexible, cost-efficient scalability to securely access data stored in multiple data platforms through a simple user interface, which is easily installed and deployed on private, public, and on-premises infrastructure.

Features and Functionality

HPE Ezmeral Unified Analytics Software provides the following features and functionality in a single UX:

Access data anywhere and manage it in one place	Connect bidirectionally to multiple data platforms and join data to create a federated data mesh that you manage in one place. Includes authentication, authorization, logging, metrics collection, and monitoring.

Robust, integrated storage layer	Includes an integrated, scalable data fabric storage layer with data-mesh like capabilities as the ephemeral storage for all types of data, including structured and unstructured data, files, objects, and streams.
Analytical workloads	Support for the most common enterprise analytics use cases ranging from traditional BI/Reporting (via PrestoDB and SparkSQL interfaces) to emerging workloads, such as exploratory data science, real-time analytics, and machine learning workflows.
Self-service data access	All users, including administrators, data engineers, data analysts, and data scientists can directly access data from HPE Ezmeral Unified Analytics Software.
Built-in access to BI dashboards and data science tools	Includes built-in BI dashboards for analytics and operational reporting, Also includes web-based notebook interfaces, such as Jupyter Lab and Visual Studio, for data science workflows (model training and serving frameworks).
Built-in SSO	Supports single sign-on experience; users sign in to access HPE Ezmeral Unified Analytics Software and compute components integrate with the storage platform infrastructure to pass the identity of each user.
Performance	Distributed, in-memory caching (explicit) that accelerates federated queries on commonly used datasets.

Compute Components

The compute components included in HPE Ezmeral Unified Analytics Software enable users to get up and running in minutes. Components connect to each other at start-up and use pre-defined storage areas in the built-in data fabric. When applicable, compute components can automatically take advantage of GPUs.

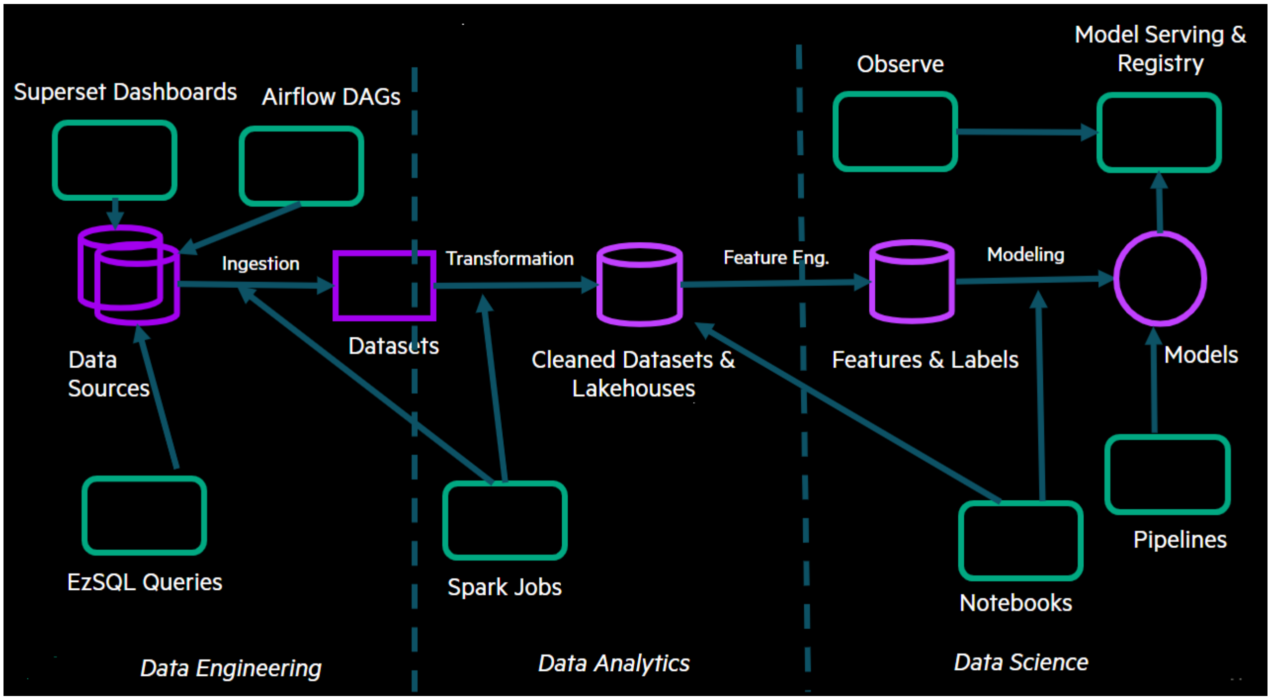
The following list describes the compute components included in HPE Ezmeral Unified Analytics Software:

Spark	Spark is a primary engine for data analytics tasks.
EzPresto	EzPresto is a distributed SQL query engine with a built-in query federation capability (distributed in-memory caching and pushdown optimizations) for fast analytic queries on data of any size.
Kubeflow	Kubeflow as an ML framework focused on model training that includes Notebooks, Pipelines (Airflow), Experiments, Kserve, and various distributed training operators.
Airflow	Airflow for data engineering and task automation.
Notebooks	Jupyter notebooks for performing varied data science tasks, such as cleaning data, labeling features, testing toy models, and launching distributed training models.
Dashboard Frameworks	Dashboard frameworks for building data models and visualizations.

Workflows and Pipelines

HPE Ezmeral Unified Analytics Software provides simplified workflows and pipelines for data engineers, data analysts, and data scientists to solve complex problems.

The following image shows some of the supported workflows and pipelines:



Tutorials

Provides a set of tutorials that you can use to experience HPE Ezmeral Unified Analytics Software and the included applications, such as tutorials for data science and data analytics workflows with notebooks and applications like Spark, MLflow, Feast, Airflow, and EzPresto.

The following sections provide links to access the complete tutorials in GitHub.

Data Source Connectivity and Exploration

Provides basic steps for using the Data Engineering space within HPE Ezmeral Unified Analytics Software.

You can connect to data sources and work with data within the Data Engineering space of HPE Ezmeral Unified Analytics Software. The Data Engineering space includes:

- **Data Sources** – View and access connected data sources; create new data source connections.
- **Data Catalog** – Select data sets (tables and views) from one or more data sources and query data across the data sets. You can cache data sets. Caching stores the data in a distributed caching layer within the data fabric for accelerated access to the data.
- **Query Editor** – Run queries against selected data sets; create views and new schemas.
- **Cached Assets** – Lists the cached data sets (tables and views).
- **Airflow Pipelines** – Links to the Airflow interface where you can connect to data sets created in HPE Ezmeral Unified Analytics Software and use them in your data pipelines.

Tutorial Objective

Although you can perform more complex tasks in HPE Ezmeral Unified Analytics Software, the purpose of this tutorial is to walk you through some Data Engineering basics and familiarize you with the interface, including how to:

- Connect data sources
- Select predefined data sets in data sources
- Join data across data sets/data sources
- Create a view
- Run a query against the view

This tutorial takes approximately 10 minutes to complete.

You may want to print the following instructions or open the instructions on a different monitor to avoid switching between HPE Ezmeral Unified Analytics Software and the tutorial on one monitor.

IMPORTANT: This tutorial demonstrates how to perform a series of tasks in HPE Ezmeral Unified Analytics Software to complete an example workflow. The data and information used in this tutorial is for example purposes only. You must connect Unified Analytics to your own data sources and use the data sets available to you in your data sources.

A – Sign in to HPE Ezmeral Unified Analytics Software

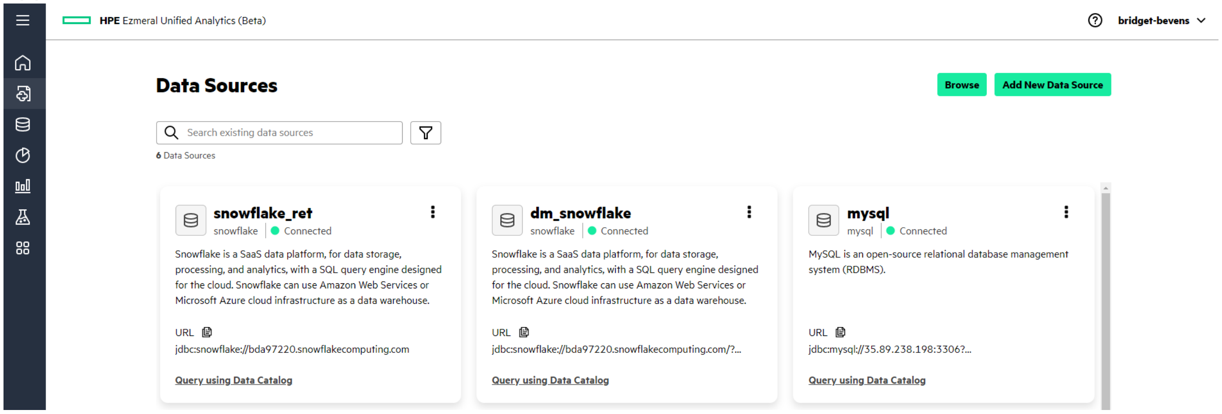
Sign in to HPE Ezmeral Unified Analytics Software with the URL provided by your administrator.

B – Connect Data Sources

Connect HPE Ezmeral Unified Analytics Software to external data sources that contain the data sets (tables and views) you want to work with. This tutorial uses MySQL and Snowflake as the connected data sources.

To connect a data source:

1. In the left navigation column, select **Data Engineering > Data Sources**. The **Data Sources** screen appears.



Snowflake is a SaaS data platform, for data storage, processing, and analytics, with a SQL query engine designed for the cloud. Snowflake can use Amazon Web Services or Microsoft Azure cloud infrastructure as a data warehouse.

jdbc:snowflake://bda97220.snowflakecomputing.com

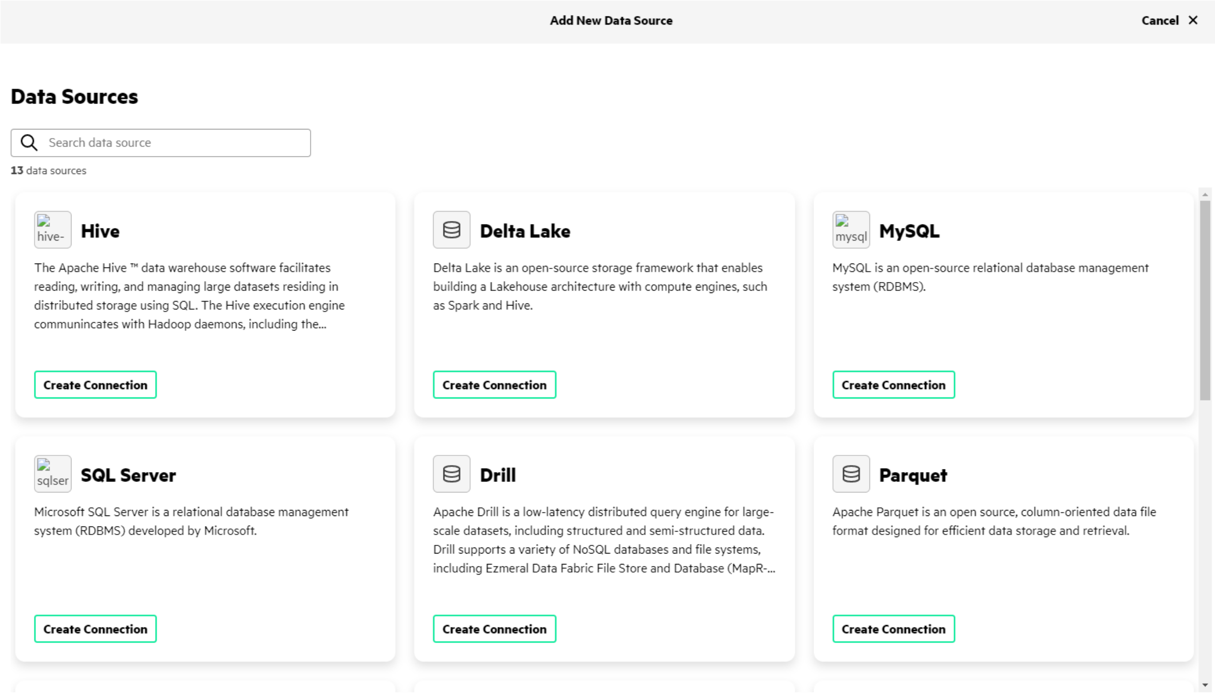
Snowflake is a SaaS data platform, for data storage, processing, and analytics, with a SQL query engine designed for the cloud. Snowflake can use Amazon Web Services or Microsoft Azure cloud infrastructure as a data warehouse.

jdbc:snowflake://bda97220.snowflakecomputing.com/?...

MySQL is an open-source relational database management system (RDBMS).

jdbc:mysql://35.89.238.198:3306?...

2. Click **Add New Data Source**.



Data Sources

13 data sources

Hive

The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. The Hive execution engine communincates with Hadoop daemons, including the...

Delta Lake

Delta Lake is an open-source storage framework that enables building a Lakehouse architecture with compute engines, such as Spark and Hive.

MySQL

MySQL is an open-source relational database management system (RDBMS).

SQL Server

Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft.

Drill

Apache Drill is a low-latency distributed query engine for large-scale datasets, including structured and semi-structured data. Drill supports a variety of NoSQL databases and file systems, including Ezmeral Data Fabric File Store and Database (MapR-...

Parquet

Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval.

3. Complete the steps required to connect to the MySQL, Snowflake, and Hive data sources:

NOTE: When you create a data source connection, do not include an underscore (_) in the data source name. EzPresto does not support underscores (_) in data source names. For example, my_sql is not supported; instead, use something like mysql.

Connecting to MySQL

a. In the **Add New Data Source** screen, click **Create Connection** in the **MySQL** tile.

b. In the drawer that opens, enter the required information in the respective fields:

NOTE: The information used here is for example purposes only.

- **Name**: mysql
- **Connection URL**: jdbc:mysql://<ip-address>:<port>
- **Connection User**: demouser
- **Connection Password**: moi123
- **Enable Local Snapshot Table**: Select the check box

TIP: When **Enable Local Snapshot Table** is selected, the system caches remote table data to accelerate queries on the tables. The cache is active for the duration of the configured TTL or until the remote tables in the data source are altered.

- Click **Connect**. Upon successful connection, the system returns the following message:

  ```
  Successfully added data source
  "mysql".
  ```

Connecting to Snowflake

a. In the **Add New Data Source** screen, click **Create Connection** in the **Snowflake** tile.

b. In the drawer that opens, enter the following information in the respective fields:

- **Name**: snowflakeret
- **Connection URL**: jdbc:snowflake://mydomain.com/
- **Connection User**: demouser
- **Connection Password**: moi123
- **Snowflake DB**: my_snowflake_db
- **Enable Local Snapshot Table**: Select the check box

  TIP: When **Enable Local Snapshot Table** is selected, the system caches remote table data to accelerate queries on the tables. The cache is active for the duration of the configured TTL or until the remote tables in the data source are altered.

- Click **Connect**. Upon successful connection, the system returns the following message:

  ```
  Successfully added data source
  "snowflakeret".
  ```

Connecting to Hive

a. In the **Add New Data Source** screen, click **Create Connection** in the **Hive** tile.

b. In the drawer that opens, enter the following information in the respective fields:

- **Name**: hiveview
- **Hive Metastore**: file
- **Hive Metastore Catalog Dir**: file:///data/shared/tmpmetastore
- In **Optional Fields**, search for the following fields and add the specified values:
  - **Hive Max Partitions Per Writers**: 10000

    - **Hive Temporary Staging Directory Enabled**: Unselect
    - **Hive Allow Drop Table**: Select
- **Enable Local Snapshot Table**: Select the check box

  TIP: When **Enable Local Snapshot Table** is selected, the system caches remote table data to accelerate queries on the tables. The cache is active for the duration of the configured TTL or until the remote tables in the data source are altered.

- Click **Connect**. Upon successful connection, the system returns the following message:

  ```
  Successfully added data source
  "hiveview".
  ```

C – Select Data Sets in the Data Catalog

In the Data Catalog, select the data sets (tables and views) in each of the data sources that you want to work with.

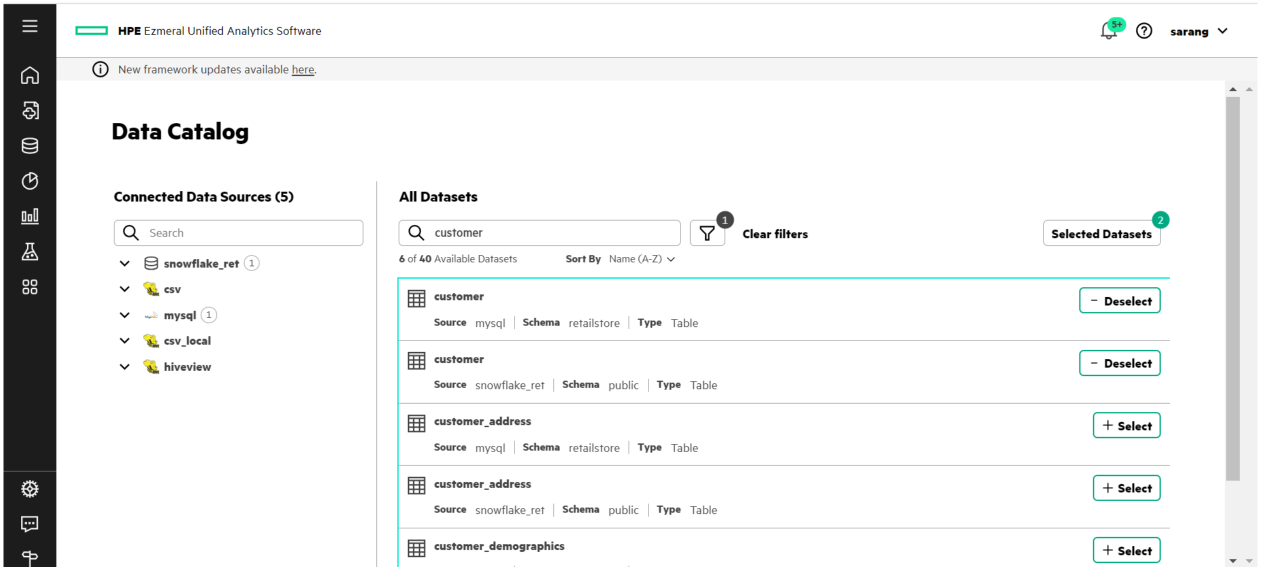
This tutorial uses the **customer** tables in the connected **mysql** and **snowflakeret** data sources. In the **mysql** data source, the schema for the customer table is **retailstore**. In the **snowflakeret** data source, the schema for the customer table is **public**.

To select the data sets that you want to work with:

1. In the left navigation bar, select **Data Engineering > Data Catalog**.
2. On the **Data Catalog** page, click the dropdown next to the **mysql** and **snowflakeret** data sources to expose the available schemas in those data sources.
3. For the **snowflakeret** data source select the **public** schema and for the **mysql** data source, select the **retailstore** schemas.
4. In the **All Datasets** search field, enter a search term to limit the number of data sets. This tutorial searches on data sets with the name **customer**. All the data sets that have *customer* in the name with *public* or *retailstore* schema display.
5. Click a **customer** table and preview its data in the **Columns** and **Data Preview** tabs.

   NOTE: Do not click the browser's back button; doing so takes you to the Data Sources screen and you will have to repeat the previous steps.

6. Click **Close** to return to the data sets.
7. Click **Select** by each of the tables named **customer**. Selected Datasets should show 2 as the number of data sets selected.
8. Click **Selected Datasets**. The Selected Datasets drawer opens, giving you another opportunity to preview the datasets or discard them. From here, you can either query or cache the selected data sets. For the purpose of this tutorial, we will query the data sets.
9. Click **Query Editor**.



New framework updates available here.

Data Catalog

Connected Data Sources (5)

- snowflake_ret (1)
- csv
- mysql (1)
- csv_local
- hiveview

All Datasets

customer

6 of 40 Available Datasets

customer
Source mysql | Schema retailstore | Type Table

customer
Source snowflake_ret | Schema public | Type Table

customer_address
Source mysql | Schema retailstore | Type Table

customer_address
Source snowflake_ret | Schema public | Type Table

customer_demographics

D – Run a JOIN Query on Data Sets and Create a View

The data sets you selected display under Selected Datasets in the Query Editor. Run a JOIN query to join data from the two customer tables and then create a view from the query. The system saves views as cached assets that you can reuse.

To view table columns and run a JOIN query:

1. Expand the customer tables in the **Selected Datasets** section to view the columns in each of the tables.
2. In the **SQL Query** workspace, click **+** to add a worksheet.
3. Copy and paste the following query into the **SQL Query** field. This query creates the a new schema in the `hiveview` data source named `demoschema`:

   ```
   create schema if not exists hiveview.demoschema;
   ```

4. Click **Run** to run the query. As the query runs, a green light pulsates next to the Query ID in the Query Results section to indicate that the query is in progress. When the query is completed, the Status column displays Succeeded.
5. In the **SQL Query** workspace, click **+** to add a worksheet.
6. Copy and paste the following query into the **SQL Query** field. This query creates a view (hiveview.demoschema) from a query that joins columns from the two **customer** tables (in the mysql and snowflakeret data sources) on the **customer ID**.

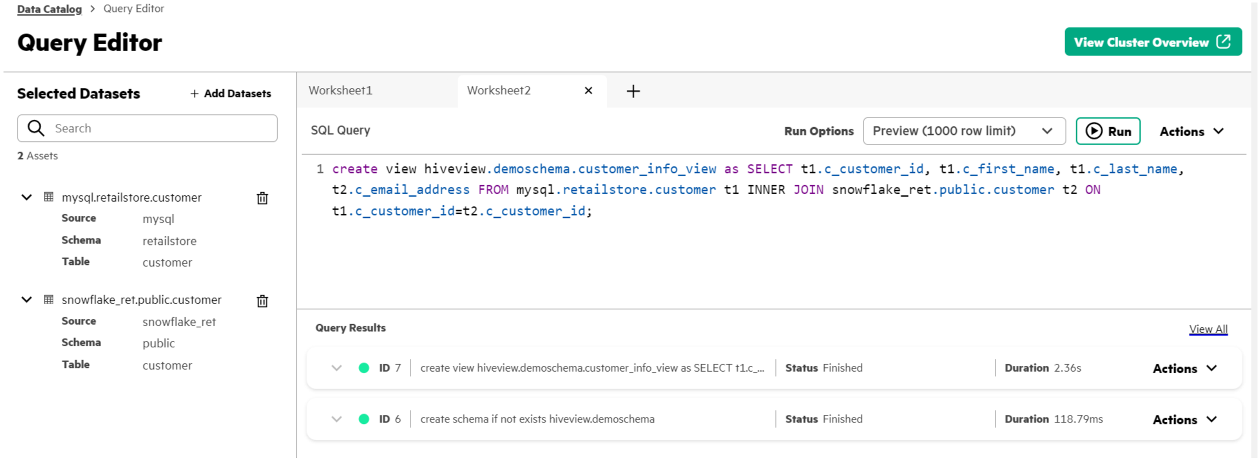
   ```
   create view hiveview.demoschema.customer_info_view as
   SELECT t1.c_customer_id, t1.c_first_name, t1.c_last_name,
   t2.c_email_address FROM mysql.retailstore.customer t1 INNER JOIN
   snowflakeret.public.customer t2 ON t1.c_customer_id=t2.c_customer_id
   ```

7. Click **Run** to run the query.
8. In the **SQL Query** workspace, click **+** to add a worksheet.

9. Copy and paste the following query into the **SQL Query** field. This runs against the view you created (hiveview.demoschema) and returns all data in the view.

```
SELECT * FROM hiveview.demoschema.customer_info_view;
```

10. Click **Run** to run the query.



mysql.retailstore.customer
Source mysql
Schema retailstore
Table customer

snowflake_ret.public.customer
Source snowflake_ret
Schema public
Table customer

```
create view hiveview.demoschema.customer_info_view as SELECT t1.c_customer_id, t1.c_first_name, t1.c_last_name,
t2.c_email_address FROM mysql.retailstore.customer t1 INNER JOIN snowflake_ret.public.customer t2 ON
t1.c_customer_id=t2.c_customer_id;
```

ID 7 | create view hiveview.demoschema.customer_info_view as SELECT t1.c_... | Status Finished | Duration 2.36s | Actions

ID 6 | create schema if not exists hiveview.demoschema | Status Finished | Duration 118.79ms | Actions

11. In the**Query Results**section, expand the**Actions** option for the query and select**View Details**to view the query session and resource utilization summary.

12. Click**Close**to exit out of Query Details.

End of Tutorial

You have completed this tutorial. This tutorial demonstrated how easy it is to connect HPE Ezmeral Unified Analytics Software to various data sources for federated access to data through a single interface using standard SQL queries.

You may also be interested in the BI Reporting (Superset) Basics on page 14, which shows you how to create a Superset dashboard using the view (customer_info_view) and schema (customer_schema) created in this tutorial.

BI Reporting (Superset) Basics

Provides basic steps for using the BI Reporting (Superset) space within HPE Ezmeral Unified Analytics Software.

You can add data sets that you created in HPE Ezmeral Unified Analytics Software to Superset and visualize the data in dashboards. You can access dashboards (Superset) from the BI Reporting space within HPE Ezmeral Unified Analytics Software.

Tutorial Objective

The purpose of this tutorial is to walk you through some Superset basics to familiarize you with the interface and how to use it with the data sets you create in HPE Ezmeral Unified Analytics, including how to:

- Add datasets created in HPE Ezmeral Unified Analytics Software to Superset
- Visualize the data set in a chart
- Create a dashboard
- Add the chart to the dashboard

This tutorial takes approximately 10 minutes to complete.

You may want to print the following instructions or open the instructions on a different monitor to avoid switching between HPE Ezmeral Unified Analytics Software and the tutorial on one monitor.

IMPORTANT: This tutorial demonstrates how to perform a series of tasks in HPE Ezmeral Unified Analytics Software to complete an example workflow. The data and information used in this tutorial is for example purposes only. You must connect Unified Analytics to your own data sources and use the data sets available to you in your data sources.

Prerequisite

This tutorial builds on Data Source Connectivity and Exploration on page 8.

In the **Data Source Connectivity and Exploration** tutorial, you created a view (customer_info_view) and a schema (customer_schema) from a query that joined customer tables from two different data sources (MySQL and Snowflake). In this tutorial, you import the view and schema into Superset, visualize the data in a chart, and add the chart to a dashboard.

A – Sign in to HPE Ezmeral Unified Analytics Software

Sign in to HPE Ezmeral Unified Analytics Software with the URL provided by your administrator.

B - Connect to the Presto Database

Complete the following steps to connect Superset to the Presto database for access to your data sources and data sets in HPE Ezmeral Unified Analytics Software. Once connected to the Presto database, you can access your data sets in HPE Ezmeral Unified Analytics Software from Superset.

To connect to the Presto database, you need the connection URI. You can get the URI from your HPE Ezmeral Unified Analytics Software administrator. The format of the connection URI is:

```
presto://<presto.domain.name>:443/<catalogname>
```

TIP: If you signed in to Superset through the HPE Ezmeral Unified Analytics Software UI, you do not have to enter your user credentials for the EzPresto connection URI because Unified Analytics authenticates you when you sign in to the system.

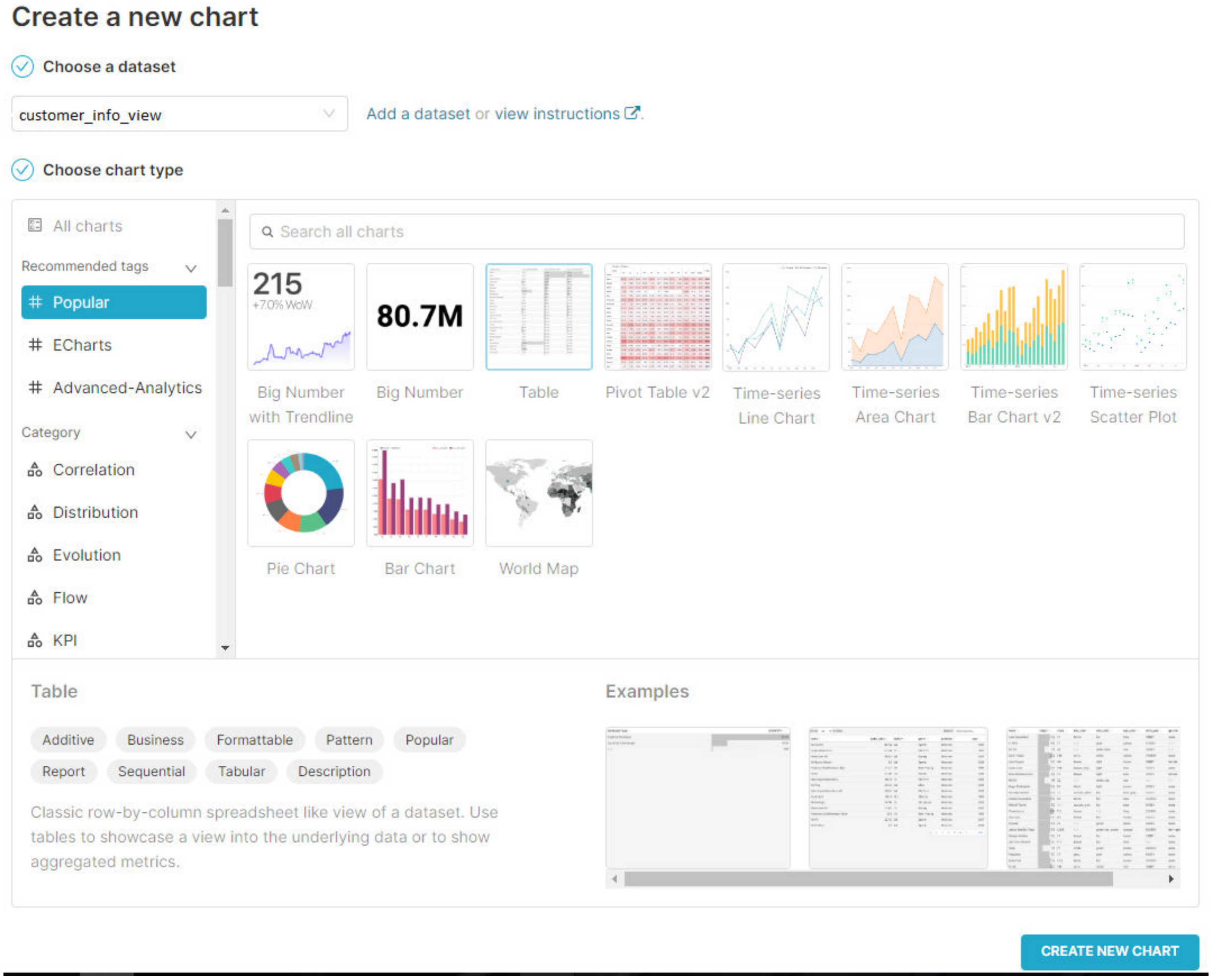
1. To open Superset, in the left navigation pane of HPE Ezmeral Unified Analytics Software, select **BI Reporting > Dashboards**. Superset opens in a new tab.
2. In Superset, select **Settings > Database Connections**.
3. Click **+DATABASE**.
4. In the **Connect a database** window, select the **Presto** tile.
5. Enter the SQLALCHEMY URI provided by your administrator.
6. Test the connection.
7. If the test was successful, click **Connect**.

C – Add a Data Set to a Chart

To add a dataset to a chart:

1. Select the **Datasets** tab.
2. Click **+ DATASET**.

3. In the **Add dataset** window, make the following selections in the fields:
   - **DATABASE:** Presto
   - **SCHEMA:** <your_schema>
   - **SEE TABLE SCHEMA:** <your_view>
4. Click **ADD DATASET AND CREATE CHART**.
5. In **Choose chart type** column, select **#Popular** and choose **Table**.
6. Click **CREATE NEW CHART**.



Create a new chart

Choose a dataset: customer_info_view

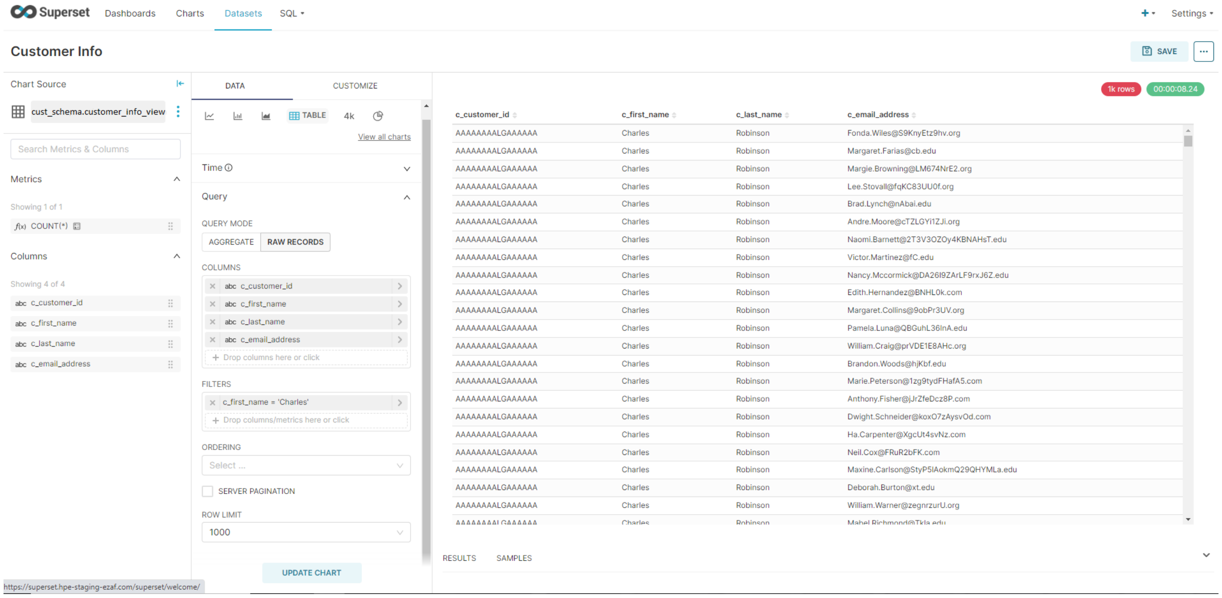
Choose chart type: Popular, Table

Table

Classic row-by-column spreadsheet like view of a dataset. Use tables to showcase a view into the underlying data or to show aggregated metrics.

7. In the chart screen, enter a name for the chart. For example, name the chart **Customer Info**.
8. Select **RAW RECORDS** as the **QUERY MODE**.
9. Drag and drop the following four columns into the **COLUMNS** field:
   - c_customer_id
   - c_first_name
   - c_last_name
   - c_email_address

10. Click into the **Filters** field and select or enter the following information in the window that opens:
    - c_first_name
    - Equal to (=)
    - Charles
11. Click **SAVE**.
12. Click **CREATE CHART**. The query runs and results that meet the query conditions display. The chart displays four columns of data for customers with the first name Charles.
13. Click **SAVE** to save the chart. A window opens. Click **SAVE** in the window. Do not add to a dashboard. Superset saves the chart.



Customer Info

Chart Source: cust_schema.customer_info_view

Metrics
Showing 1 of 1
f(x) COUNT(*)

Columns
Showing 4 of 4
abc c_customer_id
abc c_first_name
abc c_last_name
abc c_email_address

QUERY MODE: RAW RECORDS

COLUMNS
abc c_customer_id
abc c_first_name
abc c_last_name
abc c_email_address

FILTERS
c_first_name = 'Charles'

ORDERING

☐ SERVER PAGINATION

ROW LIMIT
1000

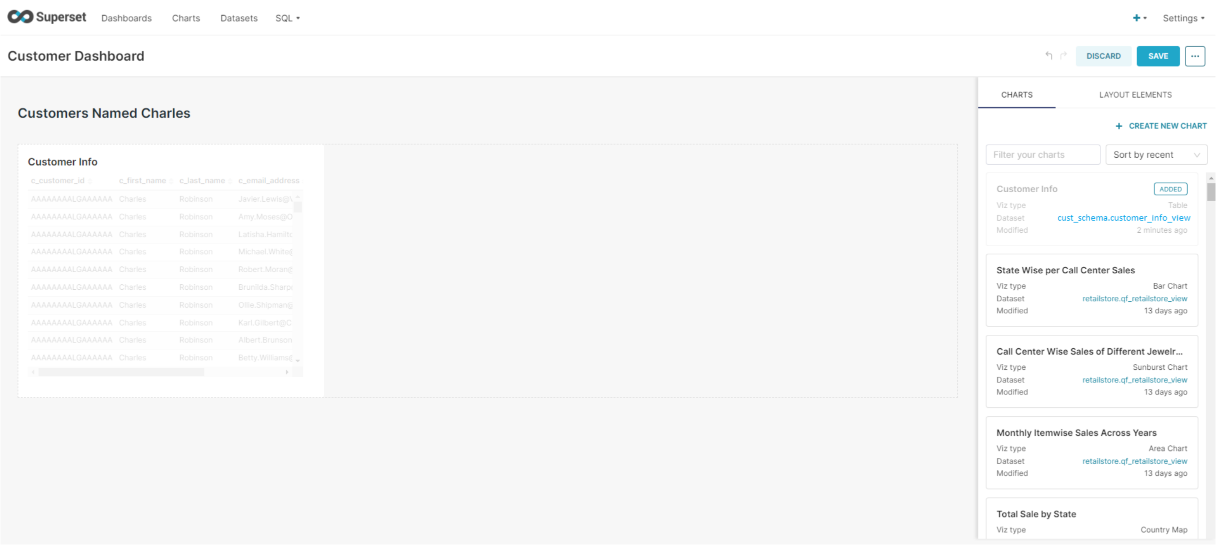
c_customer_id	c_first_name	c_last_name	c_email_address
AAAAAAAALGAAAAAA	Charles	Robinson	Fonda.Wiles@S9KnyEtz9hv.org
AAAAAAAALGAAAAAA	Charles	Robinson	Margaret.Farias@cb.edu
AAAAAAAALGAAAAAA	Charles	Robinson	Margie.Browning@LM674NrE2.org
AAAAAAAALGAAAAAA	Charles	Robinson	Lee.Stovall@fqKC83UU0f.org
AAAAAAAALGAAAAAA	Charles	Robinson	Brad.Lynch@nAbai.edu
AAAAAAAALGAAAAAA	Charles	Robinson	Andre.Moore@cTZLGYi1ZJi.org
AAAAAAAALGAAAAAA	Charles	Robinson	Naomi.Barnett@2T3V3OZOy4KBNAHsT.edu
AAAAAAAALGAAAAAA	Charles	Robinson	Victor.Martinez@fC.edu
AAAAAAAALGAAAAAA	Charles	Robinson	Nancy.Mccormick@DA26I9ZArLF9rxJ6Z.edu
AAAAAAAALGAAAAAA	Charles	Robinson	Edith.Hernandez@BNHL0k.com
AAAAAAAALGAAAAAA	Charles	Robinson	Margaret.Collins@9obPr3UV.org
AAAAAAAALGAAAAAA	Charles	Robinson	Pamela.Luna@QBGuhL36lnA.edu
AAAAAAAALGAAAAAA	Charles	Robinson	William.Craig@prVDE1E8AHc.org
AAAAAAAALGAAAAAA	Charles	Robinson	Brandon.Woods@hjKbf.edu
AAAAAAAALGAAAAAA	Charles	Robinson	Marie.Peterson@1zg9tydFHafA5.com
AAAAAAAALGAAAAAA	Charles	Robinson	Anthony.Fisher@jJrZfeDcz8P.com
AAAAAAAALGAAAAAA	Charles	Robinson	Dwight.Schneider@koxO7zAysvOd.com
AAAAAAAALGAAAAAA	Charles	Robinson	Ha.Carpenter@XgcUt4svNz.com
AAAAAAAALGAAAAAA	Charles	Robinson	Neil.Cox@FRuR2bFK.com
AAAAAAAALGAAAAAA	Charles	Robinson	Maxine.Carlson@StyP5lAokmQ29QHYMLa.edu
AAAAAAAALGAAAAAA	Charles	Robinson	Deborah.Burton@xt.edu
AAAAAAAALGAAAAAA	Charles	Robinson	William.Warner@zegnrzurU.org
AAAAAAAALGAAAAAA	Charles	Robinson	Mabel.Richmond@Tkla.edu

https://superset.hpe-staging-ezaf.com/superset/welcome/

D – Create a Dashboard and Add the Chart

To create a dashboard and add the chart you created to the dashboard:

1. In Superset, click the **Dashboards** tab.
2. Click **+DASHBOARD**.
3. Enter a name (title) for the dashboard, for example **Customer Dashboard**.
4. In the right navigation bar, click the **LAYOUT ELEMENTS** tab.
5. Drag and drop the **Header** element into the dashboard.
6. In the **Header** element, enter a title, for example **Customers Named Charles**.
7. In the right navigation bar, click the **CHARTS** tab.
8. Locate the chart you created (Customer Info) and drag and drop the chart into the dashboard. You may need to drag the chart over the Header title and drop it there to get it to stay in place. A blue line appears in the dashboard when the chart is in a place it can be dropped.
9. Click **SAVE** to save the dashboard.



Customer Dashboard

Customers Named Charles

Customer Info

c_customer_id	c_first_name	c_last_name	c_email_address
AAAAAAAALGAAAAAA	Charles	Robinson	Javier.Lewis@V
AAAAAAAALGAAAAAA	Charles	Robinson	Amy.Moses@O
AAAAAAAALGAAAAAA	Charles	Robinson	Latisha.Hamilto
AAAAAAAALGAAAAAA	Charles	Robinson	Michael.White@
AAAAAAAALGAAAAAA	Charles	Robinson	Robert.Moran@
AAAAAAAALGAAAAAA	Charles	Robinson	Brunilda.Sharp(
AAAAAAAALGAAAAAA	Charles	Robinson	Ollie.Shipman@
AAAAAAAALGAAAAAA	Charles	Robinson	Karl.Gilbert@C
AAAAAAAALGAAAAAA	Charles	Robinson	Albert.Brunson
AAAAAAAALGAAAAAA	Charles	Robinson	Betty.Williams@

Customer Info
Viz type Table
Dataset cust_schema.customer_info_view
Modified 2 minutes ago

State Wise per Call Center Sales
Viz type Bar Chart
Dataset retailstore.qf_retailstore_view
Modified 13 days ago

Call Center Wise Sales of Different Jewelr...
Viz type Sunburst Chart
Dataset retailstore.qf_retailstore_view
Modified 13 days ago

Monthly Itemwise Sales Across Years
Viz type Area Chart
Dataset retailstore.qf_retailstore_view
Modified 13 days ago

Total Sale by State
Viz type Country Map

End of Tutorial

You have completed this tutorial. This tutorial demonstrated the integration of the HPE Ezmeral Unified Analytics Software SQL query engine (EzPresto on page 294) with Superset to visualize data models that you create in the Data Engineering space using the charting and dashboarding features in Superset.

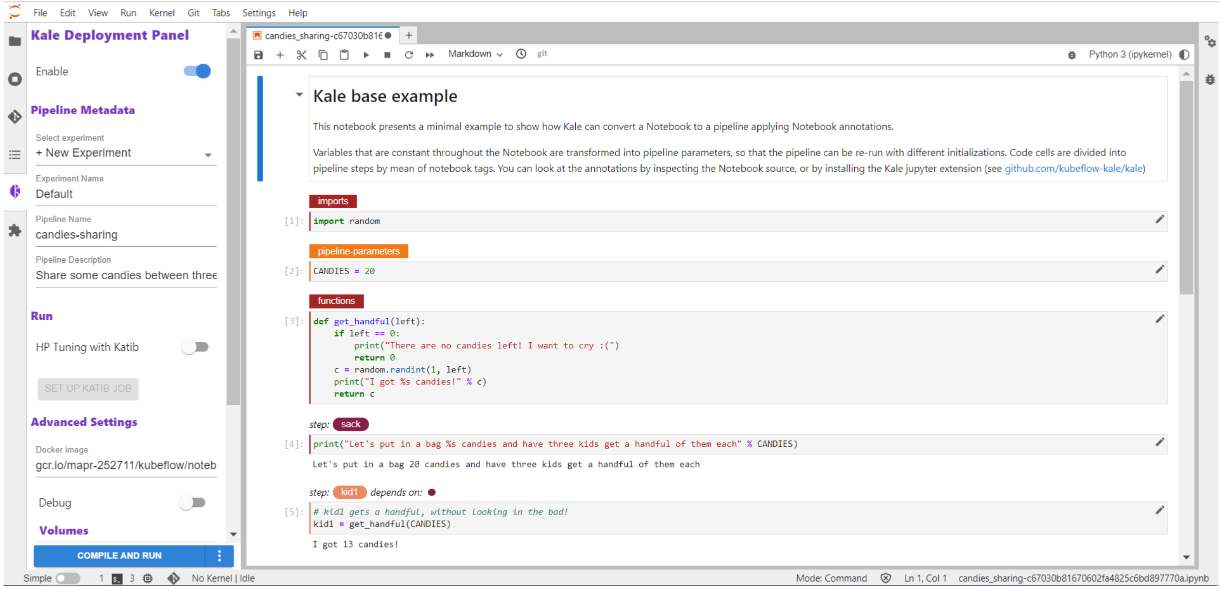
You may also be interested in the Retail Store Analysis Dashboard (Superset) on page 43, which shows you how to create a database connection, visualize data, and monitor queries used in visualizations.

Candy Sharing Tutorial (Kale)

Describes how Kale converts Notebook to pipeline by applying notebook annotations.

Upload candies_sharing.ipynb if you do not already have the file.

1. Log in to the Kubeflow notebook.
2. Upload `candies_sharing.ipynb` using the **Upload Files** button inside the Kubeflow notebook.
3. Open the `candies_sharing.ipynb` file and enable the Kale extension.
4. Run all cells in the notebook using `Run -> Run All Cells`.
5. At the bottom of `Kale Deployment Panel`, select `COMPILE AND RUN` and then click the button.


File Edit View Run Kernel Git Tabs Settings Help
candies_sharing-c67030b816
Markdown
git
Python 3 (ipykernel)
COMPILE AND RUN
Simple
No Kernel | Idle
Mode: Command
Ln 1, Col 1
candies_sharing-c67030b81670602fa4825c6bd897770a.ipynb

Kale Deployment Panel

Enable

Pipeline Metadata

Select experiment: + New Experiment

Experiment Name: Default

Pipeline Name: candies-sharing

Pipeline Description: Share some candies between three

Run

HP Tuning with Katib

SET UP KATIB JOB

Advanced Settings

Docker image: gcr.io/mapr-252711/kubeflow/noteb

Debug

Volumes

Kale base example

This notebook presents a minimal example to show how Kale can convert a Notebook to a pipeline applying Notebook annotations.

Variables that are constant throughout the Notebook are transformed into pipeline parameters, so that the pipeline can be re-run with different initializations. Code cells are divided into pipeline steps by mean of notebook tags. You can look at the annotations by inspecting the Notebook source, or by installing the Kale jupyter extension (see github.com/kubeflow-kale/kale)

imports

```
[1]: import random
```

pipeline-parameters

```
[2]: CANDIES = 20
```

functions

```
[3]: def get_handful(left):
         if left == 0:
             print("There are no candies left! I want to cry :(")
             return 0
         c = random.randint(1, left)
         print("I got %s candies!" % c)
         return c
```

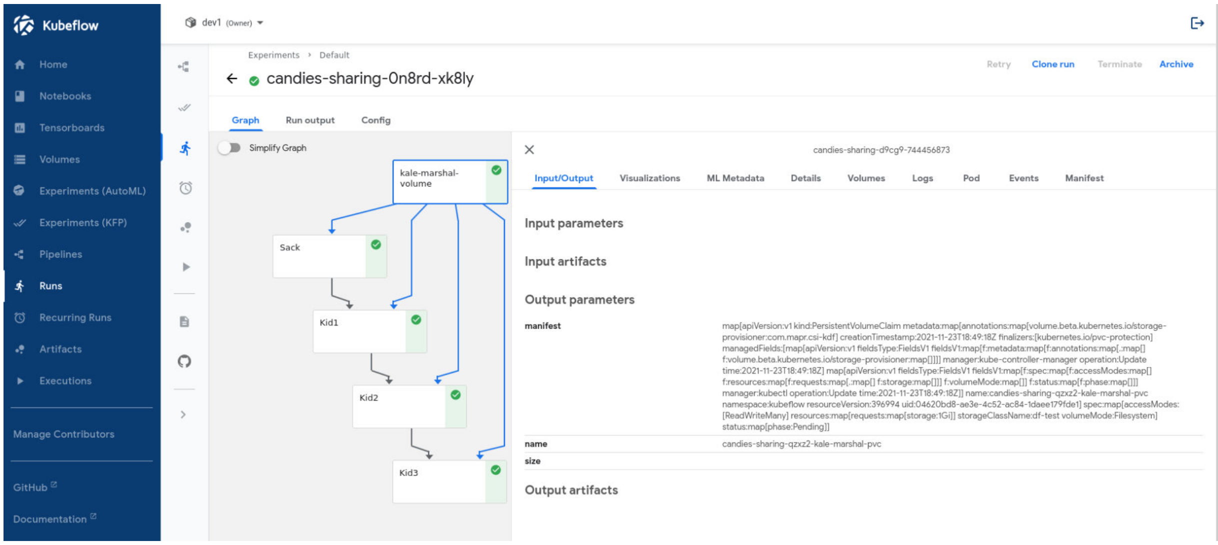
step: sack

```
[4]: print("Let's put in a bag %s candies and have three kids get a handful of them each" % CANDIES)
     Let's put in a bag 20 candies and have three kids get a handful of them each
```

step: kid1 depends on: ●

```
[5]: # kid1 gets a handful, without looking in the bad!
     kid1 = get_handful(CANDIES)
     I got 13 candies!
```

6. Open the Kubeflow Dashboard from the Runs page and check the status of the pipeline run launched from the Kubeflow notebook.



Input parameters

Input artifacts

Output parameters

manifest	map[apiVersion:v1 kind:PersistentVolumeClaim metadata:map[annotations:map[volume.beta.kubernetes.io/storage-provisioner:com.mapr.csi-kdf] creationTimestamp:2021-11-23T18:49:18Z finalizers:[kubernetes.io/pvc-protection] managedFields:[map[apiVersion:v1 fieldsType:FieldsV1 fieldsV1:map[f:metadata:map[f:annotations:map[.:map[] f:volume.beta.kubernetes.io/storage-provisioner:map[]]]] manager:kube-controller-manager operation:Update time:2021-11-23T18:49:18Z] map[apiVersion:v1 fieldsType:FieldsV1 fieldsV1:map[f:spec:map[f:accessModes:map[] f:resources:map[f:requests:map[.:map[] f:storage:map[]]] f:volumeMode:map[]] f:status:map[f:phase:map[]]] manager:kubectl operation:Update time:2021-11-23T18:49:18Z]] name:candies-sharing-qzxz2-kale-marshal-pvc namespace:kubeflow resourceVersion:396994 uid:04620bd8-ae3e-4c52-ac84-1daee179fde1] spec:map[accessModes:[ReadWriteMany] resources:map[requests:map[storage:1Gi]] storageClassName:df-test volumeMode:Filesystem] status:map[phase:Pending]]
name	candies-sharing-qzxz2-kale-marshal-pvc
size	

Output artifacts

More information

https://github.com/kubeflow-kale/kale

Feast Ride Sharing Use Case

Provides an end-to-end workflow using Feast in HPE Ezmeral Unified Analytics Software to generate training data and perform online model inference for the ride-sharing driver satisfaction model.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software.

About this task

Use Feast to generate training data and perform online model inference for the ride-sharing driver satisfaction model.

In this tutorial, you will:

1. Deploy a local feature store with a Parquet file offline store and SQLite online store.
2. Build a training dataset using time series features from Parquet files.
3. Read the latest features from the offline store for batch scoring.
4. Ingest batch features ("materialization") and streaming features into the online store.
5. Read the latest features from the online store for real-time inference.
6. Explore the Feast web interface to see Data Sources, Entities, Feature Views, Feature Services, and Datasets that are defined through feature definitions.

Procedure

1. Connect to the notebook server. See Creating and Managing Notebook Servers on page 436.
2. In the `<username>/Feast` folder, open the `ride-sharing-example.ipynb` file.

   NOTE:

   If you do not see the `Feast` folder in the `<username>` folder, copy the folder from the `shared/ezua-tutorials/current-release/Data-Science/` directory into the `<username>` folder. The `shared` directory is accessible to all users. Editing or running examples from the `shared` directory is not advised. The `<username>` directory is specific to you and cannot be accessed by other users.

   If the `Feast` folder is not available in the `shared/ezua-tutorials/current-release/Data-Science/` directory:

   a. Go to GitHub repository for tutorials.

   b. Clone the repository.

   c. Navigate to `ezua-tutorials/Data-Science`.

   d. Navigate back to the `shared` directory.

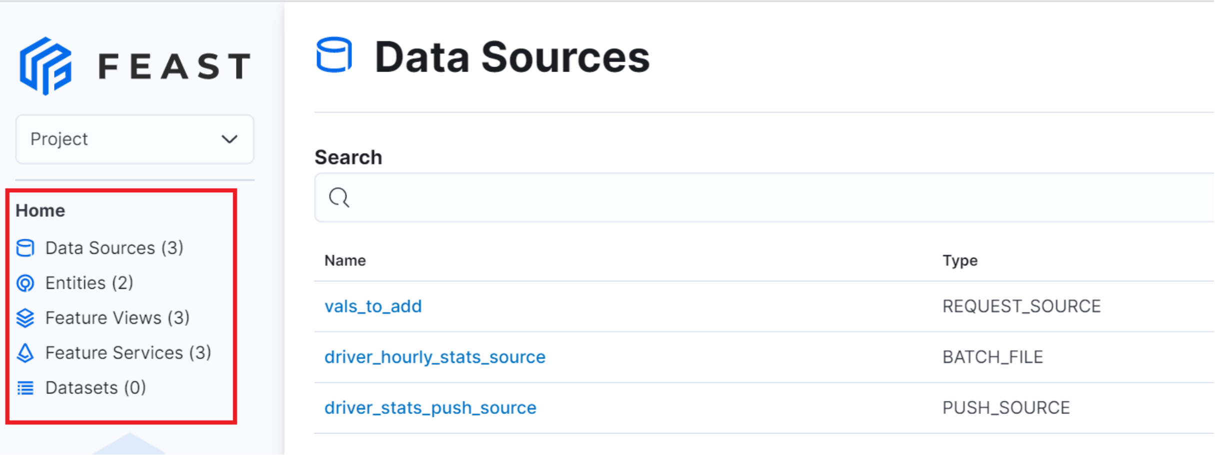
   e. Copy the `/Feast` folder from the `ezua-tutorials/Data-Science` repository into the `shared` directory.

   f. Copy the `/Feast` folder from the `shared` folder to the `<username>` directory.

3. Validate the `ride-sharing-example.ipynb` file, `definitions.py` file, and the `data` folder are available in the `/<username>/Feast` directory.
4. Validate the `driver_stats.parquet` file is available in the `<username>/Feast/data` directory.
5. Select the first cell of the `ride-sharing-example.ipynb` notebook, and click **Run Selected Cell and All Below**.

Results

1. Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Feast** tile under the **Data Science** tab and click **Open**.
2. Explore the Feast web interface to see Data Sources, Entities, Feature Views, Feature Services, and Datasets that are defined through feature definitions.



Name	Type
vals_to_add	REQUEST_SOURCE
driver_hourly_stats_source	BATCH_FILE
driver_stats_push_source	PUSH_SOURCE

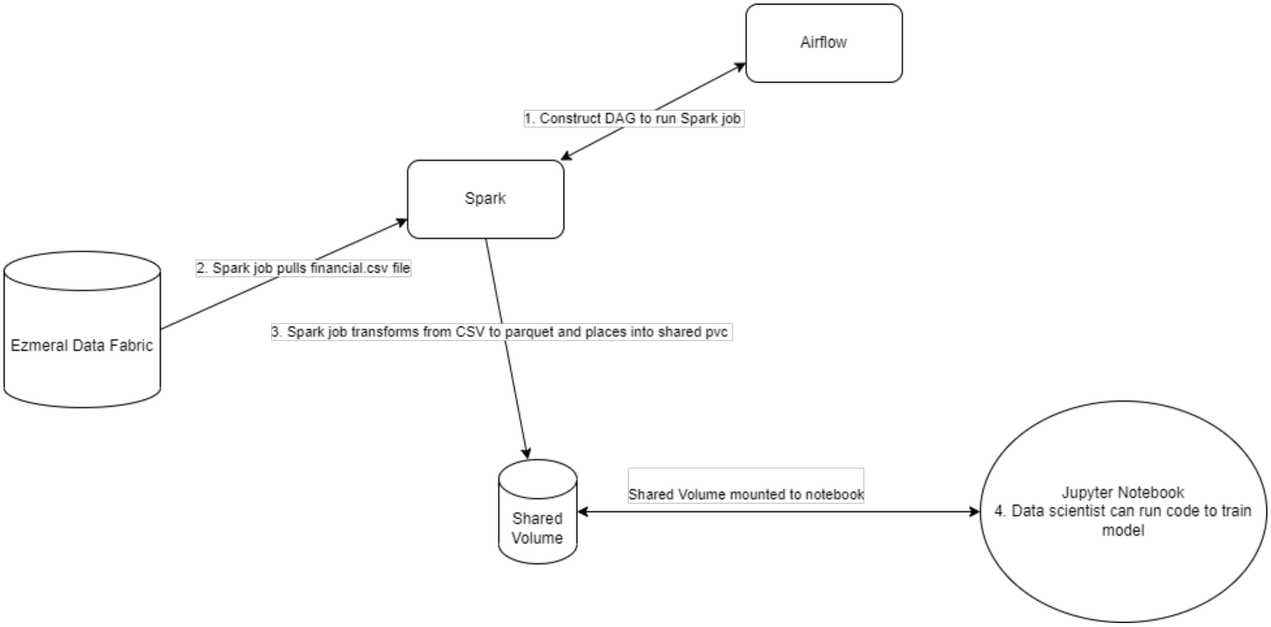
Financial Time Series Workflow

Describes how to use HPE Ezmeral Unified Analytics Software to run a Spark application from an Airflow DAG and then run a Jupyter notebook to analyze and visualize data that the Spark application puts into a shared directory in the shared volume that the data scientist’s notebook is mounted to.

Scenario

A DAG source (located in GitHub) is coded to submit a Spark job that pulls CSV data (`financial.csv`) from an S3 data source, transforms the data into Parquet format, and puts the data in a shared volume in the `financial-processed` folder.

The following diagram shows the components and applications in the workflow:


Airflow
1. Construct DAG to run Spark job
Spark
2. Spark job pulls financial.csv file
Ezmeral Data Fabric
3. Spark job transforms from CSV to parquet and places into shared pvc
Shared Volume
Shared Volume mounted to notebook
Jupyter Notebook
4. Data scientist can run code to train model

Steps

Sign in to HPE Ezmeral Unified Analytics Software and perform the following steps:

1. Prerequisites on page 22
2. Use Airflow to run a DAG that submits a Spark application.
3. View the Spark application that the DAG submitted.

4. Connect to and run the Jupyter notebook to analyze and visualize the data.

Prerequisites

Connect to your Jupyter notebook and perform setup tasks to prepare the environment to train the model. A `<username>` folder with a sample notebook file and SSL certificate is provided for the purpose of this tutorial. To connect your notebook and perform setup tasks, follow these steps:

For Administrators

An administrator must create an S3 object store bucket and load data. The Spark application reads raw data from the local-S3 Object Store.

To copy the required datasets to `ezaf-demo` bucket at `data/`, run:

```
# Code to copy finacial.csv from
ezua-tutorials to ezaf-demo bucket
at data/ for Financial Time Series
Example
import boto3
s3 = boto3.client("s3", verify=False)
bucket = 'ezaf-demo'
source_file
= '/mnt/shared/ezua-tutorials/
current-release/Data-Science/Kubeflow/
Financial-Time-Series/dataset/
financial.csv'
dest_object = 'data/financial.csv'
# Check whether bucket
is already created
buckets = s3.list_buckets()
bucket_exists = False
available_buckets = buckets["Buckets"]
for available_bucket in
available_buckets:
    if available_bucket["Name"]
== bucket:
        bucket_exists = True
        break
# Create bucket if not exists
if not bucket_exists:
    s3.create_bucket(Bucket=bucket)
# Upload file
s3.upload_file(Filename=source_file,
Bucket=bucket, Key=dest_object)
```

If you are an administrator completing these tutorials, after finishing the administrator steps, make sure to complete steps for member users.

For Members

Before completing these steps as a member user, ask the administrator to complete the steps for administrator users.

1. In the HPE Ezmeral Unified Analytics Software, go to **Tools & Frameworks**.
2. Select the **Data Science** tab and then click **Open** in the **Kubeflow** tile.
3. In Kubeflow, click **Notebooks** to open the notebooks page.

4. Click **Connect** to connect to your notebook server.
5. Go to the `/<username>` folder.
6. Copy the template `object_store_secret.yaml.tpl` file from the `shared/ezua-tutorials/current-release/Data-Analytics/Spark` directory to the `<username>` folder.
7. In the `<username>/Financial-Time-Series` folder, open the `financial_time_series_example.ipynb` file.

   NOTE: If you do not see the `Financial-Time-Series` folder in the `<username>` folder, copy the folder from the `shared/ezua-tutorials/current-release/Data-Science/Kubeflow` directory into the `<username>` folder. The `shared` directory is accessible to all users. Editing or running examples from the `shared` directory is not advised. The `<username>` directory is specific to you and cannot be accessed by other users.

   If the `Financial-Time-Series` folder is not available in the `shared/ezua-tutorials/current-release/Data-Science/Kubeflow` directory, perform:

   a. Go to GitHub repository for tutorials.
   b. Clone the repository.
   c. Navigate to `ezua-tutorials/Data-Science/Kubeflow`.
   d. Navigate back to the `<username>` directory.
   e. Copy the `Financial-Time-Series` folder from the `ezua-tutorials/Data-Science/Kubeflow` directory into the `<username>` directory.

8. To generate a secret to read data source files from S3 bucket by Spark application (Airflow DAG), run the first cell of the `financial_time_series_example.ipynb` file:

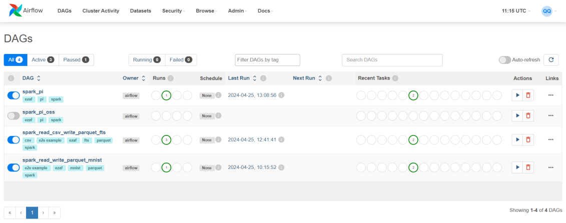
```
import kfp
kfp_client = kfp.Client()
namespace =
kfp_client.get_user_namespace()
!sed -e
"s/\$AUTH_TOKEN/$AUTH_TOKEN/" /mnt/
user/object_store_secret.yaml.tpl
> object_store_secret.yaml
```

A - Run a DAG in Airflow

In Airflow, run the DAG named `spark_read_csv_write_parquet_fts`. The DAG runs a Spark application that reads CSV data `(financial.csv)` from an S3 bucket, transforms the data into Parquet format, and writes the transformed Parquet data into the shared volume.

Run the DAG

1. Navigate to the Airflow screen using either of the following methods:
   - Click **Data Engineering > Airflow Pipelines**.
   - Click **Tools & Frameworks**, select the **Data Engineering** tab, and click **Open** in the **Airflow** tile.
2. In **Airflow**, verify that you are on the **DAGs** screen.
3. Click `spark_read_csv_write_parquet_fts` DAG.



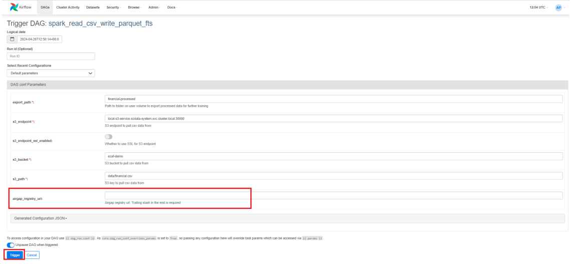
DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
spark_pi ezaf pi spark	airflow	1	None	2024-04-25, 13:08:56		2	▶ 🗑	...
spark_pi_oss ezaf pi spark	airflow		None				▶ 🗑	...
spark_read_csv_write_parquet_fts csv e2e example ezaf fts parquet spark	airflow	3	None	2024-04-25, 12:41:41		2	▶ 🗑	...
spark_read_write_parquet_mnist e2e example ezaf mnist parquet spark	airflow	1	None	2024-04-25, 10:15:52		2	▶ 🗑	...

Showing 1-4 of 4 DAGs

NOTE: The DAG is pulled from a pre-configured HPE GitHub repository. This DAG is constructed to submit a Spark application that pulls `financial.csv` file into Parquet format, and places the converted files in a shared directory. If you want to use your private GitHub repository, see Airflow DAGs Git Repository on page 355 to learn how to configure your repository.

4. Click **Code** to view the DAG code.
5. Click **Graph** to view the graphical representation of the DAG.

6. Click**Trigger DAG**(play button) to open a screen where you can configure parameters.



Trigger DAG: spark_read_csv_write_parquet_fts

Logical date: 2024-04-26T12:50:14+00:0

Run id (Optional): Run ID

Select Recent Configurations: Default parameters

DAG conf Parameters

export_path *:	financial-processed Path to folder on user volume to export processed data for further training
s3_endpoint *:	local-s3-service.ezdata-system.svc.cluster.local:30000 S3 endpoint to pull csv data from
s3_endpoint_ssl_enabled:	Whether to use SSL for S3 endpoint
s3_bucket *:	ezaf-demo S3 bucket to pull csv data from
s3_path *:	data/financial.csv S3 key to pull csv data from
airgap_registry_url:	Airgap registry url. Trailing slash in the end is required

Generated Configuration JSON

To access configuration in your DAG use {{ dag_run.conf }}. As core.dag_run_conf_overrides_params is set to True, so passing any configuration here will override task params which can be accessed via {{ params }}.

Unpause DAG when triggered

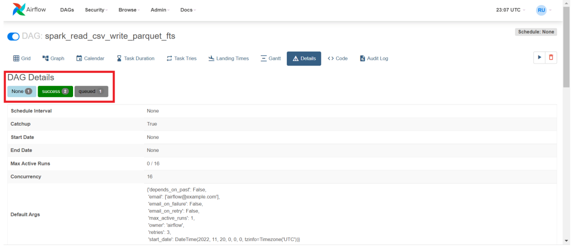
7. (Air-gapped environment only) Specify the airgap registry URL.

8. Click the **Trigger** button located at the bottom-left of the screen.

   Upon successful DAG completion, the data is accessible inside your notebook server in the following directory for further processing:

```
shared/financial-processed"
```

9. To view details for the DAG, click **Details**. Under **DAG Details**, you can see green, red, and/or yellow buttons with the number of times the DAG ran successfully or failed.



DAG: spark_read_csv_write_parquet_fts

DAG Details

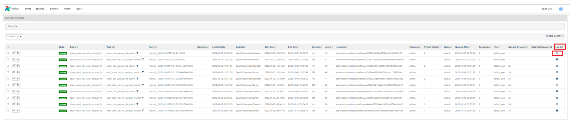
None 1 success 2 queued 1

Schedule Interval	None
Catchup	True
Start Date	None
End Date	None
Max Active Runs	0 / 16
Concurrency	16
Default Args	{'depends_on_past': False, 'email': ['airflow@example.com'], 'email_on_failure': False, 'email_on_retry': False, 'max_active_runs': 1, 'owner': 'airflow', 'retries': 3, 'start_date': DateTime(2022, 11, 20, 0, 0, 0, tzinfo=Timezone('UTC'))}

10. Click the **Success** button.

11. To find your job, sort by **End Date** to see the latest jobs that have run, and then scroll to the right and click the log icon under Log URL for that run. Note that jobs run with the configuration:

```
Conf "username":"your_username"
```



When running Spark applications using Airflow, you can see the following logs:

```
Reading from s3a://ezaf-demo/data/
financial.csv;
src format is csv 22/11/04
11:53:26 WARN
AmazonHttpClient: SSL Certificate
```

```
checking for endpoints has been
explicitly disabled.
Read complete Writing to file:///
mounts/data/financial-processed;
dest format is parquet Write
complete
```

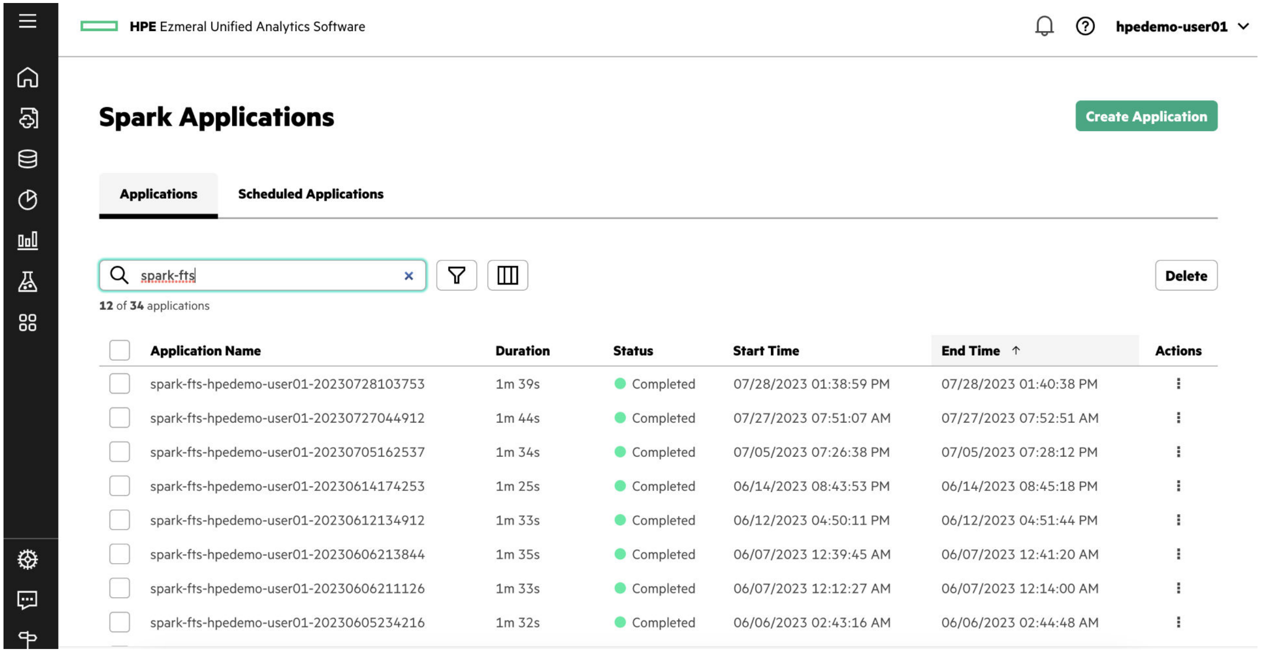
IMPORTANT: The cluster clears the logs that result from the DAG runs. The duration after which the cluster clears the logs depends on the Airflow task, cluster configuration, and policy.

B – View the Spark Application

Once you have triggered the DAG, you can view the Spark application in the **Spark Applications** screen.

To view the Spark application, go to **Analytics > Spark Applications** from the left navigation bar.

Alternatively, you can go to **Tools & Frameworks** and then click on the **Analytics** tab. On the **Analytics** tab, select the **Spark Operator** tile and click **Open**.



12 of 34 applications

□	Application Name	Duration	Status	Start Time	End Time ↑	Actions
□	spark-fts-hpedemo-user01-20230728103753	1m 39s	● Completed	07/28/2023 01:38:59 PM	07/28/2023 01:40:38 PM	⋮
□	spark-fts-hpedemo-user01-20230727044912	1m 44s	● Completed	07/27/2023 07:51:07 AM	07/27/2023 07:52:51 AM	⋮
□	spark-fts-hpedemo-user01-20230705162537	1m 34s	● Completed	07/05/2023 07:26:38 PM	07/05/2023 07:28:12 PM	⋮
□	spark-fts-hpedemo-user01-20230614174253	1m 25s	● Completed	06/14/2023 08:43:53 PM	06/14/2023 08:45:18 PM	⋮
□	spark-fts-hpedemo-user01-20230612134912	1m 33s	● Completed	06/12/2023 04:50:11 PM	06/12/2023 04:51:44 PM	⋮
□	spark-fts-hpedemo-user01-20230606213844	1m 35s	● Completed	06/07/2023 12:39:45 AM	06/07/2023 12:41:20 AM	⋮
□	spark-fts-hpedemo-user01-20230606211126	1m 33s	● Completed	06/07/2023 12:12:27 AM	06/07/2023 12:14:00 AM	⋮
□	spark-fts-hpedemo-user01-20230605234216	1m 32s	● Completed	06/06/2023 02:43:16 AM	06/06/2023 02:44:48 AM	⋮

C – Run the Jupyter Notebook

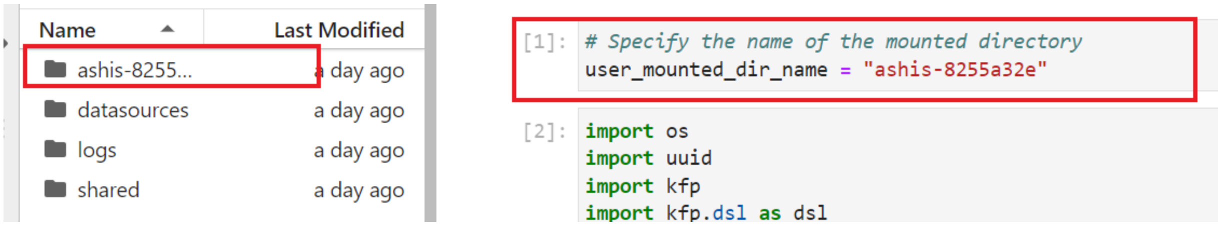
Run the Jupyter notebook file to analyze and visualize the financial time series data.

To run the notebook:

1. Connect to the notebook server. See Creating and Managing Notebook Servers on page 436.
2. In the **Notebooks** screen, navigate to the `shared/financial-processed/` folder to validate that the data processed by the Spark application is available.
3. In the `<username>/Financial-Time-Series/` folder, open the `financial_time_series_example.ipynb` file.

4. In the sixth cell of the `financial_time_series_example.ipynb` file, update the user folder name as follows:

```
user_mounted_dir_name = "<username-folder-name>"
```



Name	Last Modified
ashis-8255...	a day ago
datasources	a day ago
logs	a day ago
shared	a day ago

```
[1]: # Specify the name of the mounted directory
     user_mounted_dir_name = "ashis-8255a32e"

[2]: import os
     import uuid
     import kfp
     import kfp.dsl as dsl
```

5. In the Notebook Launcher, select the second cell of the notebook and click **Run the selected cells and advance** (play icon).

6. After the packages install, restart the notebook kernel. To restart the kernel, click the **Restart the kernel** button or select **Kernel > Restart Kernel** in the menu bar at the top of the screen.

7. After the kernel restarts, click into the second cell and select **Run the selected cells and All Below**.

8. Review the results of each notebook cell to analyze and visualize the data.

End of Tutorial

You have completed this tutorial. This tutorial demonstrated that you can use Airflow, Spark, and Notebooks in Unified Analytics to extract, transform, and load data into a shared volume and then run analytics and visualize the transformed data.

MLflow Bike Sharing Use Case

Provides an end-to-end workflow in HPE Ezmeral Unified Analytics Software for an MLflow prediction model to determine bike rentals per hour based on weather and time.

Scenario

A data scientist wants to use a Jupyter Notebook to train a model that predicts how many bikes will be rented every hour based on weather and time information.

HPE Ezmeral Unified Analytics Software includes the following components and applications to support this scenario:

Dataset	Bike sharing dataset, `bike-sharing.csv`, available in the `/shared/mlflow` directory.
Notebook (Jupyter)	Two preconfigured Jupyter notebooks: • `bike-sharing-mlflow.ipynb` - Runs code, trains models, finds the best model. • `bike-sharing-prediction.ipynb` - Predicts based on the model; deployed via KServe.
MLflow	• Tracks the experiment and trainings/runs. • Logs artifacts, metrics, and parameters for each run. • Registers the best model

Object Storage	Stores artifacts that result after running each experiment.
KServe Deployment	Downloads and deploys a model from object storage and makes the model accessible through a web service endpoint.

Steps

Sign in to HPE Ezmeral Unified Analytics Software and perform the following steps:

Run the Bike Sharing Use Case on page 28

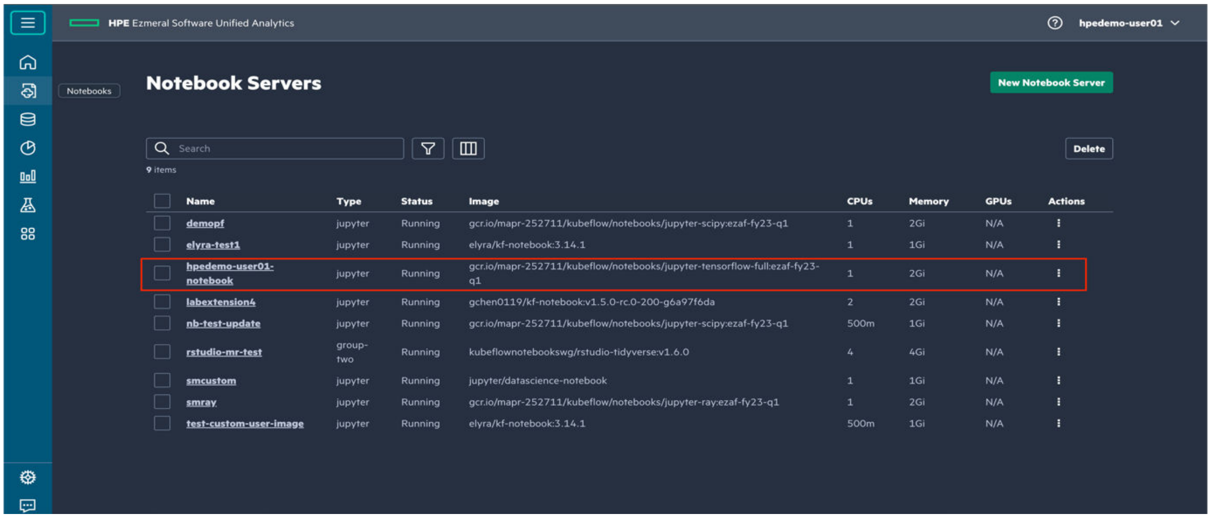
Track Experiment, Runs, and Register a Model in MLflow on page 29

Use the Model for Prediction on page 30

Run the Bike Sharing Use Case

1. In the left navigation pane, click **Notebooks**.

2. Connect to your notebook server instance. For this example, select `hpedemo-user01-notebook`.



Notebook Servers

9 items

□	Name	Type	Status	Image	CPUs	Memory	GPUs	Actions
□	demopf	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezaf-fy23-q1	1	2Gi	N/A	⋮
□	elyra-test1	jupyter	Running	elyra/kf-notebook:3.14.1	1	1Gi	N/A	⋮
□	hpedemo-user01-notebook	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezaf-fy23-q1	1	2Gi	N/A	⋮
□	labextension4	jupyter	Running	gchen0119/kf-notebook:v1.5.0-rc.0-200-g6a97f6da	2	2Gi	N/A	⋮
□	nb-test-update	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezaf-fy23-q1	500m	1Gi	N/A	⋮
□	rstudio-mr-test	group-two	Running	kubeflownotebookswg/rstudio-tidyverse:v1.6.0	4	4Gi	N/A	⋮
□	smcustom	jupyter	Running	jupyter/datascience-notebook	1	1Gi	N/A	⋮
□	smray	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-ray:ezaf-fy23-q1	1	2Gi	N/A	⋮
□	test-custom-user-image	jupyter	Running	elyra/kf-notebook:3.14.1	500m	1Gi	N/A	⋮

3. Copy the `MLFlow` folder from the `shared` directory into the `/user` directory.

NOTE:

If the `Mlflow` folder is not available in the `shared` directory, perform:

a. Go to GitHub repository for tutorials.

b. Clone the repository.

c. Navigate to `ezua-tutorials/Data-Science`.

d. Navigate back to the `shared` directory.

e. Copy the `MLflow` folder from the `ezua-tutorials/Data-Science` repository into the `shared` directory.

f. Copy the `/MLflow` folder from `shared` folder to `user` directory.

4. Open `bike-sharing-mlflow.ipynb` and import mlflow and install libraries. After you finish, restart the kernel and run all the cells, including those you previously ran.

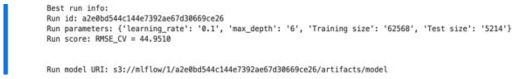
   NOTE:

   If you are using the local s3-proxy, do not set the following environment variables for MLflow. However, if you are trying to connect from outside the cluster, you must set the following environment variables.

   ```
   os.environ["AWS_ACCESS_KEY_ID"] = os.environ['MLFLOW_TRACKING_TOKEN']
   os.environ["AWS_SECRET_ACCESS_KEY"] = "s3"
   os.environ["AWS_ENDPOINT_URL"] = 'http://
   local-s3-service.ezdata-system.svc.cluster.local:30000'
   os.environ["MLFLOW_S3_ENDPOINT_URL"] = os.environ["AWS_ENDPOINT_URL"]
   os.environ["MLFLOW_S3_IGNORE_TLS"] = "true"
   os.environ["MLFLOW_TRACKING_INSECURE_TLS"] = "true"
   ```

5. Run the notebook cells.

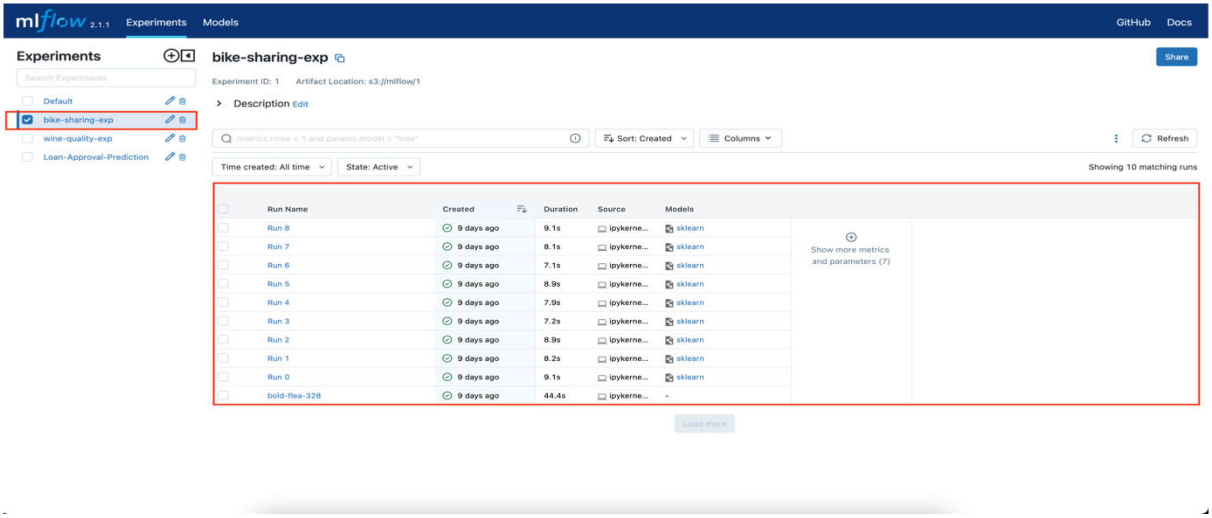
   Running the notebook returns the details of the best model:



```
Best run info:
Run id: a2e0bd544c144e7392ae67d30669ce26
Run parameters: {'learning_rate': '0.1', 'max_depth': '6', 'Training size': '62568', 'Test size': '5214'}
Run score: RMSE_CV = 44.9510


Run model URI: s3://mlflow/1/a2e0bd544c144e7392ae67d30669ce26/artifacts/model
```

Track Experiment, Runs, and Register a Model in MLflow

1. Navigate to the MLflow UI. You should see the bike-sharing-exp experiment.



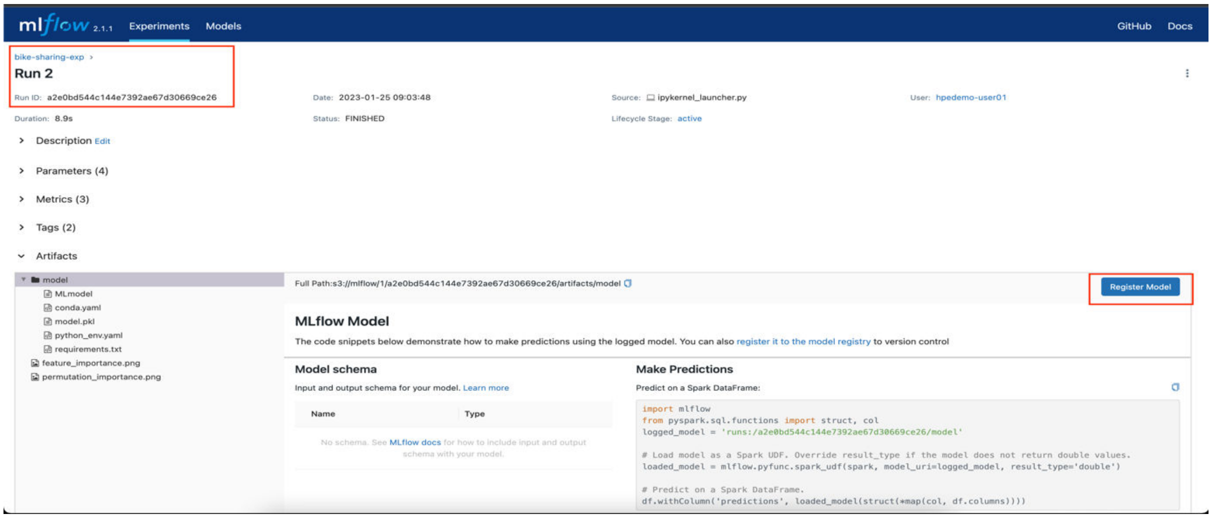
bike-sharing-exp

Experiment ID: 1 Artifact Location: s3://mlflow/1

Showing 10 matching runs

Run Name	Created	Duration	Source	Models
Run 8	9 days ago	9.1s	ipykerne...	sklearn
Run 7	9 days ago	8.1s	ipykerne...	sklearn
Run 6	9 days ago	7.1s	ipykerne...	sklearn
Run 5	9 days ago	8.9s	ipykerne...	sklearn
Run 4	9 days ago	7.9s	ipykerne...	sklearn
Run 3	9 days ago	7.2s	ipykerne...	sklearn
Run 2	9 days ago	8.9s	ipykerne...	sklearn
Run 1	9 days ago	8.2s	ipykerne...	sklearn
Run 0	9 days ago	9.1s	ipykerne...	sklearn
bold-flea-328	9 days ago	44.4s	ipykerne...	-

2. Select the best model and then select **Register Model**. In this example, the best model is run 2.



bike-sharing-exp ›

Run 2

Run ID: a2e0bd544c144e7392ae67d30669ce26 Date: 2023-01-25 09:03:48 Source: ipykernel_launcher.py User: hpedemo-user01

Duration: 8.9s Status: FINISHED Lifecycle Stage: active

› Description Edit

› Parameters (4)

› Metrics (3)

› Tags (2)

⌄ Artifacts

- model
  - MLmodel
  - conda.yaml
  - model.pkl
  - python_env.yaml
  - requirements.txt
- feature_importance.png
- permutation_importance.png

Full Path:s3://mlflow/1/a2e0bd544c144e7392ae67d30669ce26/artifacts/model

Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also register it to the model registry to version control

Model schema

Input and output schema for your model. Learn more

Name	Type

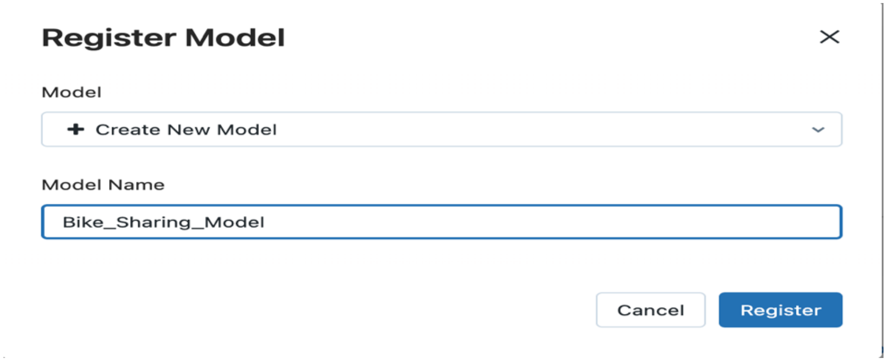
No schema. See MLflow docs for how to include input and output schema with your model.

Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/a2e0bd544c144e7392ae67d30669ce26/model'

# Load model as a Spark UDF. Override result_type if the model does not return double values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='double')

# Predict on a Spark DataFrame.
df.withColumn('predictions', loaded_model(struct(*map(col, df.columns))))
```

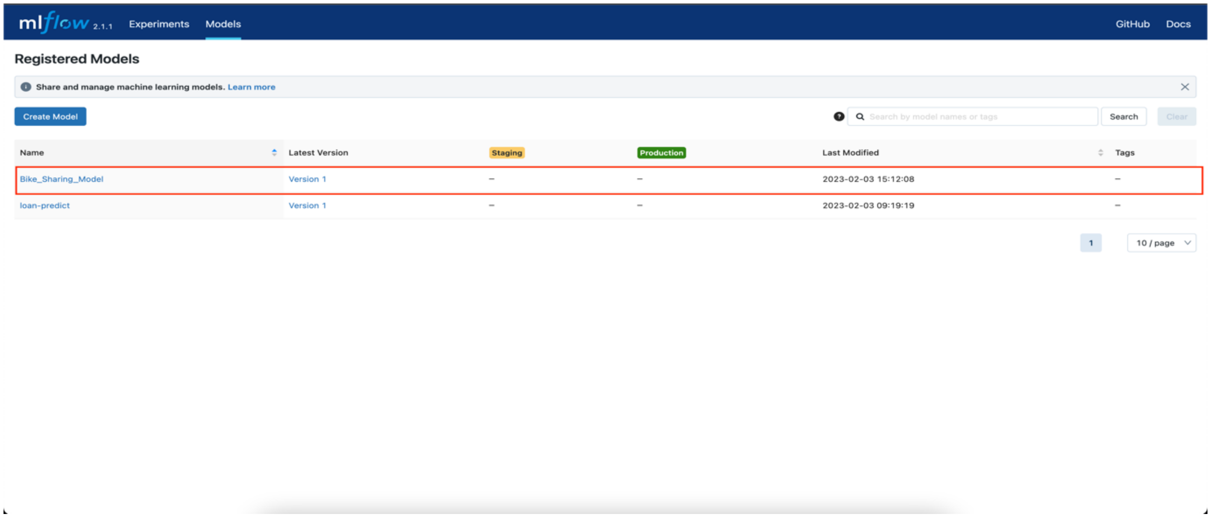
3. In the Register Model window, enter Bike_Sharing_Model and click **Register**.



Model Name

Bike_Sharing_Model

4. Click on the **Models** menu to view the registered models.



Registered Models

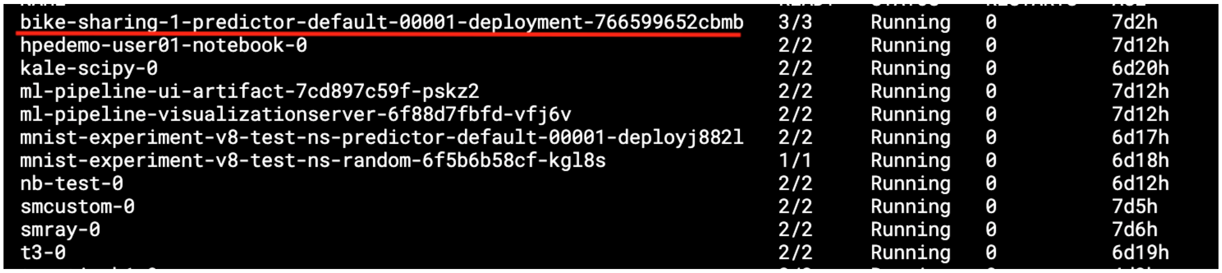
Share and manage machine learning models. Learn more

Name	Latest Version	Staging	Production	Last Modified	Tags
Bike_Sharing_Model	Version 1	–	–	2023-02-03 15:12:08	–
loan-predict	Version 1	–	–	2023-02-03 09:19:19	–

Use the Model for Prediction

1. Navigate to the notebook server and open `bike-sharing-prediction.ipynb`.

2. Run the first cell and wait until the `bike-sharing-predictor` pod goes into the running state.



```
bike-sharing-1-predictor-default-00001-deployment-766599652cbmb   3/3   Running   0   7d2h
hpedemo-user01-notebook-0                                          2/2   Running   0   7d12h
kale-scipy-0                                                       2/2   Running   0   6d20h
ml-pipeline-ui-artifact-7cd897c59f-pskz2                           2/2   Running   0   7d12h
ml-pipeline-visualizationserver-6f88d7fbfd-vfj6v                   2/2   Running   0   7d12h
mnist-experiment-v8-test-ns-predictor-default-00001-deployj882l    2/2   Running   0   6d17h
mnist-experiment-v8-test-ns-random-6f5b6b58cf-kgl8s                1/1   Running   0   6d18h
nb-test-0                                                          2/2   Running   0   6d12h
smcustom-0                                                         2/2   Running   0   7d5h
smray-0                                                            2/2   Running   0   7d6h
t3-0                                                               2/2   Running   0   6d19h
```

3. Run the second cell to deploy machine learning model using KServe inference service. Note: Update DOMAIN_NAME to your domain for external access and save changes.

   The system prints the following predictions for the input:

   ```
   Rented Bikes Per Hours:
   Input Data: {'season': 1, 'year': 2, 'month': 1, 'hour_of_day': 0,
   'is_holiday': 0, 'weekday': 6, 'is_workingday': 0, 'weather_situation':
   1, 'temperature': 0.24, 'feels_like_temperature': 0.2879, 'humidity':
   0.81, 'windspeed': 0.0}
   Bike Per Hour: 108.90178471846806
   Input Data: {'season': 1, 'year': 5, 'month': 1, 'hour_of_day': 0,
   'is_holiday': 0, 'weekday': 6, 'is_workingday': 1, 'weather_situation':
   1, 'temperature': 0.24, 'feels_like_temperature': 0.2879, 'humidity':
   0.81, 'windspeed': 0.0}
   Bike Per Hour: 84.96339548602367
   ```

End of Tutorial

You have completed this tutorial. This tutorial demonstrated how to train a model using notebooks, track experiments and runs, log artifacts with MLFlow, and use KServe to deploy and predict models.

MNIST Digits Recognition Workflow

Provides an end-to-end workflow in HPE Ezmeral Unified Analytics Software for an MNIST digits recognition example.

Scenario

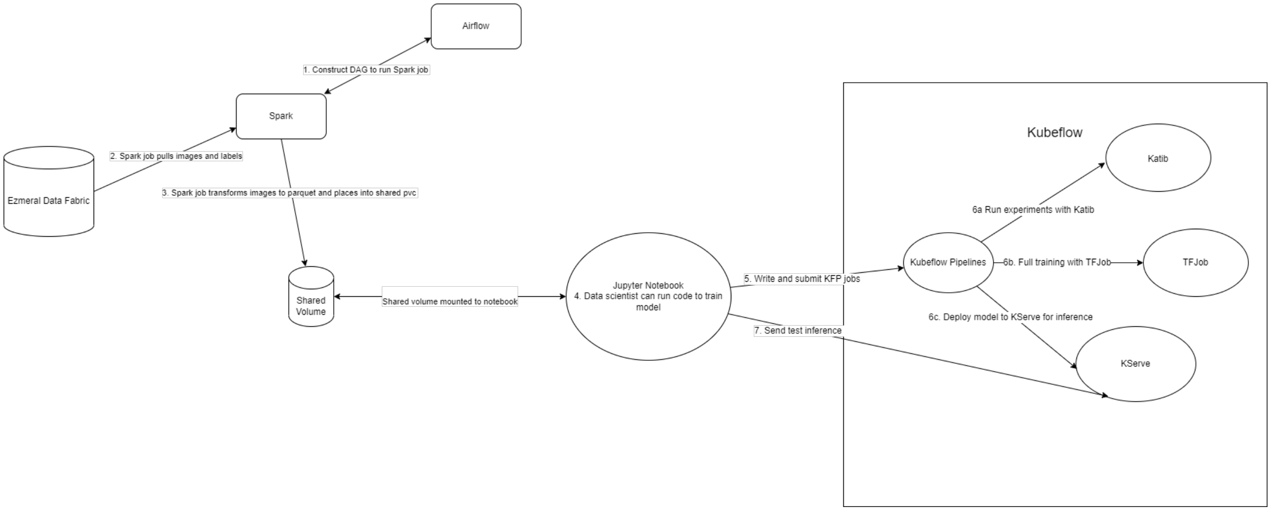
A data scientist wants to use a Jupyter Notebook to train a model that recognizes numbers in images. The image files reside in object storage and need to be transformed into Parquet format and put into a shared directory in the shared volume that the data scientist’s notebook is mounted to.

HPE Ezmeral Unified Analytics Software includes the following components and applications to support an end-to-end workflow for this scenario:

Spark	A Spark application pulls images from the HPE Ezmeral Data Fabric Object Store via MinIO endpoint, transforms the images into Parquet format, and puts the Parquet data into the shared directory in the shared volume.
Airflow	Coded Airflow DAG that runs the Spark application.
Notebook (Jupyter)	Preconfigured Jupyter notebook mounted to the shared volume to run code and train models for the following Kubeflow pipelines: • Run experiments with Katib to pick the best model and then deploy the model using KServe.

- Full training with TensorFlow jobs.

The following diagram shows the components and applications in the workflow:


Airflow
1. Construct DAG to run Spark job
Spark
2. Spark job pulls images and labels
Ezmeral Data Fabric
3. Spark job transforms images to parquet and places into shared pvc
Shared Volume
Shared volume mounted to notebook
Jupyter Notebook
4. Data scientist can run code to train model
5. Write and submit KFP jobs
7. Send test inference
Kubeflow
Katib
6a Run experiments with Katib
Kubeflow Pipelines
6b. Full training with TFJob
TFJob
6c. Deploy model to KServe for inference
KServe

Steps

Sign in to HPE Ezmeral Unified Analytics Software and perform the following steps:

- Prerequisites on page 32
- A - Run a DAG in Airflow on page 35
- B – View the Spark Application on page 38
- C- Update Path of Spark Generated Results on page 38
- D - Train the Model on page 38
- E - Serve the Model on page 39

Prerequisites

Connect to your Jupyter notebook and perform setup tasks to prepare the environment to train the model. A `<username>` folder with a sample notebook file and SSL certificate is provided for the purpose of this tutorial. To connect your notebook and perform setup tasks, follow these steps:

For Administrators

- An administrator must create an S3 object store bucket and load data as the Spark application reads raw data from the local-S3 Object Store.

  To copy the required datasets to `ezaf-demo` bucket at `data/mnist`, run:

  ```
  # Code to copy mnist digit dataset
  from ezua-tutorials to ezaf-demo
  bucket at data/mnist for Digit
  Recognition Example
  import os, boto3
  s3 = boto3.client("s3",
  verify=False)
  bucket = 'ezaf-demo'
  source_dir = '/mnt/shared/
  ezua-tutorials/current-release/
  Data-Science/Kubeflow/
  MNIST-Digits-Recognition/dataset'
  dest_dir = 'data/mnist'
  # Get list of files under dataset
  dir
  dataset_list =
  os.listdir(source_dir)# Create
  source file path and destination
  object key strings
  source_files = []
  dest_objects = []
  for dataset in dataset_list:
      source_files.append(source_dir
  + '/' + dataset)
      dest_objects.append(dest_dir +
  '/' + dataset)# check whether
  bucket is already created
  buckets = s3.list_buckets()
  bucket_exists = False
  available_buckets =
  buckets["Buckets"]
  for available_bucket in
  available_buckets:
      if available_bucket["Name"] ==
  bucket:
          bucket_exists = True
          break
  if not bucket_exists:

  s3.create_bucket(Bucket=bucket)#
  Upload files
  for i in range(len(source_files)):

  s3.upload_file(Filename=source_file
  s[i], Bucket=bucket,
  Key=dest_objects[i])
  ```

- (Air-gapped environment only) Manually pull the following images and make them available in the local repository for clusters in an airgap network to run the example:

```
nikenano/launchernew:latest
quay.io/aipipeline/
kserve-component:v0.10.1
```

Next, add your local airgap repository path prefix to the previous mentioned images in the following YAML files:

- `component/kubeflow-launcher-component.yaml`
- `component/kserve-component.yaml`

NOTE: If you are an administrator completing these tutorials, after finishing the administrator steps, make sure to complete the prerequisite steps for member users.

For Members

Before completing these steps as a member user, ask the administrator to complete the steps for administrator users.

1. In the HPE Ezmeral Unified Analytics Software, go to **Tools & Frameworks**.
2. Select the **Data Science** tab and then click **Open** in the **Kubeflow** tile.
3. In Kubeflow, click **Notebooks** to open the notebooks page.
4. Click **Connect** to connect to your notebook server.
5. Go to the `<username>` folder.
6. Copy the template `object_store_secret.yaml.tpl` file from the `shared/ezua-tutorials/current-release/Data-Analytics/Spark` directory to the `<username>` folder.
7. In the `<username>/MNIST-Digits-Recognition` folder, open the `mnist_katib_tf_kserve_example.ipynb` file.

NOTE: If you do not see the `MNIST-Digits-Recognition` folder in the `<username>` folder, copy the folder from the `shared/ezua-tutorials/current-release/Data-Science/Kubeflow` directory into the `<username>` folder. The `shared` directory is accessible to all users. Editing or running examples from the `shared` directory is not advised. The `<username>` directory is specific to you and cannot be accessed by other users

If the `MNIST-Digits-Recognition` folder is not available in the `shared/ezua-tutorials/current-release/Data-Science/Kubeflow` directory, perform:

- **a.** Go to GitHub repository for tutorials.
- **b.** Clone the repository.
- **c.** Navigate to `ezua-tutorials/Data-Science/Kubeflow`.
- **d.** Navigate back to the `<username>` directory.
- **e.** Copy the `MNIST-Digits-Recognition` folder from the `ezua-tutorials/Data-Science/Kubeflow` directory into the `<username>` directory.

8. To generate a secret to read data source files from S3 bucket by Spark application (Airflow DAG), run the first cell of the `mnist_katib_tf_kserve_example.ipynb` file:

   ```
   import kfp
   kfp_client = kfp.Client()
   namespace =
   kfp_client.get_user_namespace()
   !sed -e
   "s/\$AUTH_TOKEN/$AUTH_TOKEN/" /mnt/
   user/object_store_secret.yaml.tpl
   > object_store_secret.yaml
   ```

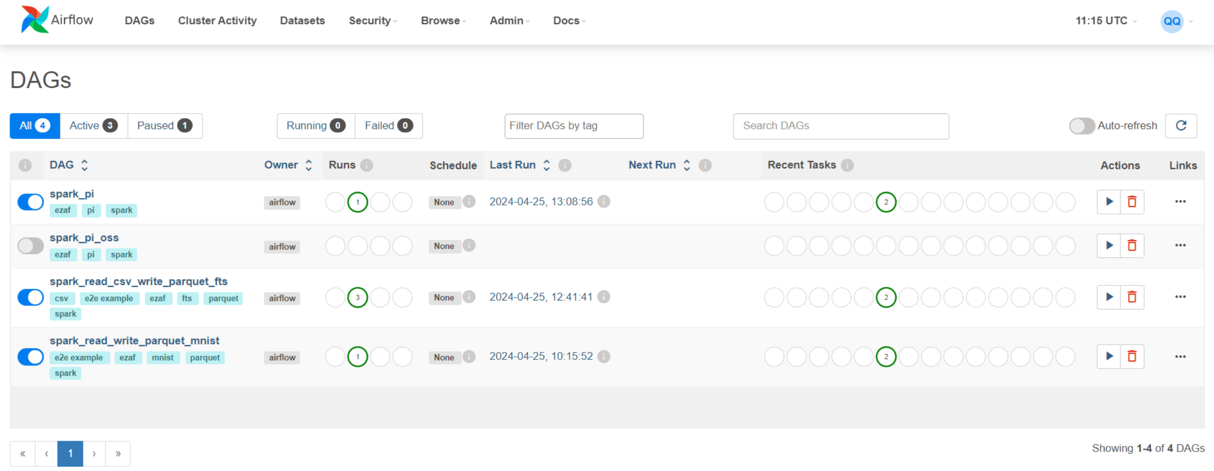
A - Run a DAG in Airflow

In Airflow, run the DAG named **spark_read_write_parquet_mnist**. The DAG runs a Spark application that pulls the images from object storage, transforms the data into Parquet format, and writes the transformed Parquet data into the shared volume.

1. Go to Airflow using either of the following methods:
   - Click **Data Engineering > Airflow Pipelines**.
   - Click **Tools & Frameworks**, select the **Data Engineering** tab, and click **Open** in the **Airflow** tile.

2. In **Airflow**, verify that you are on the **DAGs** tab.

3. Click on the **spark_read_write_parquet_mnist** DAG.



DAGs

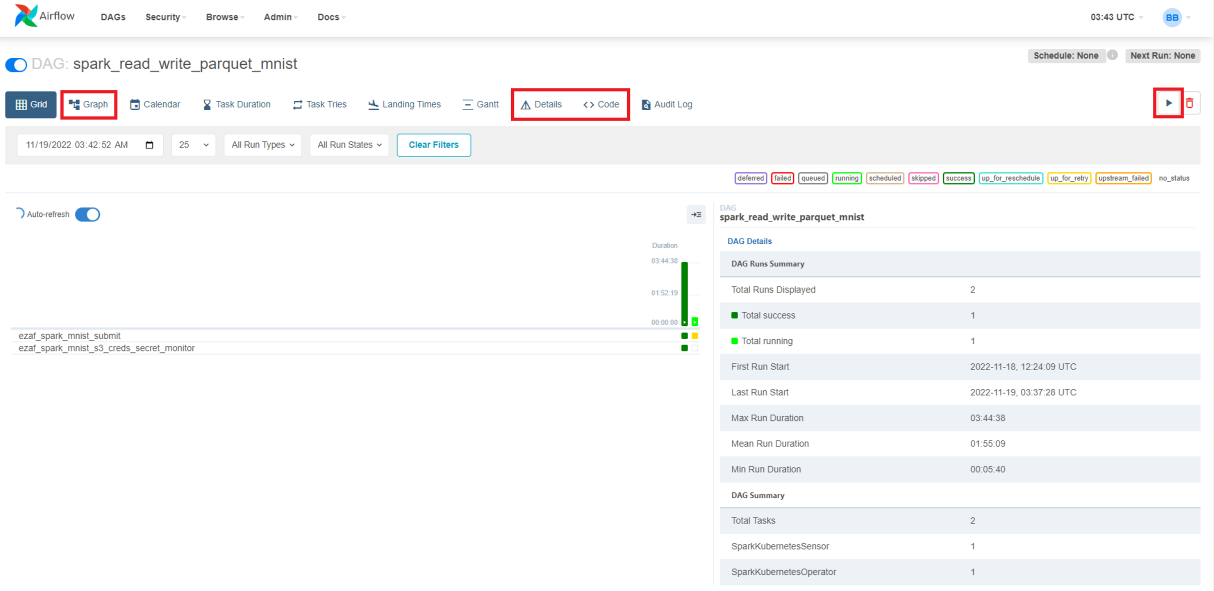
DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
spark_pi ezaf pi spark	airflow	1	None	2024-04-25, 13:08:56		2		
spark_pi_oss ezaf pi spark	airflow		None					
spark_read_csv_write_parquet_fts csv e2e example ezaf fts parquet spark	airflow	3	None	2024-04-25, 12:41:41		2		
spark_read_write_parquet_mnist e2e example ezaf mnist parquet spark	airflow	1	None	2024-04-25, 10:15:52		2		

« ‹ 1 › »

Showing **1-4** of **4** DAGs

NOTE: The DAG is pulled from a pre-configured HPE GitHub repository. This DAG is constructed to submit a Spark application that pulls ubyte.gz files from an object storage bucket, converts the images into Parquet format, and places the converted files in a shared directory. If you want to use your private GitHub repository, see Configuring Airflow on page 357 to find the steps to configure your repository.

4. Click **Code** to view the DAG code.



DAG: spark_read_write_parquet_mnist

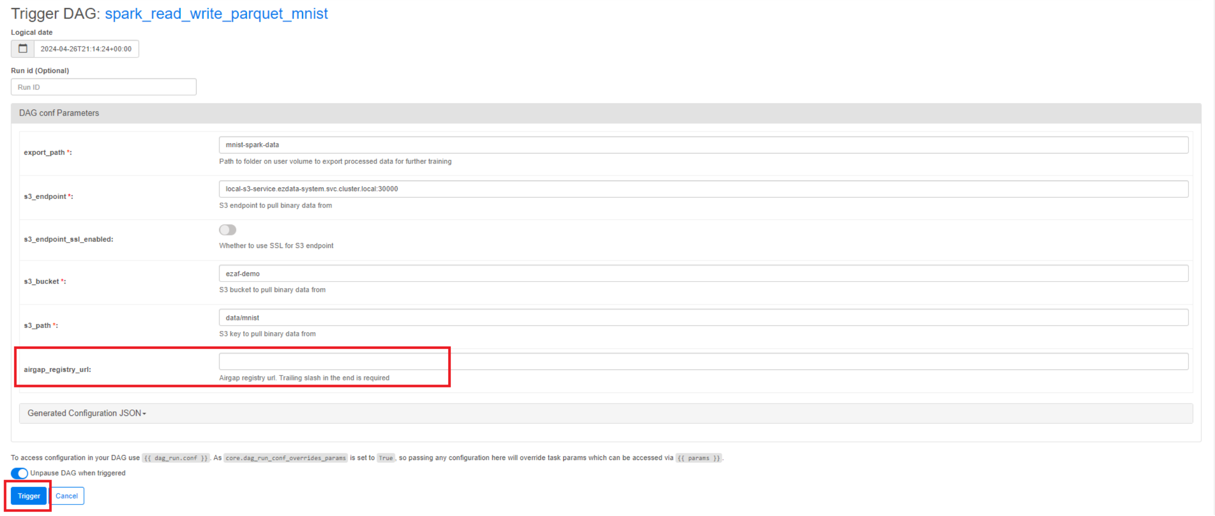
DAG
spark_read_write_parquet_mnist

DAG Details

DAG Runs Summary	
Total Runs Displayed	2
🟩 Total success	1
🟩 Total running	1
First Run Start	2022-11-18, 12:24:09 UTC
Last Run Start	2022-11-19, 03:37:28 UTC
Max Run Duration	03:44:38
Mean Run Duration	01:55:09
Min Run Duration	00:05:40
DAG Summary	
Total Tasks	2
SparkKubernetesSensor	1
SparkKubernetesOperator	1

5. Click **Graph** to view the graphical representation of the DAG.

6. Click**Trigger DAG**(play button) to open a screen where you can configure parameters.



Trigger DAG: spark_read_write_parquet_mnist

Logical date: 2024-04-26T21:14:24+00:00

Run id (Optional): Run ID

DAG conf Parameters

export_path *:	mnist-spark-data Path to folder on user volume to export processed data for further training
s3_endpoint *:	local-s3-service.ezdata-system.svc.cluster.local:30000 S3 endpoint to pull binary data from
s3_endpoint_ssl_enabled:	○ Whether to use SSL for S3 endpoint
s3_bucket *:	ezaf-demo S3 bucket to pull binary data from
s3_path *:	data/mnist S3 key to pull binary data from
airgap_registry_url:	 Airgap registry url. Trailing slash in the end is required

Generated Configuration JSON ▾

To access configuration in your DAG use `{{ dag_run.conf }}`. As `core.dag_run_conf_overrides_params` is set to `True`, so passing any configuration here will override task params which can be accessed via `{{ params }}`.

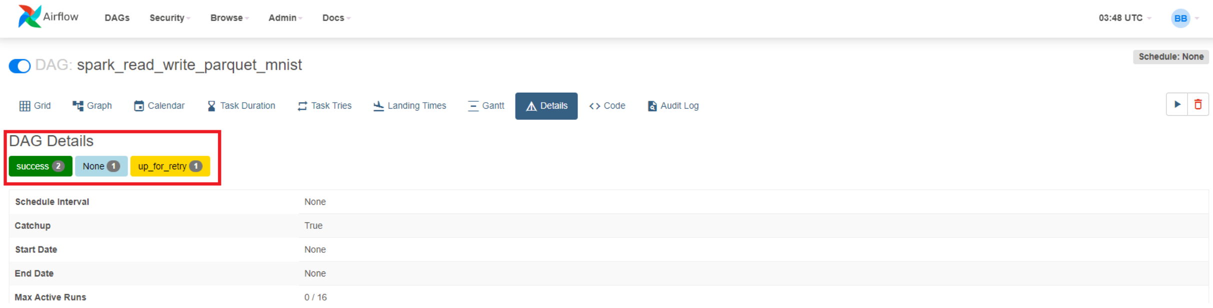
● Unpause DAG when triggered

7. (Air-gapped environment only)Specify the airgap registry URL.

8. Click the **Trigger** button on the bottom-left of the screen. Upon successful DAG completion, the data is accessible inside your notebook server by default in the following directory for further processing:

```
shared/mnist-spark-data/
```

9. To view details for the DAG, click **Details**. Under **DAG Details**, you can see green, red, and/or yellow buttons with the number of times the DAG ran successfully or failed.



DAG: spark_read_write_parquet_mnist

DAG Details

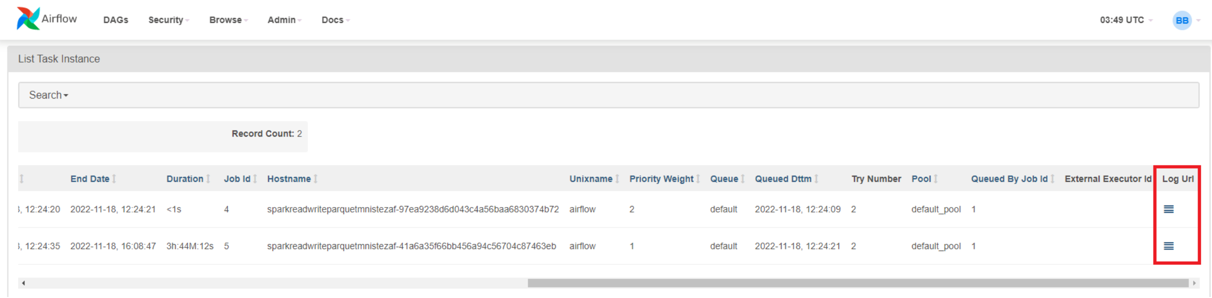
success 2 None 1 up_for_retry 1

Schedule Interval	None
Catchup	True
Start Date	None
End Date	None
Max Active Runs	0 / 16

10. Click the **Success** or **Failed** button.

11. To find your job, sort by **End Date** to see the latest jobs that have run, and then scroll to the right and click the log icon under Log URL for that run. Note that jobs run with the configuration:

```
Conf "username":"your_username"
```



List Task Instance

Record Count: 2

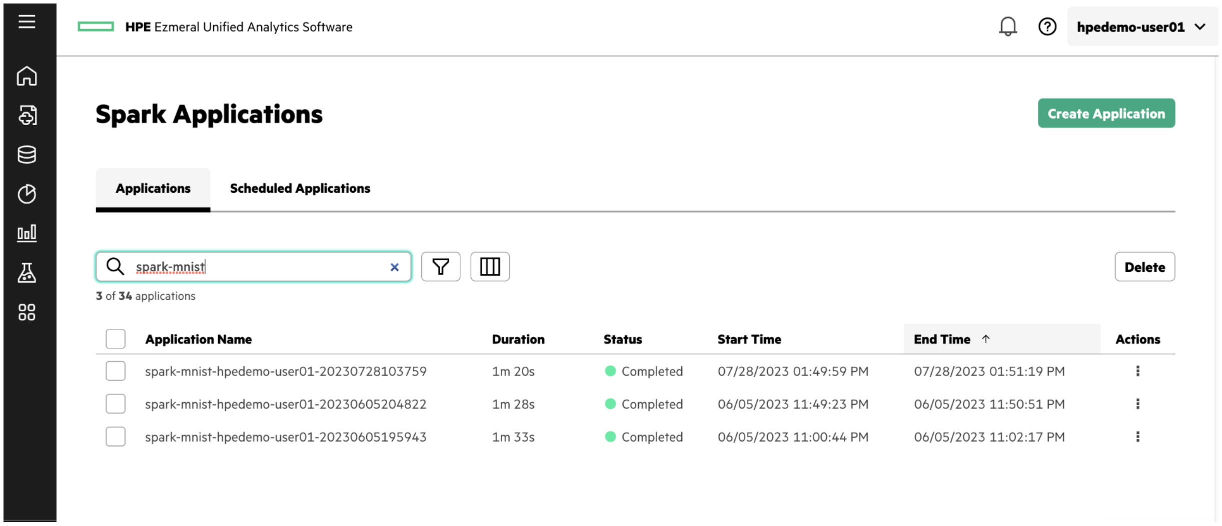
	End Date	Duration	Job Id	Hostname	Unixname	Priority Weight	Queue	Queued Dttm	Try Number	Pool	Queued By Job Id	External Executor Id	Log Url
, 12:24:20	2022-11-18, 12:24:21	<1s	4	sparkreadwriteparquetmnistezaf-97ea9238d6d043c4a56baa6830374b72	airflow	2	default	2022-11-18, 12:24:09	2	default_pool	1		≡
, 12:24:35	2022-11-18, 16:08:47	3h:44M:12s	5	sparkreadwriteparquetmnistezaf-41a6a35f66bb456a94c56704c87463eb	airflow	1	default	2022-11-18, 12:24:21	2	default_pool	1		≡

IMPORTANT: The cluster clears the logs that result from the DAG runs. The duration after which the cluster clears the logs depends on the Airflow task, cluster configuration, and policy.

B – View the Spark Application

After you run the DAG, you can view the status of the Spark application in the **Spark Applications** screen.

1. To view the Spark application, go to **Tools & Frameworks** and then click on the **Analytics** tab.
2. On the **Analytics** tab, select the **Spark Operator** tile and click **Open**.
3. Identify the **spark-mnist-<username>-<timestamp>** application, for example **spark-mnist-hpedemo-user01-20230728103759**, and view the status of the application..
4. Optionally, in the **Actions** column, click **View YAML**.



Spark Applications

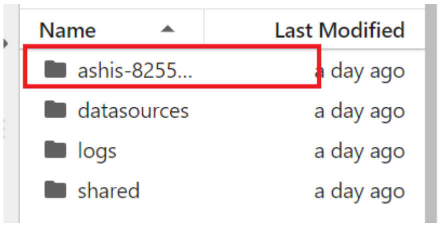
3 of 34 applications

	Application Name	Duration	Status	Start Time	End Time ↑	Actions
□	spark-mnist-hpedemo-user01-20230728103759	1m 20s	● Completed	07/28/2023 01:49:59 PM	07/28/2023 01:51:19 PM	⋮
□	spark-mnist-hpedemo-user01-20230605204822	1m 28s	● Completed	06/05/2023 11:49:23 PM	06/05/2023 11:50:51 PM	⋮
□	spark-mnist-hpedemo-user01-20230605195943	1m 33s	● Completed	06/05/2023 11:00:44 PM	06/05/2023 11:02:17 PM	⋮

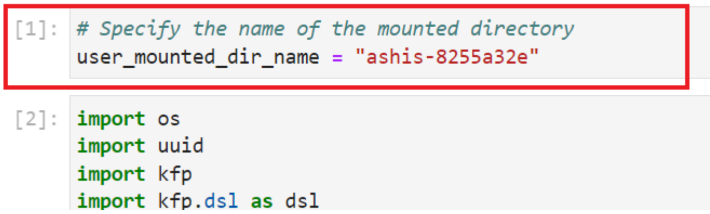
C- Update Path of Spark Generated Results

1. Open `mnist_katib_tf_kserve_example.ipynb` file.
2. In the third cell of the `mnist_katib_tf_kserve_example.ipynb` file, update the user folder name as follows:

```
user_mounted_dir_name = "<username-folder-name>"
```



Name ▲	Last Modified
ashis-8255...	a day ago
datasources	a day ago
logs	a day ago
shared	a day ago



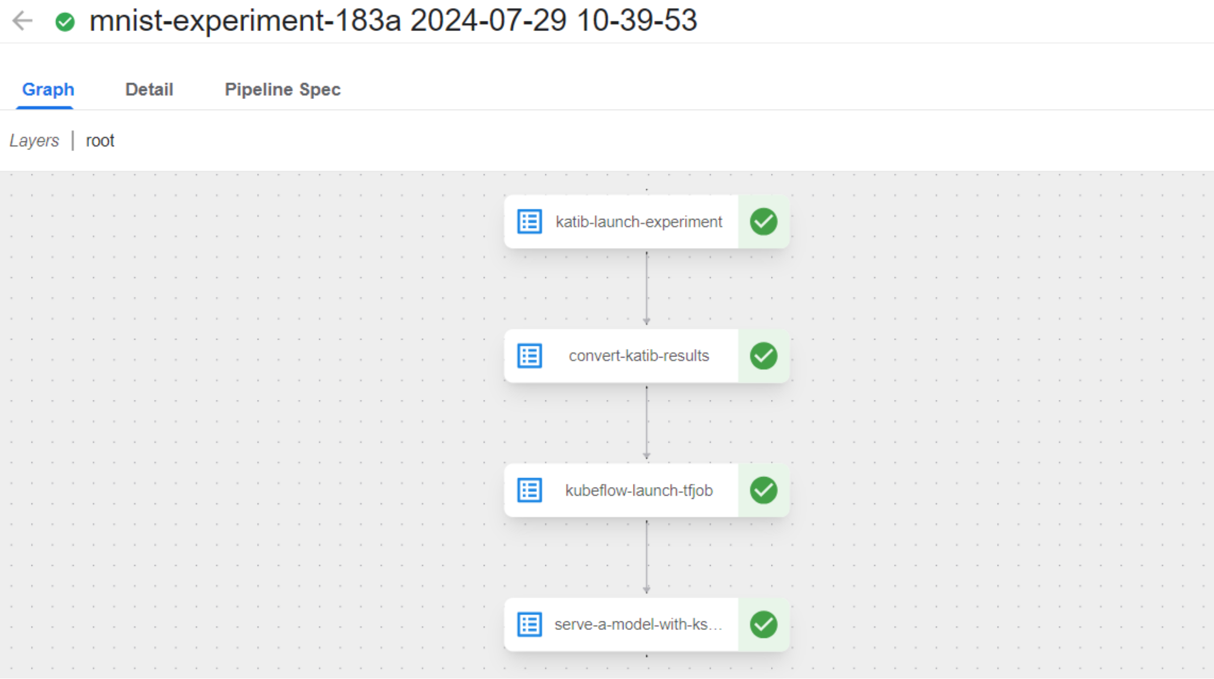
```
[1]: # Specify the name of the mounted directory
     user_mounted_dir_name = "ashis-8255a32e"

[2]: import os
     import uuid
     import kfp
     import kfp.dsl as dsl
```

D - Train the Model

To train the model:

1. In the Notebook Launcher, select the second cell of the notebook and select **Run-->Run Selected Cell and All Below**.
2. In the second to last cell, follow the `Run Details` link to open your Kubeflow Pipeline.

3. Run the Kubeflow pipeline in the UI and wait for it to successfully complete.

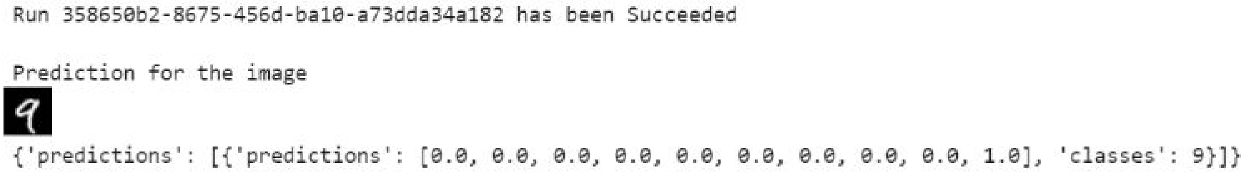


mnist-experiment-183a 2024-07-29 10-39-53

4. To get details about components created by the pipeline run, go to the **Experiments (AutoML)** and **Models** pages in the Kubeflow UI.

E - Serve the Model

To serve the model with KServe and get the prediction, wait for the the Kubeflow pipeline to successfully complete the run. The output displays the following results:



```
Run 358650b2-8675-456d-ba10-a73dda34a182 has been Succeeded

Prediction for the image

{'predictions': [{'predictions': [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0], 'classes': 9}]}
```

End of Tutorial

You have completed this tutorial. This tutorial demonstrated that you can use Airflow, Spark, and Notebooks in HPE Ezmeral Unified Analytics Software to extract, transform, and load data into a shared volume and then run analytics and train models using Kubeflow pipelines.

Rent Forecasting Model (Ray Serve)

Provides an end-to-end example for creating a notebook server and building a machine learning model to forecast rental prices, evaluate its accuracy, and deploy it for real-time predictions using Ray Serve in HPE Ezmeral Unified Analytics Software.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software.
- Verify that the installed Ray client and server versions match. To verify, complete the following steps in the terminal:

1. To switch to Ray's environment, run:

   ```
   source /opt/conda/etc/profile.d/conda.sh && conda activate ray
   ```

2. To verify that the Ray client and server versions match, run :

   ```
   ray --version
   ```

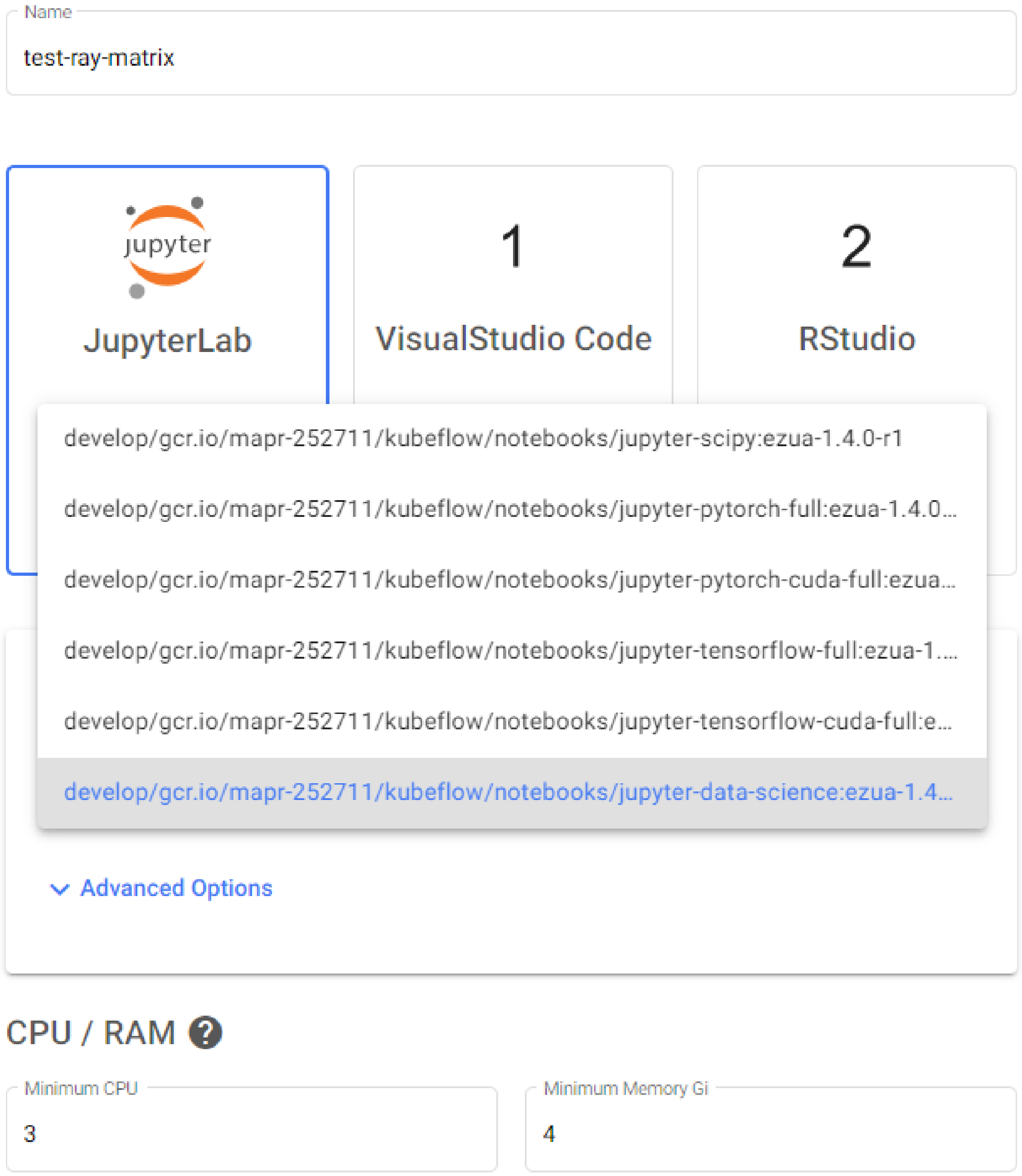
About this task

In this tutorial, you will complete the following steps:

1. Generate a synthetic dataset of rental properties with attributes such as square footage, number of bedrooms, number of bathrooms, and furnishing status to train the prediction model.
2. Format and input the generated data to the model for training.
3. Use the **Random Forest** model to predict the monthly rental prices.
4. Evaluate the predictive performance of the model using the **Mean Absolute Error (MAE)** on the testing data set.
5. Visualize the performance of the model using `matplotlib` which shows the graph for **Actual vs Predicted Rent Prices**.
6. After completing the model training process, save the model and deploy the model as a web service using Ray Serve. This allows you to manage request handling and scalability efficiently.
   a. First, initialize the Ray environment and start Ray Serve with the appropriate configuration settings to ensure smooth deployment and operation of service.
   b. Once Ray Serve is up and running, it manages the incoming `HTTP` requests and directs them to the deployed model for prediction.
7. View the deployed application in the **Ray Dashboard** under the **Serve** tab.
8. Wait for the deployed application to be in a *Running* state.
9. Send `HTTP` requests to the deployed model to obtain prediction results. These requests contain the input data that you want the model to make predictions on, and the response contains the corresponding predictions generated by the model.
10. After obtaining the prediction results, terminate deployment.

Procedure

1. Create a notebook server using the `jupyter-data-science` image with at least 3 CPUs and 4 Gi of memory in Kubeflow. See Creating and Managing Notebook Servers on page 436.



Name: test-ray-matrix

develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezua-1.4.0-r1
develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-full:ezua-1.4.0...
develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-cuda-full:ezua...
develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezua-1....
develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:e...
develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-data-science:ezua-1.4...

CPU / RAM ?

Minimum CPU: 3

Minimum Memory Gi: 4

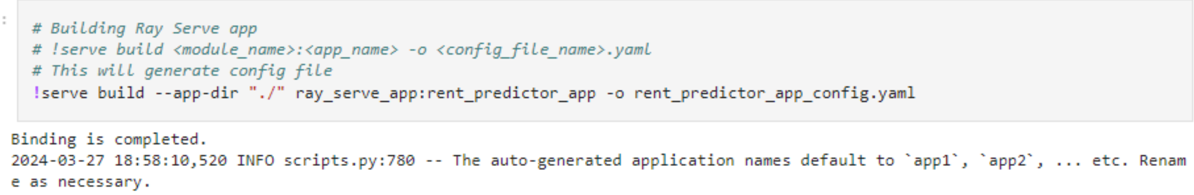
2. In your notebook environment, activate the Ray-specific Python kernel.

3. To ensure optimal performance, use dedicated directories containing only the essential files needed for that job submission as a working directory.

   For example, if you do not see the `Ray-Serve` folder in the `<username>` directory, copy the folder from the `shared/ezua-tutorials/current-release/Data-Science/Ray` directory into the `<username>` directory. The `shared` directory is accessible to all users. Editing or running examples from the `shared` directory is not advised. The `<username>` directory is specific to you and cannot be accessed by other users.

4. Open the `ray-serve-executor.ipynb` file in the `<username>/Ray-Serve` directory.

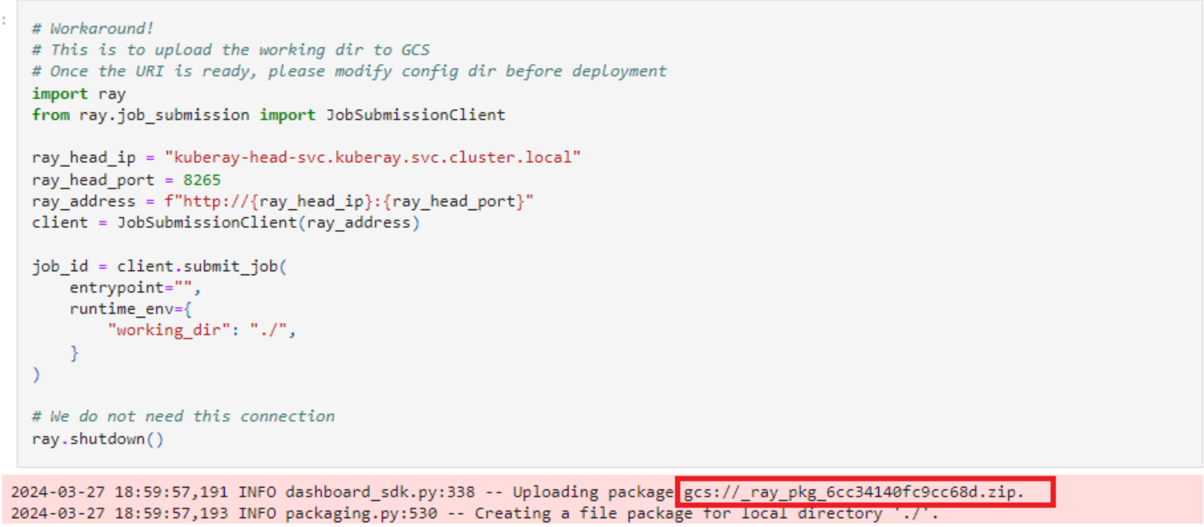
5. Select the first cell of the `ray-serve-executor.ipynb` notebook and click **Run the selected cells and advance (play icon)**. Continue until you run the following block of code which generates the `rent_predictor_app_config.yaml` configuration file.



```
# Building Ray Serve app
# !serve build <module_name>:<app_name> -o <config_file_name>.yaml
# This will generate config file
!serve build --app-dir "./" ray_serve_app:rent_predictor_app -o rent_predictor_app_config.yaml
```

```
Binding is completed.
2024-03-27 18:58:10,520 INFO scripts.py:780 -- The auto-generated application names default to `app1`, `app2`, ... etc. Renam
e as necessary.
```

6. Run the following block of code to generate URI. Currently, `serve deploy` does not directly support the `--working-dir` option. You must specify the generated URI from the output of this block of code in the `rent_predictor_app_config.yaml` file.



```
# Workaround!
# This is to upload the working dir to GCS
# Once the URI is ready, please modify config dir before deployment
import ray
from ray.job_submission import JobSubmissionClient

ray_head_ip = "kuberay-head-svc.kuberay.svc.cluster.local"
ray_head_port = 8265
ray_address = f"http://{ray_head_ip}:{ray_head_port}"
client = JobSubmissionClient(ray_address)

job_id = client.submit_job(
    entrypoint="",
    runtime_env={
        "working_dir": "./",
    }
)

# We do not need this connection
ray.shutdown()
```

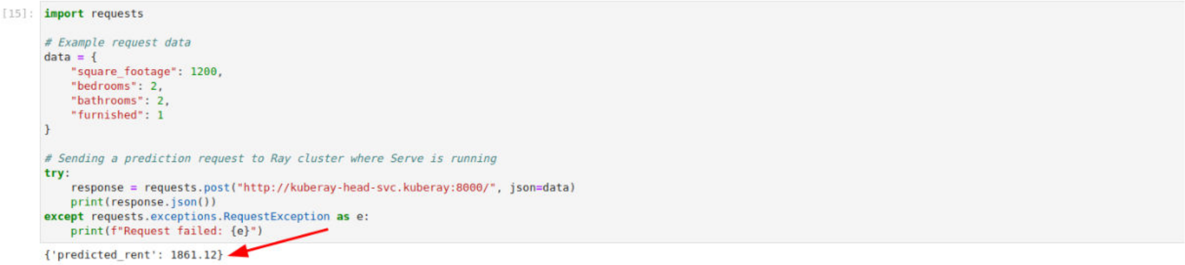
```
2024-03-27 18:59:57,191 INFO dashboard_sdk.py:338 -- Uploading package gcs://_ray_pkg_6cc34140fc9cc68d.zip.
2024-03-27 18:59:57,193 INFO packaging.py:530 -- Creating a file package for local directory './'.
```

7. Open the `rent_predictor_app_config.yaml` file.

8. Specify the generated URI in the `rent_predictor_app_config.yaml` file as follows:

```
runtime_env:
    working_dir: "<generated-URI>" #example: gcs://
_ray_pkg_fef565b457f470d9.zip
```

9. Navigate back to the `ray-serve-executor.ipynb` notebook file and continue to run cells until you reach the following block of code.



```
[15]: import requests

      # Example request data
      data = {
          "square_footage": 1200,
          "bedrooms": 2,
          "bathrooms": 2,
          "furnished": 1
      }

      # Sending a prediction request to Ray cluster where Serve is running
      try:
          response = requests.post("http://kuberay-head-svc.kuberay:8000/", json=data)
          print(response.json())
      except requests.exceptions.RequestException as e:
          print(f"Request failed: {e}")

      {'predicted_rent': 1861.12}
```

10. View the deployed application in **Ray Dashboard** under the **Serve** tab.
    a) Click the **Applications & Frameworks** icon on the left navigation bar.

b) Navigate to the **Ray** tile under the **Data Science** tab and click **Open**.

c) Navigate to the **Serve** tab.

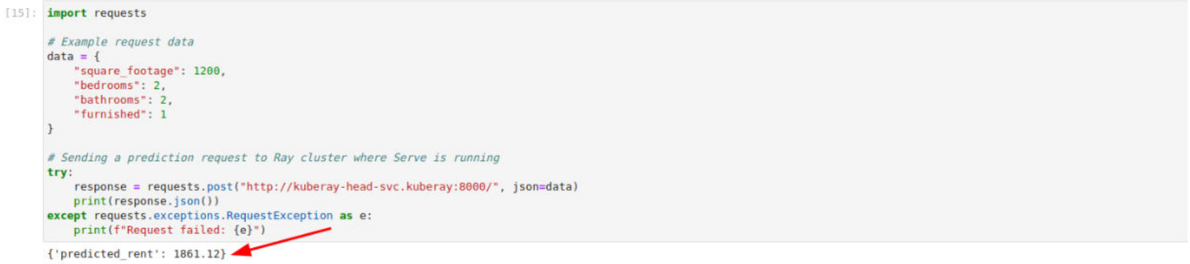
d) Locate and view your deployed application.

> **NOTE:** The auto-generated application names default to `app1`, `app2`, and so on. You can rename them as necessary.

e) Wait for the deployed application to be in a *Running* state.

11. Navigate back to the `ray-serve-executor.ipynb` notebook file.

12. Run the following block of code:



```
[15]: import requests

      # Example request data
      data = {
          "square_footage": 1200,
          "bedrooms": 2,
          "bathrooms": 2,
          "furnished": 1
      }

      # Sending a prediction request to Ray cluster where Serve is running
      try:
          response = requests.post("http://kuberay-head-svc.kuberay:8000/", json=data)
          print(response.json())
      except requests.exceptions.RequestException as e:
          print(f"Request failed: {e}")

      {'predicted_rent': 1861.12}
```

You will obtain the prediction result as *Predicted rent: 1861.12*.

13. After obtaining the prediction results, terminate deployment.

Results

This tutorial shows that by using Ray Serve and Ray cluster deployed in HPE Ezmeral Unified Analytics Software, you can efficiently deploy, manage, and scale your machine learning model as a web service to obtain prediction results.

Retail Store Analysis Dashboard (Superset)

Provides an end-to-end workflow example for a retail store analysis scenario in HPE Ezmeral Unified Analytics Software using EzPresto and Superset.

Scenario

A data analyst wants to visualize data sets from MySQL, SQL Server, and Hive data sources in Superset. The data analyst signs in to HPE Ezmeral Unified Analytics Software and connects Unified Analytics to MySQL, SQL Server, and Hive data sources. The data analyst runs a federated query against the data sets and then creates a view from the query. The analyst accesses the view from Superset and uses it to visualize the data in a bar chart and adds the chart to a dashboard.

HPE Ezmeral Unified Analytics Software includes the following components and applications to support an end-to-end workflow for this scenario:

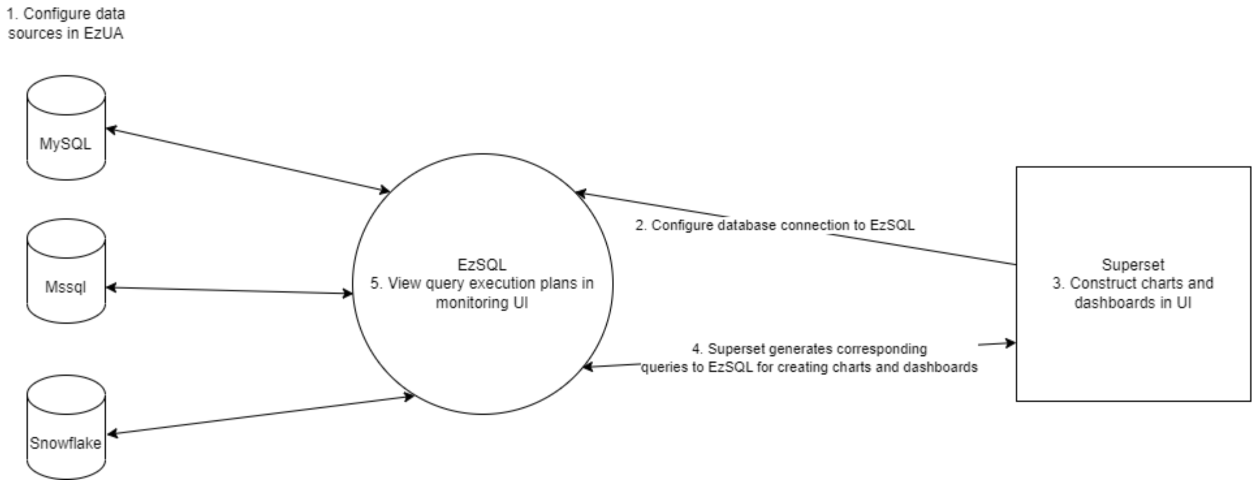
EzPresto

An MPP SQL query engine that runs accelerated queries against connected data sources and returns results to Superset for visualization. EzPresto connects to Superset through a database connection, enabling direct access to the data sources connected to Unified Analytics from Superset.

Superset

An analytical dashboarding application that communicates with EzPresto to send queries and receive the query results needed to visualize data from the selected data sets.

The following diagram shows the components and applications in the workflow:


1. Configure data sources in EzUA
MySQL
Mssql
Snowflake
EzSQL
5. View query execution plans in monitoring UI
2. Configure database connection to EzSQL
Superset
3. Construct charts and dashboards in UI
4. Superset generates corresponding queries to EzSQL for creating charts and dashboards

Steps

Sign in to HPE Ezmeral Unified Analytics Software and perform the following steps:

- A - Connect Data Sources on page 44
- B – Select Data Sets and Create a View on page 47
- C - Connect to the Presto Database on page 48
- D - Add the View to Superset and Create a Chart on page 49
- E - Specify Query Conditions to Visualize Results in the Chart on page 50
- F – Create a Superset Dashboard and Add the Chart (Visualized Data) on page 52
- G – Monitor Queries on page 52

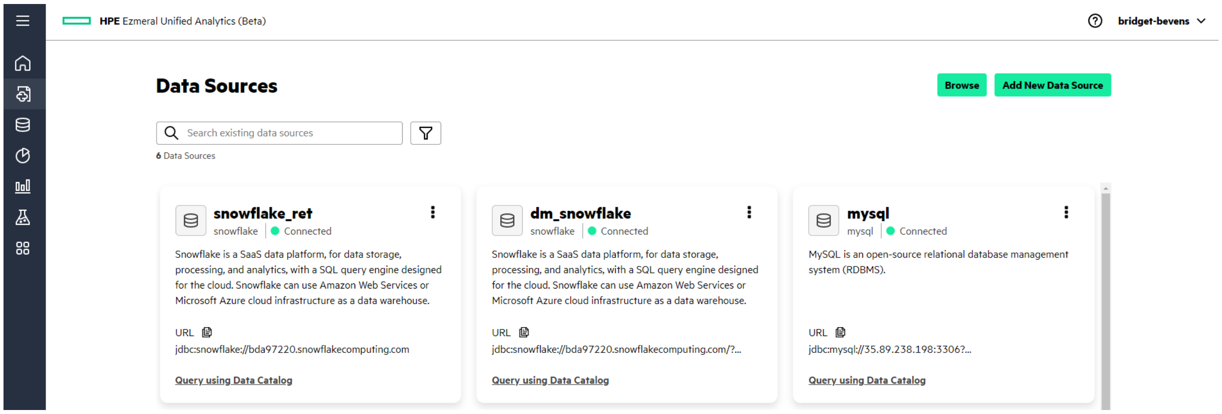
IMPORTANT: This tutorial demonstrates how to perform a series of tasks in HPE Ezmeral Unified Analytics Software to complete an example workflow. The data and information used in this tutorial are for example purposes only. You must connect Unified Analytics to your own data sources and use the data sets available to you in your data sources.

A - Connect Data Sources

Connect HPE Ezmeral Unified Analytics Software to external data sources that contain the data sets (tables and views) you want to work with. This tutorial uses MySQL, SQL Server, and Hive as the connected data source examples.

To connect a data source:

1. In the left navigation column, select **Data Engineering > Data Sources**. The **Data Sources** screen appears.



HPE Ezmeral Unified Analytics (Beta) bridget-bevens

Data Sources

Browse | Add New Data Source

Search existing data sources

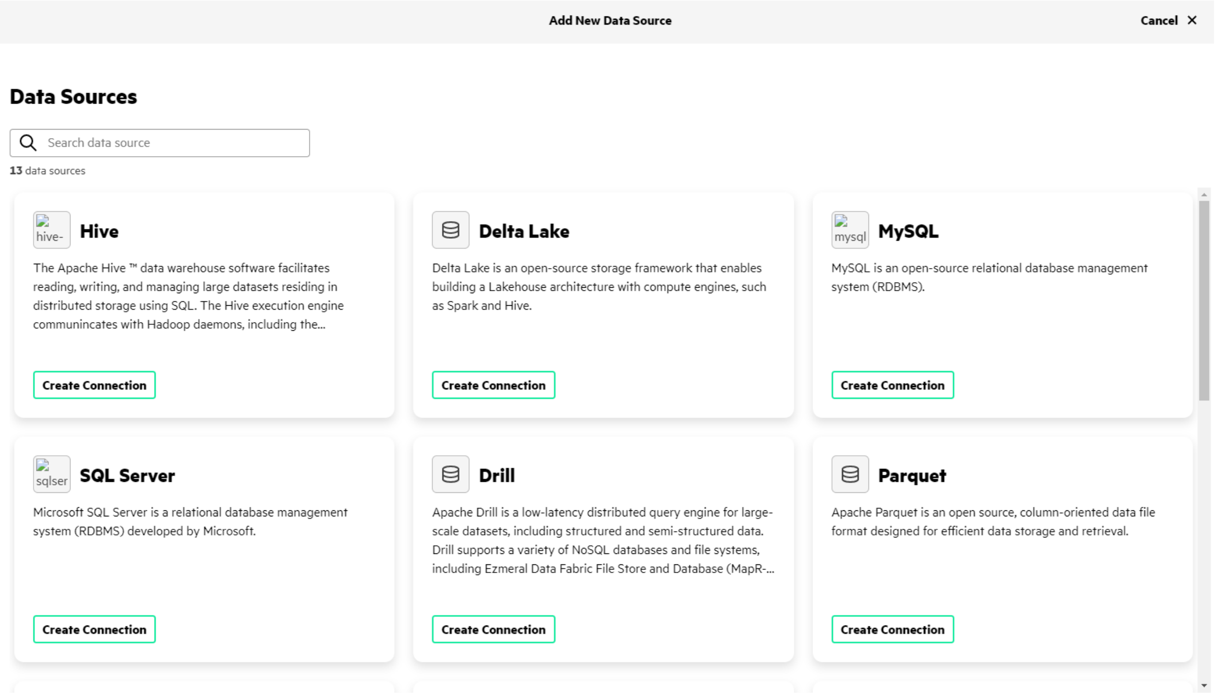
6 Data Sources

snowflake_ret
snowflake | Connected
Snowflake is a SaaS data platform, for data storage, processing, and analytics, with a SQL query engine designed for the cloud. Snowflake can use Amazon Web Services or Microsoft Azure cloud infrastructure as a data warehouse.
URL
jdbc:snowflake://bda97220.snowflakecomputing.com
Query using Data Catalog

dm_snowflake
snowflake | Connected
Snowflake is a SaaS data platform, for data storage, processing, and analytics, with a SQL query engine designed for the cloud. Snowflake can use Amazon Web Services or Microsoft Azure cloud infrastructure as a data warehouse.
URL
jdbc:snowflake://bda97220.snowflakecomputing.com/?...
Query using Data Catalog

mysql
mysql | Connected
MySQL is an open-source relational database management system (RDBMS).
URL
jdbc:mysql://35.89.238.198:3306?...
Query using Data Catalog

2. Click **Add New Data Source**.



Add New Data Source | Cancel ✕

Data Sources

Search data source

13 data sources

Hive
The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. The Hive execution engine communincates with Hadoop daemons, including the...
Create Connection

Delta Lake
Delta Lake is an open-source storage framework that enables building a Lakehouse architecture with compute engines, such as Spark and Hive.
Create Connection

MySQL
MySQL is an open-source relational database management system (RDBMS).
Create Connection

SQL Server
Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft.
Create Connection

Drill
Apache Drill is a low-latency distributed query engine for large-scale datasets, including structured and semi-structured data. Drill supports a variety of NoSQL databases and file systems, including Ezmeral Data Fabric File Store and Database (MapR-...
Create Connection

Parquet
Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval.
Create Connection

3. Complete the steps required to connect to the MySQL, SQL Server, and Hive data sources:

Connecting to MySQL

a. In the **Add New Data Source** screen, click **Create Connection** in the **MySQL** tile.

b. In the drawer that opens, enter the following information in the respective fields:

- **Name**: mysql
- **Connection URL**: jdbc:mysql://<ip-address>:<port>
- **Connection User**: myaccount
- **Connection Password**: moi123

- **Enable Local Snapshot Table**: Select the check box

  TIP: When **Enable Local Snapshot Table** is selected, the system caches remote table data to accelerate queries on the tables. The cache is active for the duration of the configured TTL or until the remote tables in the data source are altered.

- Click **Connect**. Upon successful connection, the system returns the following message:

  ```
  Successfully added data source
  "mysql".
  ```

Connecting to SQL Server

a. In the **Add New Data Source** screen, click **Create Connection** in the **SQL Server** tile.

b. In the drawer that opens, enter the following information in the respective fields:

   - **Name**: mssqlret2
   - **Connection URL**: jdbc:sqlserver:<ip-address>:<port>;database=retailstore
   - **Connection User**: myaccount
   - **Connection Password**: moi123
   - **Enable Local Snapshot Table**: Select the check box

     TIP: When **Enable Local Snapshot Table** is selected, the system caches remote table data to accelerate queries on the tables. The cache is active for the duration of the configured TTL or until the remote tables in the data source are altered.

   - Click **Connect**. Upon successful connection, the system returns the following message:

     ```
     Successfully added data source
     "mssqlret2".
     ```

Connecting to Hive

a. In the **Add New Data Source** screen, click **Create Connection** in the **Hive** tile.

b. In the drawer that opens, enter the following information in the respective fields:

   - **Name**: hiveview
   - **Hive Metastore**: file

- **Hive Metastore Catalog Dir**: file:///data/shared/tmpmetastore
- In **Optional Fields**, search for the following fields and add the specified values:
  - **Hive Max Partitions Per Writers**: 10000
  - **Hive Temporary Staging Directory Enabled**: Unselect
  - **Hive Allow Drop Table**: Select
- **Enable Local Snapshot Table**: Select the check box

  TIP: When **Enable Local Snapshot Table** is selected, the system caches remote table data to accelerate queries on the tables. The cache is active for the duration of the configured TTL or until the remote tables in the data source are altered.

- Click **Connect**. Upon successful connection, the system returns the following message:

  ```
  Successfully added data source
  "hiveview".
  ```

B – Select Data Sets and Create a View

In HPE Ezmeral Unified Analytics Software, complete the following steps to create a view. First select data sources and data sets to work with. Then, run a federated query against the selected data sets and create a view from the query. This tutorial creates an example view named *qf_retailstore_view*.

1. Select datasets.
   a. In the left navigation bar, select **Data Engineering > Data Catalog**.
   b. On the **Data Catalog** page, click the dropdown next to the **mysql** and **mssqlret2** data sources to expose the available schemas in those data sources.
   c. Select schemas for each of the data sources:
      - For the **mysql** data source, select the **retailstore** schema.
      - For the **mssqlret2** data source, select the **dbo** schema.
   d. In the **All Datasets** section, click the **filter icon** to open the **Filters** drawer.
   e. Use the filter to identify and select the following data sets in the selected schemas:
      - For the **dbo** schema, filter for and select the following datasets:
        - call_center
        - catalog_sales
        - data_dim
        - item

- For the **retailstore** schema, filter for and select the following datasets:
  - customer
  - customer_address
  - customer_demographics

f. After you select all the data sets, click **Apply**.

g. Click **Selected Datasets** (button that is displaying the number of selected data sets).

h. In the drawer that opens, click **Query Editor**. Depending on the number of selected data sets, you may have to scroll down to the bottom of the drawer to see the **Query Editor** button.

2. Query the datasets and create a view.

   a. In the **Query Editor**, click **+** to **Add Worksheet**.

   b. Run the following command to create a new schema, such as `hiveview.demoschema`, for example:

```
create schema if not exists hiveview.demoschema;
```

   c. Run a query to create a new view from a federated query against the selected data sets, for example:

```
create view hiveview.demoschema.qf_retailstore_view as select * from
mssqlret2.dbo.catalog_sales cs
inner join mssqlret2.dbo.call_center cc on cs.cs_call_center_sk =
cc.cc_call_center_sk
inner join mssqlret2.dbo.date_dim d on cs.cs_sold_date_sk =
d.d_date_sk
inner join mssqlret2.dbo.item i on cs.cs_item_sk = i.i_item_sk
inner join mysql.retailstore.customer c on cs.cs_bill_customer_sk =
c.c_customer_sk
inner join mysql.retailstore.customer_address ca on
c.c_current_addr_sk = ca.ca_address_sk
inner join mysql.retailstore.customer_demographics cd on
c.c_current_cdemo_sk = cd.cd_demo_sk
```

   d. Click **Run**. When the query completes, the status, **Finished**, displays.

C - Connect to the Presto Database

Complete the following steps to connect Superset to the Presto database for access to your data sources and data sets in HPE Ezmeral Unified Analytics Software. Once connected to the Presto database, you can access the view you created in the previous step (step B). To connect to the Presto database, you need the connection URI. You can get the URI from your HPE Ezmeral Unified Analytics Software administrator.

1. To open Superset, in the left navigation pane of HPE Ezmeral Unified Analytics Software, select **BI Reporting > Dashboards**. Superset opens in a new tab.

2. In Superset, select **Settings > Database Connections**.

3. Click **+DATABASE**.

4. In the **Connect a database** window, select the **Presto** tile.

5. Enter the SQLALCHEMY URI provided by your administrator.

6. Test the connection.

7. If the test was successful, click **Connect**.

D - Add the View to Superset and Create a Chart

Complete the following steps to import the view you created in HPE Ezmeral Unified Analytics Software and create a bar chart. This tutorial demonstrates how to import the view *qf_retailstore_view*.

1. In the left navigation bar, select **BI Reporting > Dashboards** to open Superset.

2. In Superset, click the **Datasets** tab.

3. Click **+DATASET**.

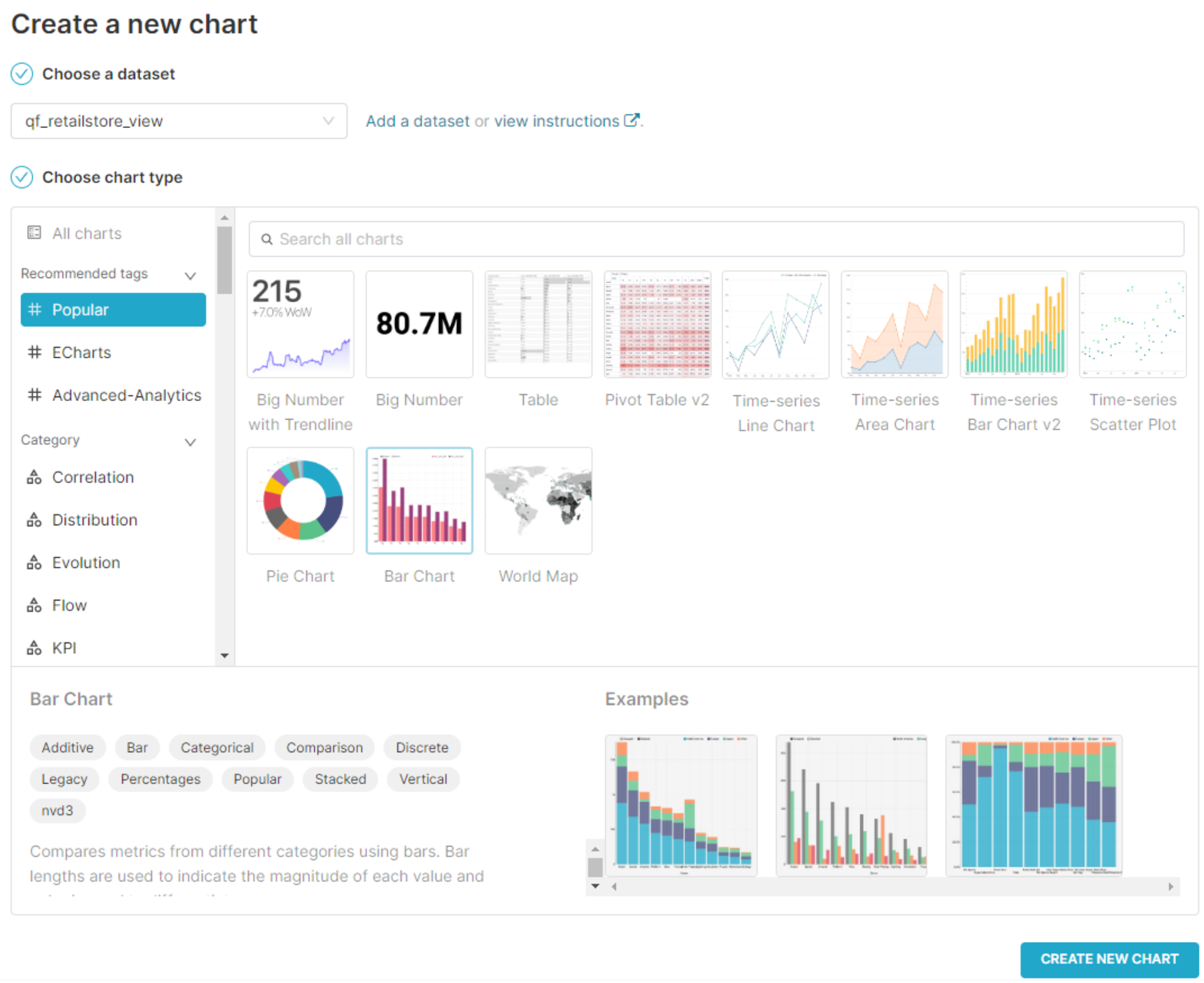
4. In the **Add Dataset** window, select the following options:

   - **DATABASE:** Presto
   - **SCHEMA:** <your_schema>
   - **SEE TABLE SCHEMA:** <your_view>

   This tutorial uses the *retailstore* schema and *qf_retailstore_view*.

5. Click **ADD DATASET AND CREATE CHART**.

6. In the **Create a New Chart** window, select **Bar Chart**.



Create a new chart

Choose a dataset

qf_retailstore_view

Add a dataset or view instructions.

Choose chart type

Compares metrics from different categories using bars. Bar lengths are used to indicate the magnitude of each value and

7. Click **CREATE NEW CHART**.

8. Enter a name for the chart, such as **Retail Store View**.

E - Specify Query Conditions to Visualize Results in the Chart

In Superset, charts visualize data based on the query conditions that you specify. The charts created in Superset automatically generate queries that Superset passes to the SQL query engine. Superset visualizes the query results in the chart. Try applying query conditions to visualize your data. Save your chart when done.

The following steps demonstrate how query conditions were applied to visualize data in the resulting example bar chart (shown in step 2):

1. Enter the specified query parameters in the following fields:

METRICS

a. Click into the **METRICS** field (located on the **DATA** tab). A metrics window opens.

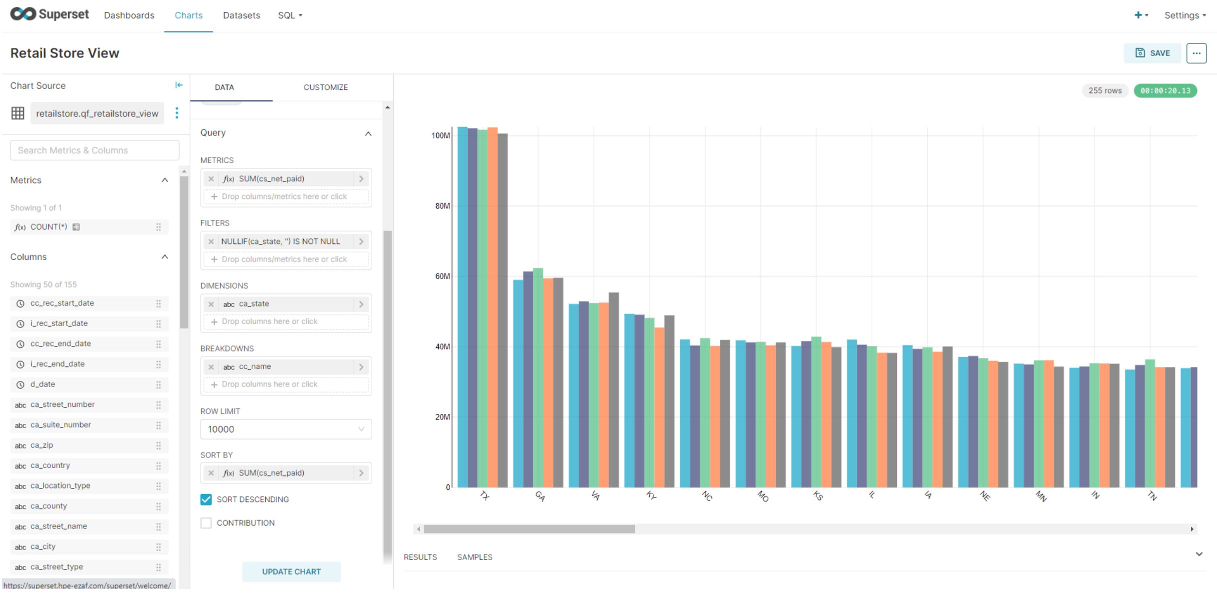
b. Select the **Simple** tab.

c. Click the **edit** icon and enter a name for the metric, such as **SUM(cs_net_paid)**.

d. In the **Column** field, select **cs_net_paid**.

	e. In the **Aggregate** field, select **SUM**. **f.** Click **Save**.
FILTERS	**a.** Click into the **FILTERS** field (located on the DATA tab). **b.** In the window that opens, select the **CUSTOM SQL** tab. **c.** Select the **WHERE** filter and enter the following: `NULLIF(ca_state, '') IS NOT NULL` **d.** Click **Save**.
DIMENSIONS	**a.** Drag and drop the **ca_state** column into the **DIMENSIONS** field. **b.** Click into the **BREAKDOWNS** column. **c.** In the window that opens, select the **SIMPLE** tab and select the **cc_name** column. **d.** Click **Save**.
SORT BY	**a.** Click into the **SORT BY** field. **b.** In the window that opens, select the **SIMPLE** tab and enter **cs_net_paid** as the COLUMN and **SUM** as the AGGREGATE. **c.** Click **Save**.

2. Click **CREATE CHART**. The bar chart displays results when the query finishes processing.



Retail Store View

retailstore.qf_retailstore_view

METRICS: SUM(cs_net_paid)

FILTERS: NULLIF(ca_state, '') IS NOT NULL

DIMENSIONS: ca_state

BREAKDOWNS: cc_name

ROW LIMIT: 10000

SORT BY: SUM(cs_net_paid)

☑ SORT DESCENDING

☐ CONTRIBUTION

https://superset.hpe-ezaf.com/superset/welcome/

3. Click **Save** to save the chart. In the **Save Chart** window that opens, do not enter or select a dashboard.

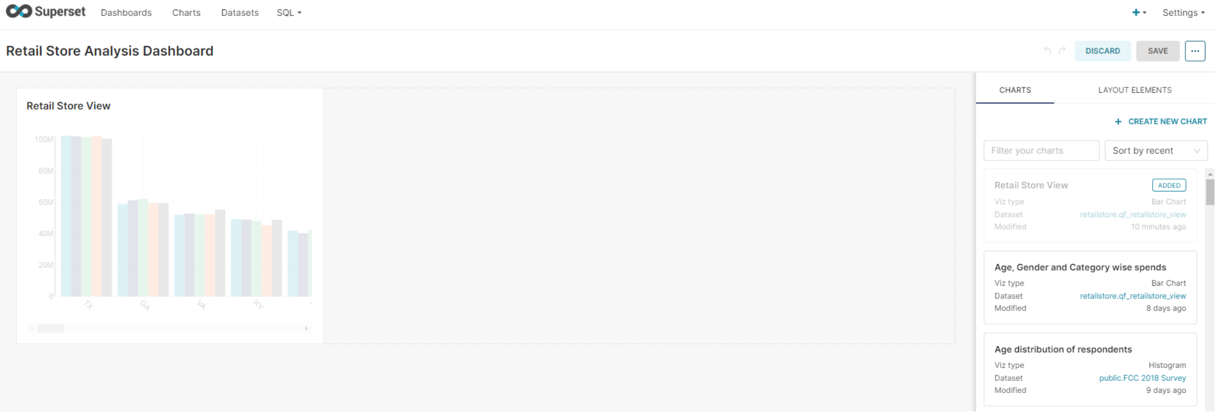
4. Click **Save** to continue.

F – Create a Superset Dashboard and Add the Chart (Visualized Data)

Complete the following steps to create a new dashboard and add your chart to the dashboard. This tutorial adds the *Retail Store View* chart to a dashboard named *Retail Store Analysis Dashboard*.

To create a new dashboard and add your visualized data:

1. In Superset, click on the **Dashboards** tab.
2. Click **+ DASHBOARD**.
3. Enter a name for the dashboard, for example **Retail Store Analysis Dashboard**.
4. Drag and drop your chart into the dashboard.



Retail Store View
Viz type Bar Chart
Dataset retailstore.qf_retailstore_view
Modified 10 minutes ago

Age, Gender and Category wise spends
Viz type Bar Chart
Dataset retailstore.qf_retailstore_view
Modified 8 days ago

Age distribution of respondents
Viz type Histogram
Dataset public.FCC 2018 Survey
Modified 9 days ago

5. Click **Save** to save the dashboard.

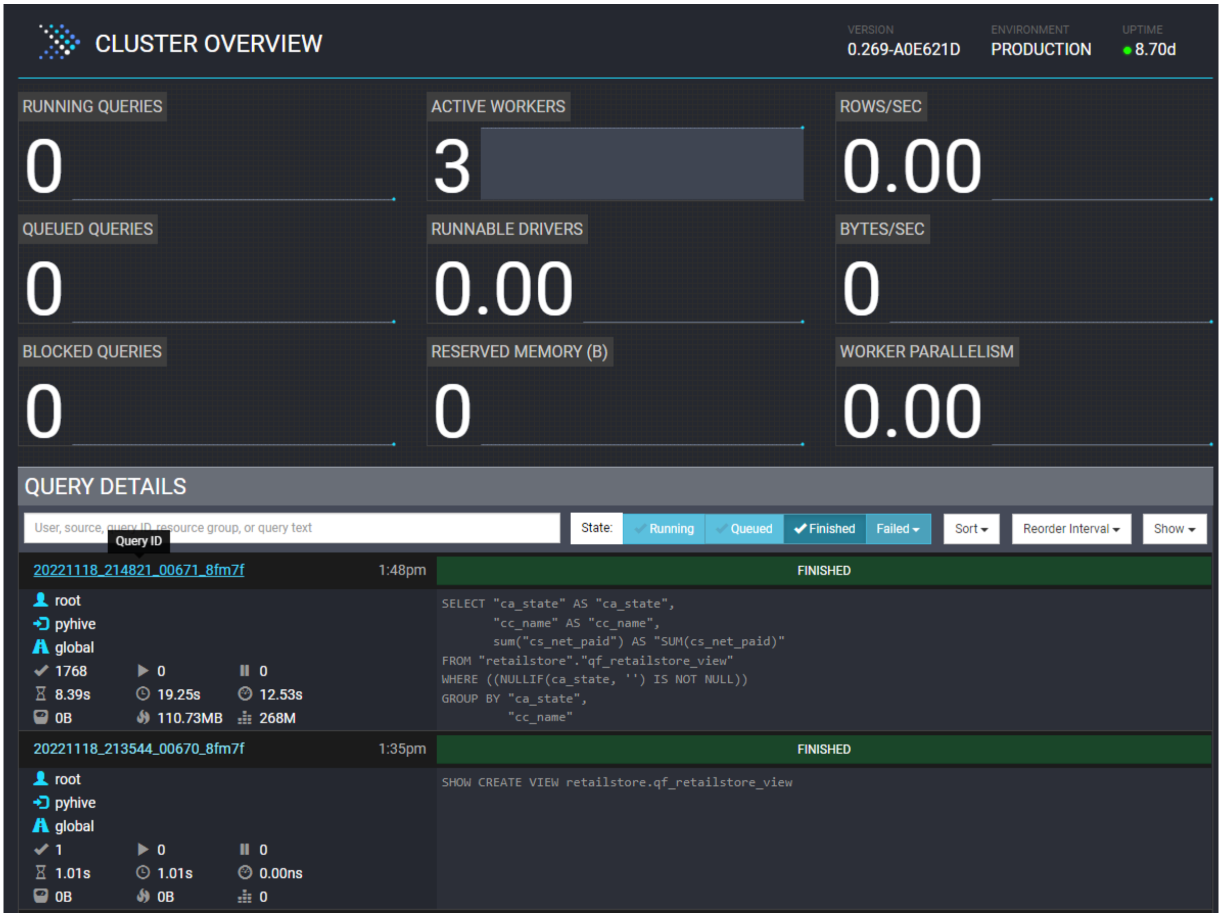
NOTE: Any time you open a chart or dashboard, Superset and the SQL query engine work together to visualize data. Loading a dashboard page triggers the queries against the database. As the queries run, buffering icons display until the data loads. When data is loaded, the visualizations display.

G – Monitor Queries

You can monitor queries generated through Superset from the EzPresto endpoint. You can access the EzPresto endpoint in the EzPresto tile in the **Applications & Frameworks** space in HPE Ezmeral Unified Analytics Software.

Complete the following steps to monitor the query that the chart generates:

1. Return to the HPE Ezmeral Unified Analytics Software UI.
2. In the left navigation bar, select **Applications & Frameworks**.
3. On the **Data Engineering** tab, click the EzPresto endpoint in the EzPresto tile. The EzPresto UI opens in a new tab.
4. In the **Query Details** section, verify that **Finished** is selected. Selected options have a visible checkmark.



VERSION 0.269-A0E621D
ENVIRONMENT PRODUCTION
UPTIME ● 8.70d

RUNNING QUERIES 0
ACTIVE WORKERS 3
ROWS/SEC 0.00
QUEUED QUERIES 0
RUNNABLE DRIVERS 0.00
BYTES/SEC 0
BLOCKED QUERIES 0
RESERVED MEMORY (B) 0
WORKER PARALLELISM 0.00

20221118_214821_00671_8fm7f 1:48pm FINISHED

root
pyhive
global
✓ 1768 ▶ 0 ⏸ 0
8.39s 19.25s 12.53s
0B 110.73MB 268M

```
SELECT "ca_state" AS "ca_state",
       "cc_name" AS "cc_name",
       sum("cs_net_paid") AS "SUM(cs_net_paid)"
FROM "retailstore"."qf_retailstore_view"
WHERE ((NULLIF(ca_state, '') IS NOT NULL))
GROUP BY "ca_state",
         "cc_name"
```

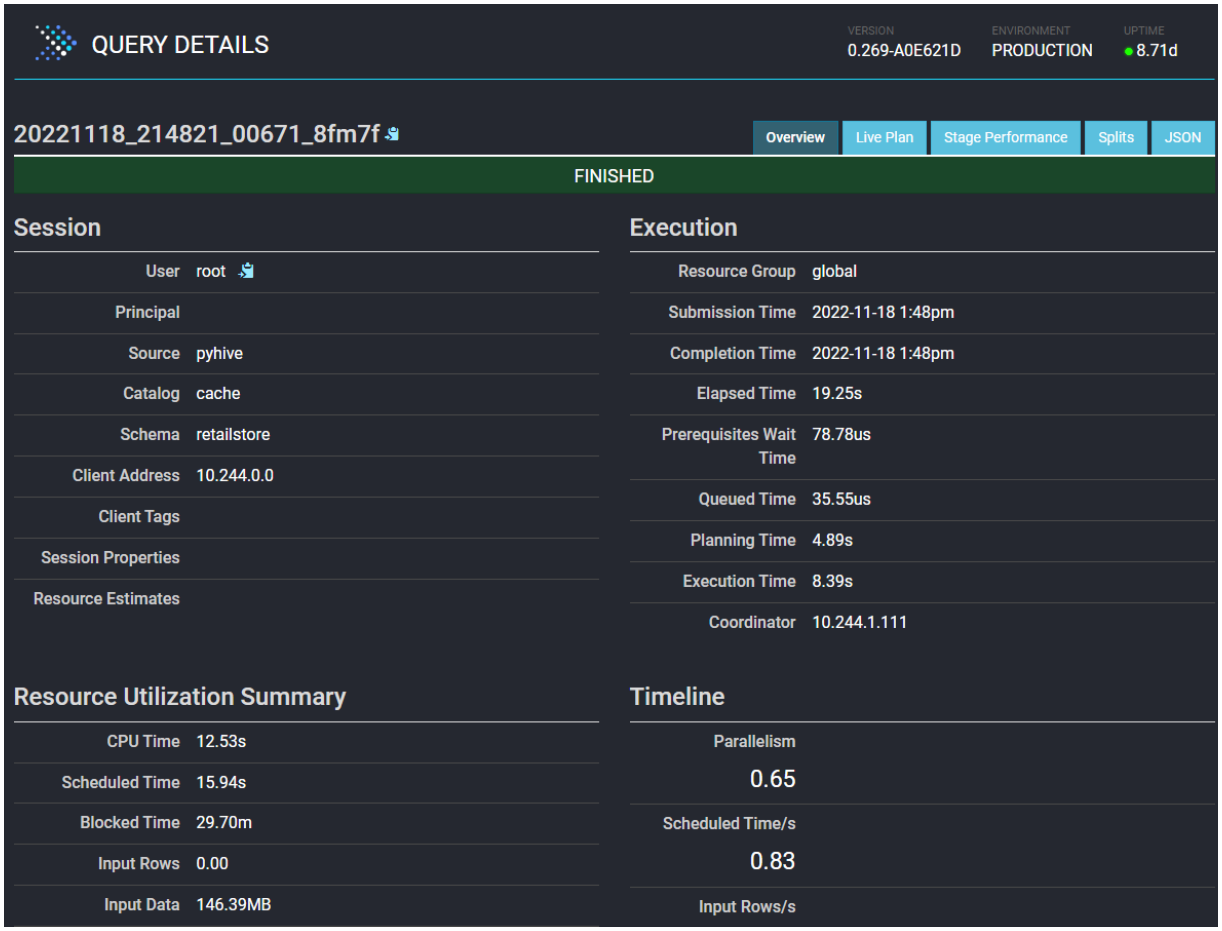
20221118_213544_00670_8fm7f 1:35pm FINISHED

root
pyhive
global
✓ 1 ▶ 0 ⏸ 0
1.01s 1.01s 0.00ns
0B 0B 0

```
SHOW CREATE VIEW retailstore.qf_retailstore_view
```

You can see the query that ran to populate the *Retail Store View* bar chart in the *Retail Store Analysis Dashboard*.

5. Click on the **Query ID** to see the query details.



QUERY DETAILS

VERSION 0.269-A0E621D ENVIRONMENT PRODUCTION UPTIME ● 8.71d

20221118_214821_00671_8fm7f

FINISHED

Session

User	root
Principal	
Source	pyhive
Catalog	cache
Schema	retailstore
Client Address	10.244.0.0
Client Tags	
Session Properties	
Resource Estimates	

Execution

Resource Group	global
Submission Time	2022-11-18 1:48pm
Completion Time	2022-11-18 1:48pm
Elapsed Time	19.25s
Prerequisites Wait Time	78.78us
Queued Time	35.55us
Planning Time	4.89s
Execution Time	8.39s
Coordinator	10.244.1.111

Resource Utilization Summary

CPU Time	12.53s
Scheduled Time	15.94s
Blocked Time	29.70m
Input Rows	0.00
Input Data	146.39MB

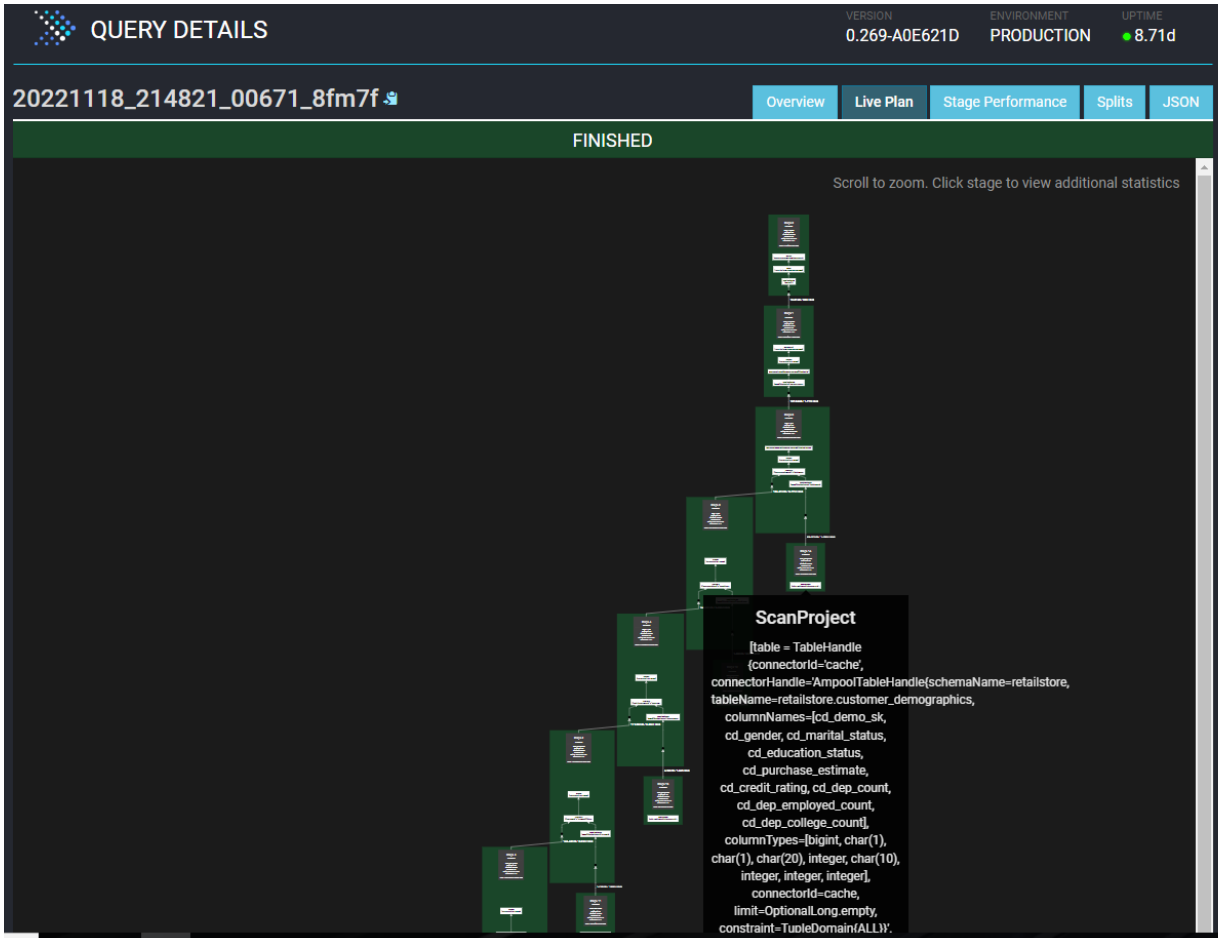
Timeline

Parallelism
0.65

Scheduled Time/s
0.83

Input Rows/s

6. To see a visualized query plan and metadata for the query, click **Live Plan** and hover over different areas of the visualized plan. You can also click on various parts of the visualized plan to zoom in on details.



ScanProject

[table = TableHandle {connectorId='cache', connectorHandle='AmpoolTableHandle{schemaName=retailstore, tableName=retailstore.customer_demographics, columnNames=[cd_demo_sk, cd_gender, cd_marital_status, cd_education_status, cd_purchase_estimate, cd_credit_rating, cd_dep_count, cd_dep_employed_count, cd_dep_college_count], columnTypes=[bigint, char(1), char(1), char(20), integer, char(10), integer, integer, integer], connectorId=cache, limit=OptionalLong.empty, constraint=TupleDomain{ALL}}',

End of Tutorial

You have completed this tutorial. This tutorial demonstrated the integration of the HPE Ezmeral Unified Analytics Software SQL query engine (EzPresto on page 294) with Superset to visualize the results of a query on data sets made available through the default Presto database connection. This tutorial also showed you how to monitor queries from the EzPresto on page 294 Cluster Monitoring tool.

Running Independent Tune Trials (Ray Tune)

Provides an end-to-end workflow for running independent Tune trials in HPE Ezmeral Unified Analytics Software.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software.
- Verify that the installed Ray client and server versions match. To verify, complete the following steps in the terminal:

  1. To switch to Ray's environment, run:

     ```
     source /opt/conda/etc/profile.d/conda.sh && conda activate ray
     ```

  2. To verify that the Ray client and server versions match, run :

     ```
     ray --version
     ```

About this task

In this tutorial, you will run `N` independent model training trials using Tune as a simple grid sweep.

You will complete the following steps:

Procedure

1. Create a notebook server using the `jupyter-data-science` image with at least 3 CPUs and 4 Gi of memory in Kubeflow. See Creating and Managing Notebook Servers on page 436.

2. In your notebook environment, activate the Ray-specific Python kernel.

3. To ensure optimal performance, use dedicated directories containing only the essential files needed for that job submission as a working directory.

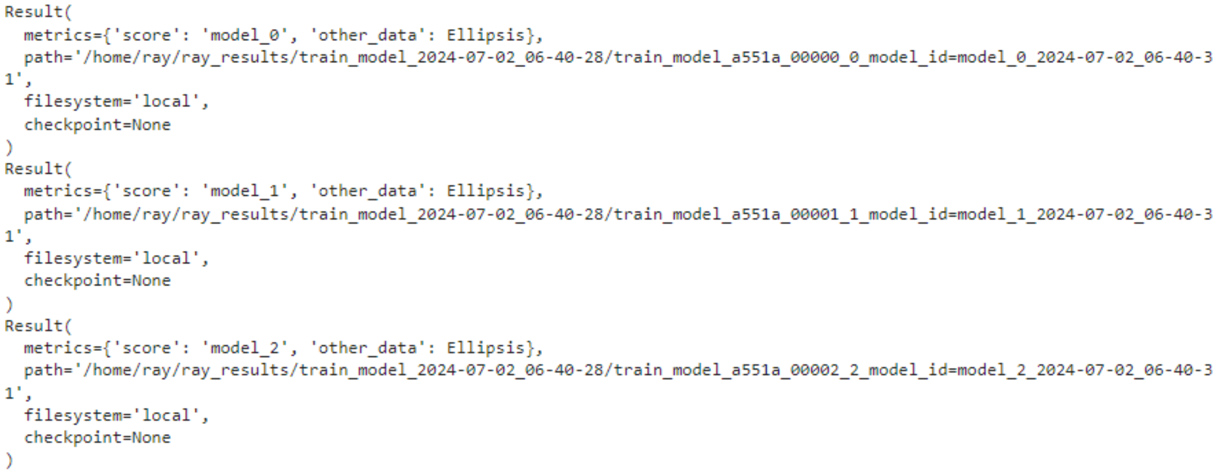
   For example, if you do not see the `Ray-Tune` folder in the `<username>` directory, copy the folder from the `shared/ezua-tutorials/current-release/Data-Science/Ray/Ray-Tune` directory into the `<username>` directory. The `shared` directory is accessible to all users. Editing or running examples from the `shared` directory is not advised. The `<username>` directory is specific to you and cannot be accessed by other users.

4. Open the `independent-tune-trials-executor.ipynb` file in the `<username>/Ray-Tune` directory.

5. Select the first cell of the `independent-tune-trials-executor.ipynb` notebook and click **Run the selected cells and advance (play icon)**. Continue until you run all cells.

Results

After successful completion, you can view the trial metadata as follows:



```
Result(
  metrics={'score': 'model_0', 'other_data': Ellipsis},
  path='/home/ray/ray_results/train_model_2024-07-02_06-40-28/train_model_a551a_00000_0_model_id=model_0_2024-07-02_06-40-3
1',
  filesystem='local',
  checkpoint=None
)
Result(
  metrics={'score': 'model_1', 'other_data': Ellipsis},
  path='/home/ray/ray_results/train_model_2024-07-02_06-40-28/train_model_a551a_00001_1_model_id=model_1_2024-07-02_06-40-3
1',
  filesystem='local',
  checkpoint=None
)
Result(
  metrics={'score': 'model_2', 'other_data': Ellipsis},
  path='/home/ray/ray_results/train_model_2024-07-02_06-40-28/train_model_a551a_00002_2_model_id=model_2_2024-07-02_06-40-3
1',
  filesystem='local',
  checkpoint=None
)
```

To learn about this tutorial in detail, see Ray Tune Example from open-source Ray documentation.

Running Ray GPU Example

Describes how to run the Ray GPU example in HPE Ezmeral Unified Analytics Software.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software.

- Verify that the installed Ray client and server versions match. To verify, complete the following steps in the terminal:
  1. To switch to Ray's environment, run:

     ```
     source /opt/conda/etc/profile.d/conda.sh && conda activate ray
     ```

  2. To verify that the Ray client and server versions match, run :

     ```
     ray --version
     ```

- Verify that the GPU support is enabled in your Ray cluster. See Enabling GPU Support During HPE Ezmeral Unified Analytics Software Installation on page 428 or Enabling GPU Support and Configuring Resources After HPE Ezmeral Unified Analytics Software Installation on page 428.

About this task

In this tutorial, you will run the sample Ray GPU example and analyze logs to ensure that Unified Analytics is running the GPU-accelerated jobs.

You will complete the following steps:

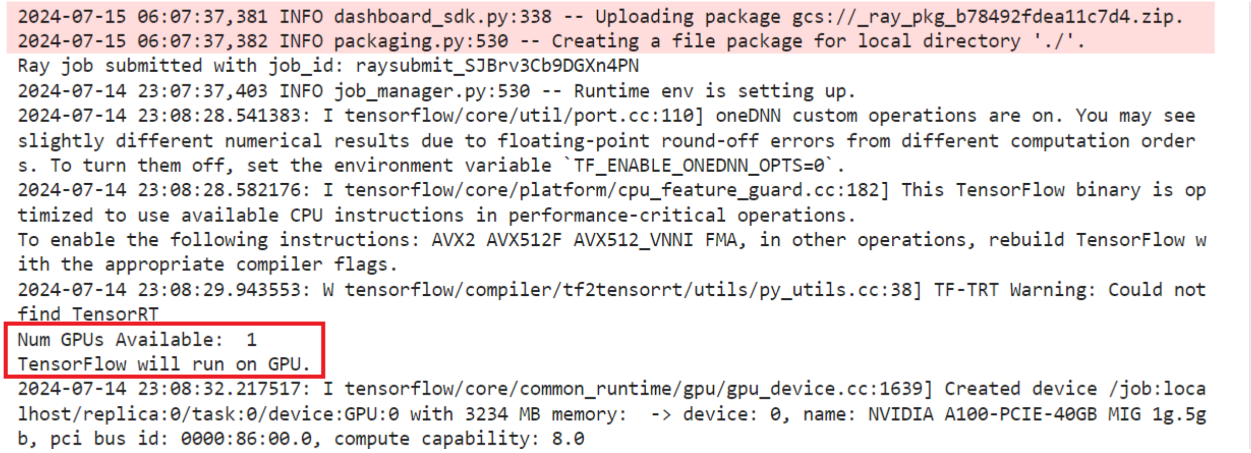
Procedure

1. Create a notebook server using the `jupyter-tensorflow-cuda-full` image with at least 3 CPUs and 4 Gi of memory in Kubeflow. See Creating GPU-Enabled Notebook Servers on page 438.
2. In your notebook environment, activate the Ray-specific Python kernel.
3. To ensure optimal performance, use dedicated directories containing only the essential files needed for that job submission as a working directory.

   For example, if you do not see the `Ray-GPU` folder in the `<username>` directory, copy the folder from the `shared/ezua-tutorials/current-release/Data-Science/Ray/Ray-GPU` directory into the `<username>` directory. The `shared` directory is accessible to all users. Editing or running examples from the `shared` directory is not advised. The `<username>` directory is specific to you and cannot be accessed by other users.
4. Open the `ray-gpu-executor.ipynb` file in the `<username>/Ray-GPU` directory.
5. Select the first cell of the `ray-gpu-executor.ipynb` notebook and click **Run the selected cells and advance (play icon)**. Continue until you run all cells.

Results

After successful completion, you can view that Unified Analytics is running the GPU-accelerated Ray job.



```
2024-07-15 06:07:37,381 INFO dashboard_sdk.py:338 -- Uploading package gcs://_ray_pkg_b78492fdea11c7d4.zip.
2024-07-15 06:07:37,382 INFO packaging.py:530 -- Creating a file package for local directory './'.
Ray job submitted with job_id: raysubmit_SJBrv3Cb9DGXn4PN
2024-07-14 23:07:37,403 INFO job_manager.py:530 -- Runtime env is setting up.
2024-07-14 23:08:28.541383: I tensorflow/core/util/port.cc:110] oneDNN custom operations are on. You may see
slightly different numerical results due to floating-point round-off errors from different computation order
s. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-07-14 23:08:28.582176: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is op
timized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow w
ith the appropriate compiler flags.
2024-07-14 23:08:29.943553: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT
Num GPUs Available:  1
TensorFlow will run on GPU.
2024-07-14 23:08:32.217517: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1639] Created device /job:loca
lhost/replica:0/task:0/device:GPU:0 with 3234 MB memory:  -> device: 0, name: NVIDIA A100-PCIE-40GB MIG 1g.5g
b, pci bus id: 0000:86:00.0, compute capability: 8.0
```

Running Ray Matrix Multiplication Application

Provides an end-to-end example for creating a notebook server and submitting a matrix multiplication application job in local and distributed setting using Ray in HPE Ezmeral Unified Analytics Software.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software.
- Verify that the installed Ray client and server versions match. To verify, complete the following steps in the terminal:
  1. To switch to Ray's environment, run:

     ```
     source /opt/conda/etc/profile.d/conda.sh && conda activate ray
     ```

  2. To verify that the Ray client and server versions match, run :

     ```
     ray --version
     ```

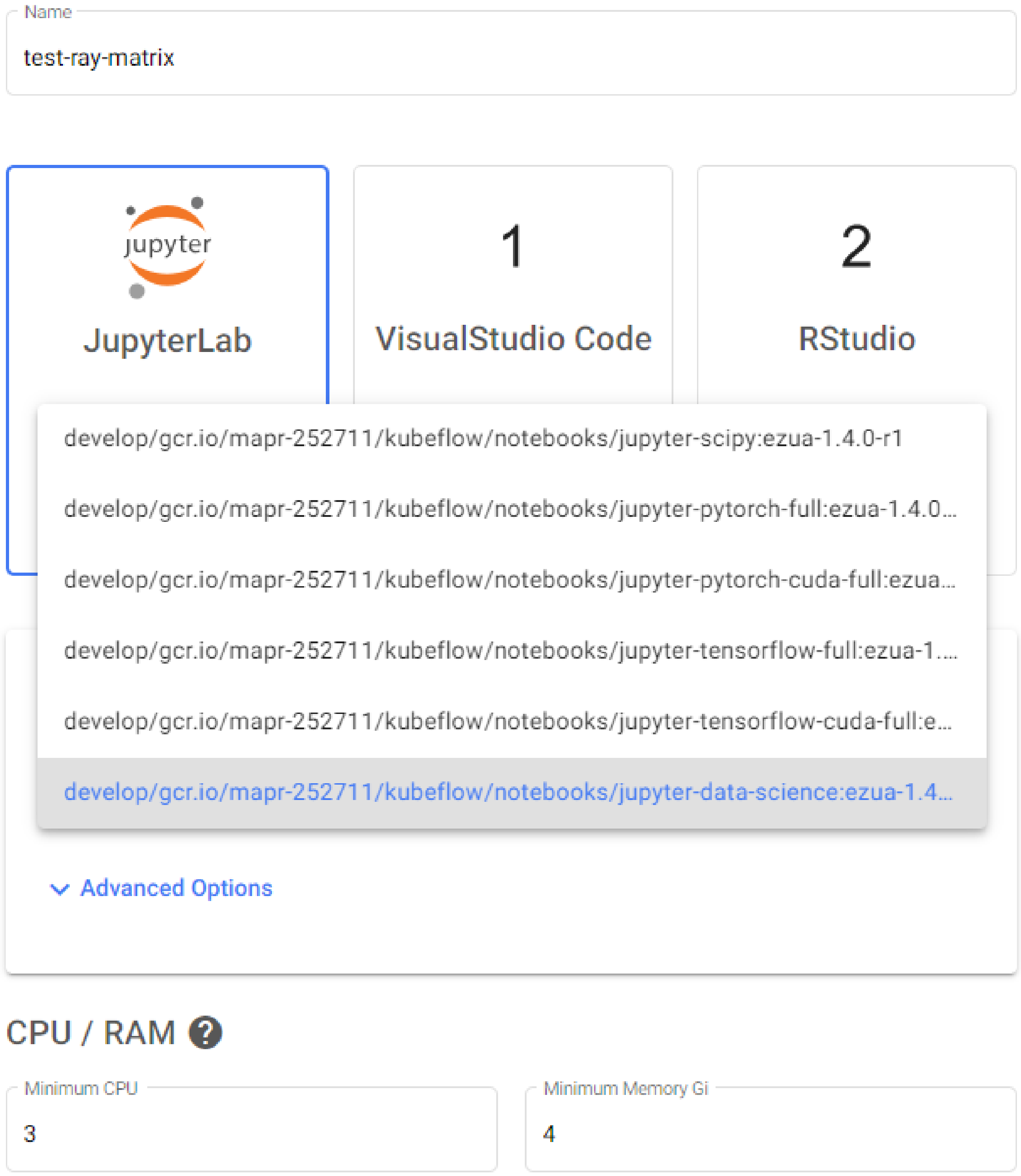
About this task

In this tutorial, you will:

1. Submit the regular Python functions as the Ray tasks using `JobSubmissionClient` to utilize Ray's distributed computing capabilities.
2. Generate two random matrices and multiply the generated matrices locally and using Ray utilizing the NumPy package.
3. Record the duration for matrix generation and multiplication to observe Ray’s efficiency under heavy workloads.

Procedure

1. Create a notebook server using the `jupyter-data-science` image with at least 3 CPUs and 4 Gi of memory in Kubeflow. See Creating and Managing Notebook Servers on page 436.



Name: test-ray-matrix

develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezua-1.4.0-r1

develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-full:ezua-1.4.0...

develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-cuda-full:ezua...

develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezua-1....

develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:e...

develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-data-science:ezua-1.4...

CPU / RAM

Minimum CPU: 3

Minimum Memory Gi: 4

2. In your notebook environment, activate the Ray-specific Python kernel.

3. To ensure optimal performance, use dedicated directories containing only the essential files needed for that job submission as a working directory.

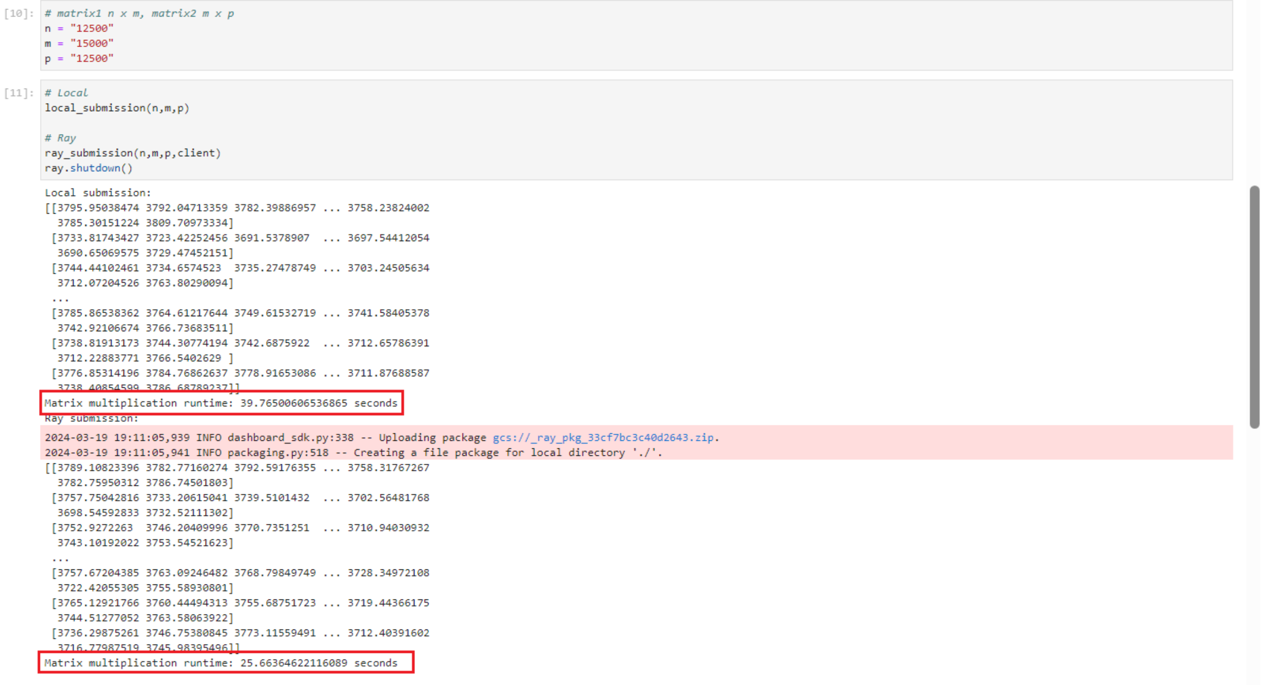
   For example, if you do not see the `Matrix_Multiplication` folder in the `<username>` directory, copy the folder from the `shared/ezua-tutorials/current-release/Data-Science/Ray/Ray-CPU` directory into the `<username>` directory. The `shared` directory is accessible to all users. Editing or running examples from the `shared` directory is not advised. The `<username>` directory is specific to you and cannot be accessed by other users.

4. Open the `ray-matrix_multiplication-executor.ipynb` file in the `<username>/Matrix_Multiplication` directory.

5. Select the first cell of the `ray-matrix_multiplication-executor.ipynb` notebook and click **Run the selected cells and advance (play icon)**. Continue until you run all cells.

Results

After running the final block of code, you will get the following output:



```
[10]: # matrix1 n x m, matrix2 m x p
      n = "12500"
      m = "15000"
      p = "12500"

[11]: # Local
      local_submission(n,m,p)

      # Ray
      ray_submission(n,m,p,client)
      ray.shutdown()

      Local submission:
      [[3795.95038474 3792.04713359 3782.39886957 ... 3758.23824002
        3785.30151224 3809.70973334]
       [3733.81743427 3723.42252456 3691.5378907  ... 3697.54412054
        3690.65069575 3729.47452151]
       [3744.44102461 3734.6574523  3735.27478749 ... 3703.24505634
        3712.07204526 3763.80290094]
       ...
       [3785.86538362 3764.61217644 3749.61532719 ... 3741.58405378
        3742.92106674 3766.73683511]
       [3738.81913173 3744.30774194 3742.6875922  ... 3712.65786391
        3712.22883771 3766.5402629 ]
       [3776.85314196 3784.76862637 3778.91653086 ... 3711.87688587
        3738.40854599 3786.68789237]]
      Matrix multiplication runtime: 39.76500606536865 seconds
      Ray submission:
      2024-03-19 19:11:05,939 INFO dashboard_sdk.py:338 -- Uploading package gcs://_ray_pkg_33cf7bc3c40d2643.zip.
      2024-03-19 19:11:05,941 INFO packaging.py:518 -- Creating a file package for local directory './'.
      [[3789.10823396 3782.77160274 3792.59176355 ... 3758.31767267
        3782.75950312 3786.74501803]
       [3757.75042816 3733.20615041 3739.5101432  ... 3702.56481768
        3698.54592833 3732.52111302]
       [3752.9272263  3746.20409996 3770.7351251  ... 3710.94030932
        3743.10192022 3753.54521623]
       ...
       [3757.67204385 3763.09246482 3768.79849749 ... 3728.34972108
        3722.42055305 3755.58930801]
       [3765.12921766 3760.44494313 3755.68751723 ... 3719.44366175
        3744.51277052 3763.58063922]
       [3736.29875261 3746.75380845 3773.11559491 ... 3712.40391602
        3716.77987519 3745.98395496]]
      Matrix multiplication runtime: 25.6636462211608 seconds
```

Matrix multiplication runtime for local submission is 39.76 seconds.

Matrix multiplication runtime for Ray submission is 25.66 seconds.

The performance of the Ray job submission is better than that of the local job submission.

Submitting a Spark Wordcount Application

Provides an end-to-end example for creating and submitting a wordcount Spark Application in HPE Ezmeral Unified Analytics Software.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software.
- Download the `wordcount.yaml` file from the wordcount folder.

About this task

The wordcount Spark application counts the number of occurrences of each unique word in the `wordcount.txt` input file.

1. In HPE Ezmeral Unified Analytics Software, use one the following methods to go to **Spark Applications**:
   - In the left navigation bar, click the **Analytics icon** and click **Spark Applications**.
   - In the left navigation bar, click the **Tools & Frameworks** icon. On the **Analytics** tab, click **Open** in the **Spark Operator** tile.

2. Click **Create Application** on the Spark Applications screen. Navigate through each step within the **Create Spark Application** wizard:

   a. **Application Details**: Choose **Upload YAML**.

      YAML File

      Click **Select File** to upload the downloaded `wordcount.yaml` file from your local system. The fields in the wizard are populated with the information from YAML.

      Name:

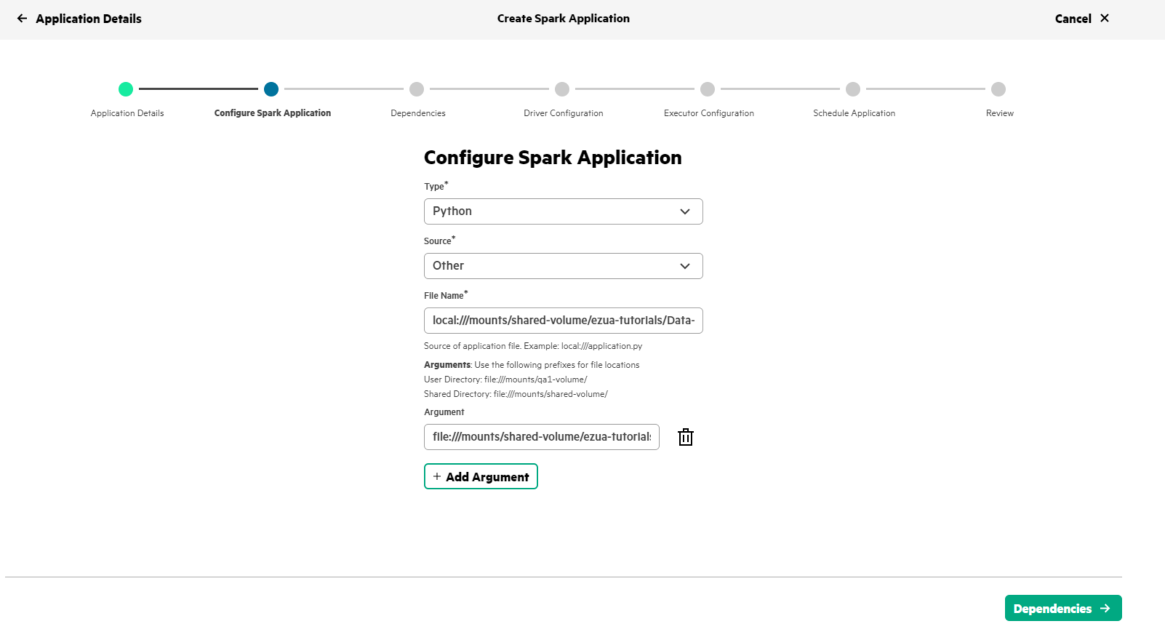
      Update the application name as `username-word-count`.

      NOTE: The application name must be unique.

      Description:

      Enter the application description. For example: This application counts words in a text file.

   b. **Configure Spark Application**: The fields in this wizard are populated with the information from YAML.



Type*: Python

Source*: Other

File Name*: local:///mounts/shared-volume/ezua-tutorials/Data-

Source of application file. Example: local:///application.py

Arguments: Use the following prefixes for file locations
User Directory: file:///mounts/qa1-volume/
Shared Directory: file:///mounts/shared-volume/

Argument: file:///mounts/shared-volume/ezua-tutorial:

   c. Click **Dependencies**. The wordcount application does not require any additional dependencies.

   d. Click **Driver Configuration**. When boxes in this wizard are left blank, default values are set. The default values are as follows:

      - Number of Cores: 1
      - Core Limit: unlimited
      - Memory: 1g

   e. Click **Executor Configuration**. When boxes in this wizard are left blank, default values are set. The default values are as follows:

      - Number of Executors: 1
      - Number of Cores per Executor: 1

- Core Limit per Executor: unlimited
- Memory per Executor: 1g

f. Click **Schedule Application**. If you want to schedule a Spark application, see Creating Spark Applications on page 379 for details.

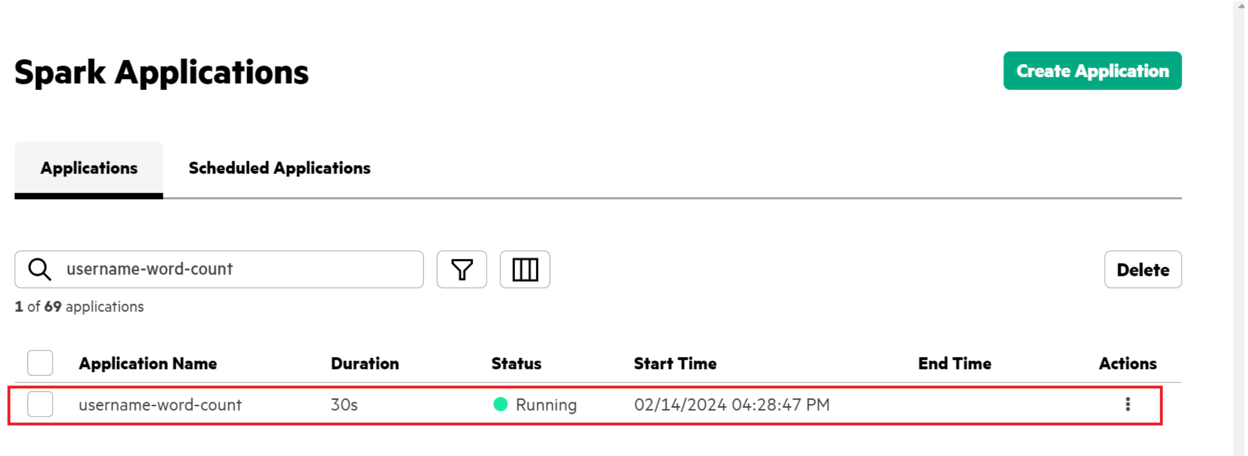
g. Click **Review.** To view the application configuration, click **Edit YAML**. To apply the changes, click **Save Changes**. To cancel the changes, click **Discard Changes**.

You can also click the **pencil** icon in each section to navigate to the specific step to change the application configuration.

3. Click **Create Spark Application** on the bottom right of the **Review** step.

Results

The wordcount Spark application is created and submitted. You can view it on the **Spark Applications** screen.



username-word-count

1 of 69 applications

□	Application Name	Duration	Status	Start Time	End Time	Actions
□	username-word-count	30s	● Running	02/14/2024 04:28:47 PM		⋮

You can also view the logs to check the output of the wordcount application. To see the logs, click the **menu** icon in the **Actions** column of the `username-word-count` application, and click **View Logs**.

Resources

Provides links to additional resources such as product licensing information, on-demand training, videos, blogs, and HPE Ezmeral Unified Analytics Software community.

In addition to the product documentation, you may be interested in the following resources:

Download Documentation

Click here to download a PDF of the HPE Ezmeral Unified Analytics Software documentation.

Contact for Support

Get in touch with HPE Ezmeral Unified Analytics Software support team through HPE Support Center.

HPE Ezmeral Software Resources

Slack Community for Developers	https://slack.hpedev.io/
Videos, Reports, and Case Studies	https://www.hpe.com/us/en/resource-library.html

HPE GreenLake Marketplace	https://www.hpe.com/us/en/software/marketplace.html/platform/ezmeraldata

Glossary

To find the list of terms (with description) used in HPE Ezmeral Unified Analytics Softwaredocumentation. See Glossary on page 453.

Third-Party Licenses

To download a spreadsheet listing the third-party components in HPE Ezmeral Unified Analytics Software in Microsoft Excel format (.xlsx), click .

Administration

Provides information about managing applications and clusters in HPE Ezmeral Unified Analytics Software.

Installation

Provides links to HPE Ezmeral Unified Analytics Software installation and service activation topics.

Installing on User-Provided Hosts (Connected and Air-gapped Environments)

Provides the steps for installing HPE Ezmeral Unified Analytics Software on user-provided hosts in connected and air-gapped environments. A user-provided host is a bare metal machine or virtual machine (VM) that meets the installation prerequisites.

HPE Ezmeral Unified Analytics Software supports bare metal and VM installations on AWS, GCP, and Azure. You can install HPE Ezmeral Unified Analytics Software from a laptop or host machine.

Complete the following steps to install HPE Ezmeral Unified Analytics Software on a bare metal machine or virtual machine (VM):

1. Review the prerequisites and verify that the requirements have been met.
2. Run the installation script to access the Installer Web UI.
3. In the Installer Web UI, provide the pertinent information on each of the following screens:
   a. Node Setup
   b. Installation Details
   c. User Authentication Details on page 69
   d. Tools & Frameworks on page 70
   e. Storage Details on page 70
   f. Review
4. Complete the post-installation steps.

The HPE Ezmeral Unified Analytics Software deployment runs on a Kubernetes cluster. Components within HPE Ezmeral Unified Analytics Software cannot launch until they download their respective container images. How the components download the container images depends on your environment.

The following table describes container downloads in different environments:

Environment	Description
Direct connection	If the machine is directly connected to the internet (UI accessible), you do not have to provide any proxy settings during installation. However, the firewall settings can prevent the packages from being downloaded.
Proxy connection	If the machine is connected to the internet via proxy, you must provide the proxy server information for http, https, and no_proxy during installation
Air-gapped environment	The Airgap Utility prerequisite describes the requirements for an air-gapped environment. See Installation Prerequisites on page 73.

Prerequisites

See Installation Prerequisites on page 73.

Run the Installation Script to Access the Installer Web UI

To run the installation script and open the Installer Web UI, complete the following steps:

1. Go to the directory where you extracted the installer bundle (`HPE_Ezmeral_Unified_Analytics_Installer_S1U85-70034`)

   ```
   cd S1U85-70034
   ```

2. Run the installation script on a host, but do not run it on the hosts used to deploy HPE Ezmeral Unified Analytics Software. See Installation Prerequisites on page 73 for details.

   ```
   ./start_ezua_installer_ui.sh
   ```

   The launcher guides you through the prompts to start the Installer Web UI.

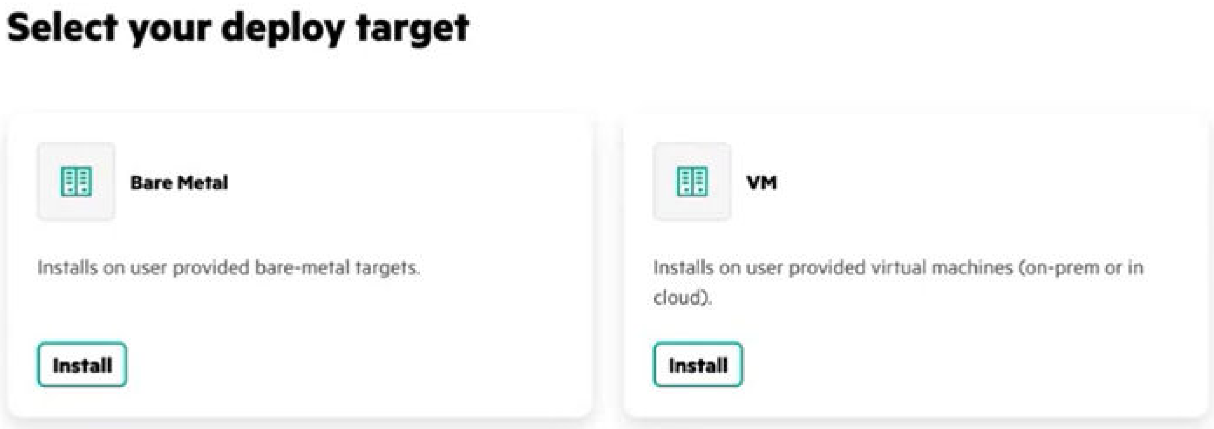
   NOTE:

   - If you get a permission denied error, run `chmod +x start_ezua*.sh` before you run the installation script.
   - If the image is locally available, the container starts right away. If the image is not local, it takes time to download the image. Time for the image to download and start the container UI depends on network speed.
   - If you ran the script on a laptop, you can access the installer UI by connecting to the browser using `localhost:8080`. If you ran the script on a different node, you can access the installer UI by connecting to the browser using `<node-ip-address>:8080`. Verify that port 8080 is opened through firewalls from the laptop to the node running the installer.
   - If proxy settings are present in the environment, include the control plane node DNS names of the workload and coordinator clusters in the NO_PROXY list.

3. On the screen that appears, select one of the options. The HPE Ezmeral Coordinator is the component that orchestrates the deployment of HPE Ezmeral Unified Analytics Software instances.

Installation Using New HPE Ezmeral Coordinator	• For first time installation, select this option. • When you install with a new HPE Ezmeral Coordinator, you designate the control plane (HPE Ezmeral Coordinator and management cluster nodes) and worker nodes, as described in the following section, Node Setup on page 65.
Installation Using Existing HPE Ezmeral Coordinator	• If you previously installed HPE Ezmeral Unified Analytics Software, select this option to use the existing HPE Ezmeral Coordinator to create a new HPE Ezmeral Unified Analytics Software cluster. • All files in the existing cluster are cleared, except for the `kubeconfig` file for the HPE Ezmeral Coordinator. You do not have to reconfigure the management cluster or upload the configuration file again.

4. On the **Select your deploy target** screen, select **Install** in the **Bare Metal** or **VM** tile.



Select your deploy target

Bare Metal

Installs on user provided bare-metal targets.

VM

Installs on user provided virtual machines (on-prem or in cloud).

The **Node Setup** screen appears.

Node Setup

Node setup sets up the control plane and worker nodes. You can upload a YAML file or manually configure the nodes through fields in the Installer Web UI. You can also run an installation pre-check script, as described in step 2 of this section.

The following table describes control plane and worker nodes:

Node Type	Description	Minimum Required		Minimum Required for High Availability
Control Plane	Enter a comma-separated list of nodes (IP addresses). If you chose to install using a new HPE Ezmeral Coordinator, the first node listed becomes the HPE Ezmeral Coordinator node. This node orchestrates the deployment of HPE Ezmeral Unified Analytics Software instances. The remainder of the nodes in the list serve as the management cluster.	Installation Using New HPE Ezmeral Coordinator (First-time installation)	2*	4**
		Installation Using Existing HPE Ezmeral Coordinator	1	3
Worker	Enter a comma-separated list of nodes (IP addresses). These nodes run the HPE Ezmeral Unified Analytics Software service. Calculate the number of worker nodes based on the VCPUs you enter in step 7. Must be a minimum of 96 VCPUs. The accumulated total VCPU of the worker nodes should match or exceed the number of VCPUs that you enter in step 7. **ATTENTION:** If you plan to use GPU nodes as storage capable hosts to meet the vCPU sizing requirements, contact HPE Support.	3		N/A

* Requires one node for the HPE Ezmeral Coordinator and one node for the workload.

** Requires one node for the HPE Ezmeral Coordinator and three nodes for the workload.

IMPORTANT:

- Either the SSH password or SSH key is required. The SSH pass phrase is optional and only applicable if the SSH key is provided.
- Wall clock time on the hosts in the deployment must be synchronized.

On the **Node Setup** screen, complete the following steps:

1. Upload a YAML file or complete the fields to manually configure the nodes.

   If you upload a YAML file, the system runs a validation check against the file and returns an error message if the file is invalid.

   TIP: A YAML template file is provided and includes the following fields:

   ```
   controlplanes: "" # comma-separated list of ip values
   workers: "" # comma-separated list of ip values
   ssh_username: root
   ssh_password: ""
   ssh_key: "" # base64 encoded string
   ssh_passphrase: ""
   ```

2. Click **Pre-check** to run the installation pre-check script.

   The installation pre-check script runs checks against each of the host machines configured for HPE Ezmeral Unified Analytics Software, including the HPE Ezmeral Coordinator, control plane, and worker hosts. The script also does an aggregated check to verify that the hosts, operating as a cluster, have enough resources to support the installation. If the pre-check script identifies any issues, the system provides detailed messages in the UI, as well as log files for troubleshooting. To view the logs, click **View details**. Installation cannot resume until the pre-check script runs successfully.

3. Click **Next** to proceed to **Installation Details**.

Installation Details

On the **Installation Details** screen, complete the following steps:

1. Complete the following fields:

Field	Description
Installation Name	Enter a unique name for the installation. The installation name must consist of lowercase alphanumeric characters or `-`. For example, `installation-1`. This name becomes the name of the cluster namespace. In the future, if you need to add additional hosts to increase resources for applications, you will use this name as the namespace when adding hosts, as described in Expanding the Cluster on page 113.
Domain Name	Enter a valid DNS domain name to connect to the cluster via the browser. **NOTE:** • The HPE Ezmeral Unified Analytics Software cluster domain name cannot be the same as the DNS host domain name. • Do not enter your corporate top level domain (TLD) name in this field. If you enter the corporate TLD name, you must set up a wildcard record that points all subdomains of the corporate domain to the HPE Ezmeral Unified Analytics Software ingress gateway hosts. • Best practice is to enter a subdomain off the corporate domain. For example, if your corporate domain is `company.com`, you could enter `ezua.company.com` as your domain name. • As you continue the installation process, you will set up wildcard records for the domain name you enter in this field. The DNS name resolution to those records should work for pods and any member of your organization that needs access to HPE Ezmeral Unified Analytics Software.
VCPU	The number of VCPUs that you enter is determined by the number of worker nodes. Typically, 96 VCPUs translates to three worker nodes, and entering 97 would translate to four worker nodes. If you need to distinguish between cores and VCPUs, for example in cases where hyperthreading is enabled, run the `lscpu` tool to accurately determine the VCPUs for your hosts.
High Availability	When selected, three controller nodes are enabled. Currently, HA is available for the workload cluster only. The management cluster does not support HA.
Use GPU	See GPU Support on page 137.
Air Gap Environment	Select this option when installing in an air-gapped environment (no internet access). If you select Air Gap Environment, you must provide the registry details.

Field	Description
Registry URL	Enter the registry URL. Only required for air-gapped environments, but can also be used for a custom image registry in connected environments. Make sure you add the trailing `/` at the end of the URL, as shown in the following example: `my-registry.mip.storage.mycompany.net/ezua/`
Username	Enter the user name for the registry.
Password	Enter the password for the registry.
Registry Insecure	Select this option if the registry is *not* secure. If the registry is secure, do not select this option.
CA Certificate	Upload the CA certificate. See Working with Certs and the Truststore on page 230.
TLS Certificates	- **Use Self Signed Certificate** - Typically only selected for POCs and demos. For production environments, HPE recommends uploading your own certificates (CA certificate and Private Key). - **CA Certificate -** Upload the CA certificate - **Private Key -** Upload the private key. - **Certificate -** Upload additional certificates.

Proxy Details	**NOTE:** The proxy details apply to the HPE Ezmeral Unified Analytics Software application; they do not apply to the host.
	- **HTTP Proxy -** Enter the URL for the proxy data center.
	- **HTTPS Proxy -** Enter the URL for the proxy data center.
	- **No Proxy -** Each of the hosts in the HPE Ezmeral Unified Analytics Software cluster must have the IP addresses of the *coordinator* and *control plane* hosts of the *workload cluster* in the no_proxy list. Add the FQDN of the *master host* in the *workload cluster* OR a *comma-separated list of IP addresses* or *hostnames*. Note that some of the IP addresses in the cluster are required to bypass the proxy settings to reach the internal pod/container entities. Use the following string of IP addresses to bypass the proxy settings:
	`10.96.0.0/12,10.224.0.0/16,10.43.0.0/16,\`
	`.external.hpe.local,localhost,.cluster.local,.svc,\`
	`.default.svc,127.0.0.1`
	For example, if your domain is `ezua.company.com`, you would enter the following string for no_proxy:
	`10.96.0.0/12,10.224.0.0/16,10.43.0.0/16,`
	`.external.hpe.local,localhost,.cluster.local,.svc,\`
	`.default.svc,127.0.0.1, ezua.company.com`
	- **External URL -** This field only applies to the workload nodes and is only required if you select HA for the HPE Ezmeral Unified Analytics Software application. If you want HA for the HPE Ezmeral Coordinator, contact HPE Support before you install on the HPE Ezmeral Coordinator node.

2. Click **Next** to proceed to **User Authentication Details**.

User Authentication Details

Connected and air-gapped installations can use internal or external LDAP. Internal LDAP is typically used for POC and demo scenarios. External LDAP is typically used for production environments. To learn about AD/LDAP servers in detail, see AD/LDAP Servers on page 227.

To add user authentication details, complete the following steps:

1. Either select or do not select the option to use an internal LDAP server.
   - If you select the **Use Internal LDAP Server** option. Provide the following information to create the default Unified Analytics administrative user. This user must be part of your organization and have an organization email, for example `bob@company.com`.
     - Username
     - Full Name
     - Email
     - Password
   - If you *do not* select **Use Internal LDAP Server**, complete the related fields. The user that you enter becomes the default Unified Analytics administrative user. This user must already exist in the AD/LDAP server that you specify.

     List of related fields:
     - Select **Active Directory** if the LDAP is an Active Directory (ADLDAP)
     - Security Protocol
     - LDAP Server Address
     - Server Port
     - Bind DN
     - Bind Password
     - Search Base DN
     - Trust Store File
     - Trust Store Password
     - Username Attribute
     - Fullname Attribute
     - Email Attribute

       NOTE: If the admin performing the installation selects the **Allow Login By Email Address** option, users can sign in using their email address or username; otherwise, users can only sign in with their usernames.

       Even if you do not select the option **Allow Login By Email Address**, you can still specify an email attribute for users, and their email addresses (if available) will be discovered for display purposes. However, in this scenario, users will not be able to sign in to Unified Analytics using their email address.
     - UID Attribute

- GID Attribute
- Group Name
- Group GID
- Username of the default admin user
- Validation options

2. Click **Next** to proceed to **Tools & Frameworks**.

Tools & Frameworks

You do not have to install all of the tools and frameworks packaged with HPE Ezmeral Unified Analytics Software. You have the option of deselecting the following applications:

- Superset
- EzPresto
- Livy
- Feast
- MLDE

Deselect the applications that you do not want to install. Any application that you do not install now can be installed later. Note that options related to the uninstalled applications and frameworks will not appear in the Unified Analytics UI. For example, if you do not install EzPresto, the Data Catalog, Query Editor, and Cached Assets options will not appear in the left navigation panel of the Unified Analytics UI. However, if you choose to install EzPresto in the future, these options become available.

Click **Next** to proceed to **Storage Details**.

Storage Details

Enter the required information to connect Unified Analytics to a storage platform (primary storage) and object store. Unified Analytics supports HPE Ezmeral Data Fabric for primary storage and HPE Ezmeral Data Fabric, HPE GreenLake for File Storage, MinIO, and AWS S3 for object storage.

For details about how to prepare the HPE Ezmeral Data Fabric cluster to be primary storage for Unified Analytics, see Preparing HPE Ezmeral Data Fabric to be Primary Storage for HPE Ezmeral Unified Analytics Software on page 78.

To configure primary and object storage for Unified Analytics, complete the following steps:

1. In the **Data Volumes** section, enter the following information:

Field	Description
Data Source	Select **Ezmeral Data Fabric**
CLDB Hosts	Enter a comma-separated list of CLDB hosts that the HPE Ezmeral Data Fabric administrator provided for the Unified Analytics deployment.
REST Servers	Enter a comma-separarted list of REST servers that the HPE Ezmeral Data Fabric administrator provided for the Unified Analytics deployment.
Tenant Ticket	Enter the content of the tenant ticket that the HPE Ezmeral Data Fabric administrator created for the Unified Analytics deployment.
Username	Enter the username for the HPE Ezmeral Data Fabric user that the HPE Ezmeral Data Fabric administrator created for the Unified Analytics deployment.

Field	Description
Password	Enter the password for the HPE Ezmeral Data Fabric user that the HPE Ezmeral Data Fabric administrator created for the Unified Analytics deployment.
CA Certificate	Apply the CA certificate that the HPE Ezmeral Data Fabric administrator obtained from the HPE Ezmeral Data Fabric for the Unified Analytics deployment.
Mount prefix	Enter the mount prefix that the HPE Ezmeral Data Fabric administrator created for the Unified Analytics deployment.

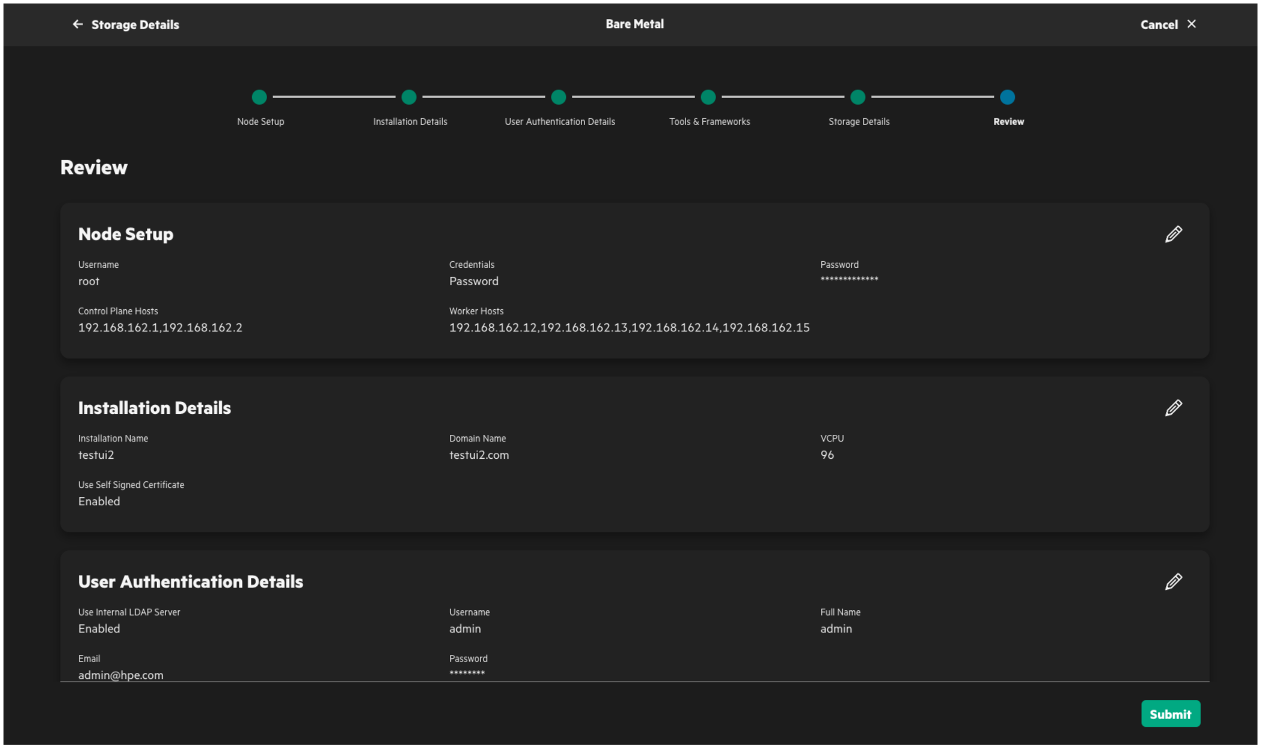
2. In the **Object Store** section, enter the following information:

Field	Description
Object Source	Select which object store you want to connect Unified Analytics to. You can connect Unified Analytics to the following object stores: • HPE Ezmeral Data Fabric Object Store • HPE GreenLake for File Storage • MinIO • AWS
End Point	Enter the endpoint URL for the object store. For secure object stores (https), specify the FQDN in the object store endpoint TLS certificate. For example, if the object store is located at 192.168.0.10 and uses a TLS certificate with the subject name `myobjecstore.example.com`, you would specify the endpoint as: `https://myobjecstore.example.com:9000` Specifying `https://192.168.0.10:9000` would cause a connectin failure.
Access Key	Enter the access key. The access key and secret key must correspond to an IAM account on the object store that has permission to create buckets, read from buckets, and write to buckets on the object store.
Secret Key	Enter the secret key.
Root Certificate	Apply the root certificate. You must provide the CA certificate for the object store. Unified Analytics validates the TLS certificate presented by the object store endpoint. If the object storage is within the same primary storage platform, such as HPE Ezmeral Data Fabric and HPE Ezmeral Data Fabric Object Store, use the same CA file that was used for the primary storage.

3. Click **Review** to proceed.

Review

On the **Review** screen, review all the selections and entries you made before you start the Unified Analytics installation process in the cluster. Click the pencil icon to go back to any of the installation screens and make changes. Click **Submit** when you want to start the installation.



Review

Node Setup

Username: root
Credentials: Password
Password: *************
Control Plane Hosts: 192.168.162.1,192.168.162.2
Worker Hosts: 192.168.162.12,192.168.162.13,192.168.162.14,192.168.162.15

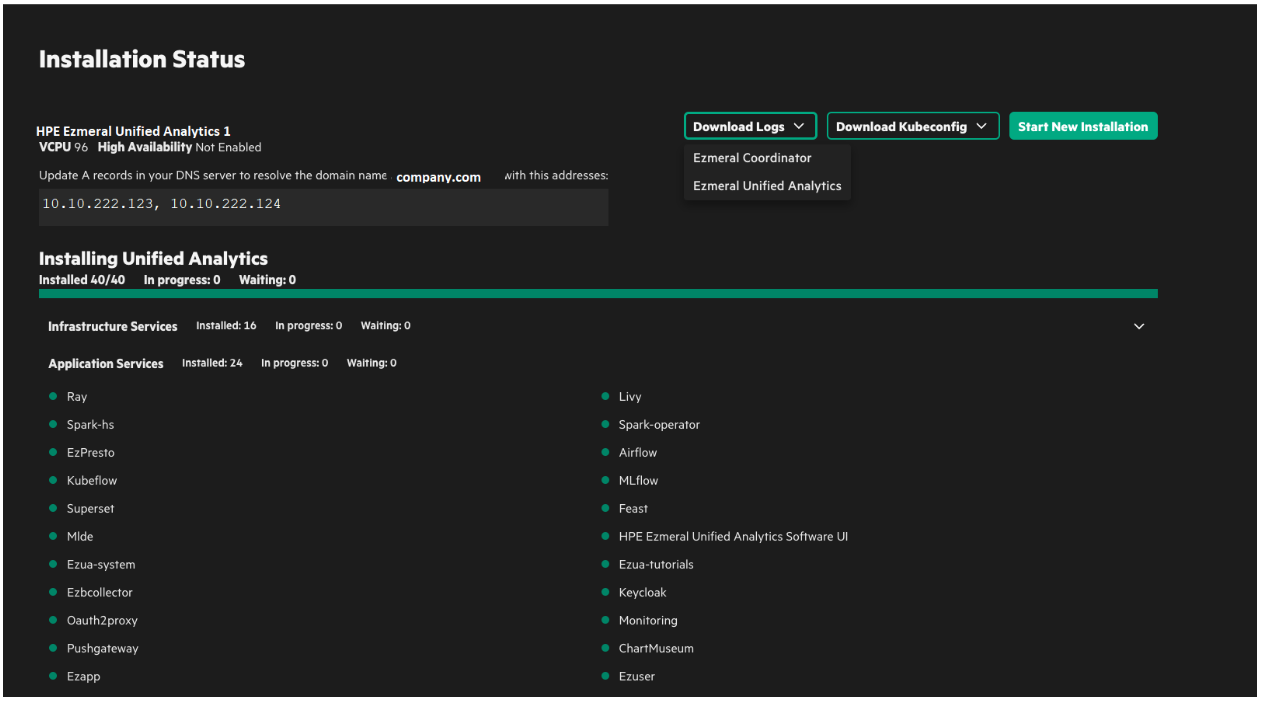
Installation Details

Installation Name: testui2
Domain Name: testui2.com
VCPU: 96
Use Self Signed Certificate: Enabled

User Authentication Details

Use Internal LDAP Server: Enabled
Username: admin
Full Name: admin
Email: admin@hpe.com
Password: ********

After you click **Submit**, the installation of components and applications begins. The **Installation Status** screen displays the installation status of the components and applications as the installation progresses.



Installation Status

HPE Ezmeral Unified Analytics 1
VCPU 96 High Availability Not Enabled

Update A records in your DNS server to resolve the domain name company.com with this addresses:

```
10.10.222.123, 10.10.222.124
```

Installing Unified Analytics
Installed 40/40 In progress: 0 Waiting: 0

Infrastructure Services Installed: 16 In progress: 0 Waiting: 0

Application Services Installed: 24 In progress: 0 Waiting: 0

- Ray
- Spark-hs
- EzPresto
- Kubeflow
- Superset
- Mlde
- Ezua-system
- Ezbcollector
- Oauth2proxy
- Pushgateway
- Ezapp
- Livy
- Spark-operator
- Airflow
- MLflow
- Feast
- HPE Ezmeral Unified Analytics Software UI
- Ezua-tutorials
- Keycloak
- Monitoring
- ChartMuseum
- Ezuser

IMPORTANT:

- Note the IP addresses on this screen. You need these to complete the post installation steps and update your DNS A and DNS records.
- If the installation fails at any point, click **Download Logs** to access the logs files for the HPE Ezmeral Coordinator or **Ezmeral Unified Analytics**. The **Ezmeral Unified Analytics** logs include both the infrastructure services and application services logs. Review the log files to troubleshoot the failure. If you cannot resolve the installation failure issue, contact HPE Support.

TIP:

- The first status update shows the progress of the HPE Ezmeral Coordinator. When complete, the **Download Kubeconfig** button appears on the screen. You can download the `kubeconfig` for the HPE Ezmeral Coordinator and worker cluster.
- Clicking **Open HPE Ezmeral Unified Analytics Software** launches the UI.
- Clicking **Start New Installation** installs another instance of HPE Ezmeral Unified Analytics Software.

Post Installation Steps

See Post Installation Steps on page 82.

Related concepts

Installation Prerequisites on page 73
Lists the prerequisites for HPE Ezmeral Unified Analytics Software installation on user-provided hosts in connected (internet access) and air-gapped (no internet access) environments.

Post Installation Steps on page 82
Provides steps to complete after you install HPE Ezmeral Unified Analytics Software on user-provided hosts.

AD/LDAP Servers on page 227
Describes the differences between the internal OpenLDAP server in HPE Ezmeral Unified Analytics Software and external AD/LDAP servers. Also describes some of the server-related configuration options that you set during installation.

Related tasks

Managing Imported Tools and Frameworks on page 122
Describes how to configure, delete, and update imported tools and frameworks in HPE Ezmeral Unified Analytics Software.

More information

Administration on page 63
Provides information about managing applications and clusters in HPE Ezmeral Unified Analytics Software.

Configuring Included Frameworks on page 124
Describes how to configure tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

Upgrading Included Frameworks on page 125
Describes how to upgrade tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

Installation Prerequisites

Lists the prerequisites for HPE Ezmeral Unified Analytics Software installation on user-provided hosts in connected (internet access) and air-gapped (no internet access) environments.

You can install HPE Ezmeral Unified Analytics Software on user-provided hosts. User-provided hosts are machines that meet the prerequisite criteria listed on this page. A user-provided host is a bare-metal machine or virtual machine (VM) that meets the documented prerequisites. HPE Ezmeral Unified Analytics Software supports bare-metal and VM installations on AWS, GCP, and Azure.

If you need to add additional user-provided hosts to increase the amount of resources for applications and users after you install HPE Ezmeral Unified Analytics Software, you can expand the cluster, as described in Expanding the Cluster on page 113.

❗ **IMPORTANT:**

- The HPE Ezmeral Unified Analytics Software product downloaded includes the software binaries, installation script, Air Gap Utility (for air-gapped environments), and a README.txt file. After purchasing HPE Ezmeral Unified Analytics Software, the downloads are made available to you through the **Access your products** button in the **HPE Subscription Electronic Receipt** email that you receive from HPE.
- When creating a domain name, opt for a subdomain name that is only used for HPE Ezmeral Unified Analytics Software. For example, if your top level domain (TLD) name is `company.com`, use a subdomain name such as `ezua.company.com`.

Software Binaries

The README.txt file included with the product provides instructions for downloading and extracting the HPE Ezmeral Unified Analytics Software binaries that are required to install the product, including the Air Gap Utility.

Air Gap Utility (Required only for air-gapped environments)

Use the Air Gap Utility to get the required container images. Create a local repository if you do not already have one. In addition to setting up a local repository, you must also set up a RHEL/Rocky 8-based yum repository. The installer runs `yum` commands against this repository. See Using the Air Gap Utility on page 95 for additional information. For operating system support, see the Operating System on page 205 support matrix.

Host Machines

HPE Ezmeral Unified Analytics Software installation requires two types of host machines with the following minimum storage requirements:

Table

Machine Type	VCPU	Memory Size (GB)	Disk Size (Disk Count)	Machine Count
Control plane (HPE Ezmeral Coordinator/ Management Cluster)	4	32	500 (1)	2
Workload	32	128	500 (2)	3
Launcher Host[1]	2	4		

[1] This is the host that runs the installation script. This host is separate from the hosts that deploy HPE Ezmeral Unified Analytics Software. This host must also have Docker version 20.10 (with a minimum of 20GB storage) and a firewall allowed on port 8080.

IMPORTANT:

- If you plan to use GPU nodes as storage-capable hosts to meet the vCPU sizing requirements, contact HPE Support.
- Host machines must have a `sudo` password.

Mount Points

The supported mount points and their minimum sizes depend 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.

Table

Mount Point	Minimum Size (GB)	Purpose
`/`	70	Root file system where the Unified Analytics 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 Unified Analytics software. `/opt/ezkube` (on Kubernetes hosts hosts only), `/opt/bluedata`, and `/opt/hpe` are used to install Unified Analytics.

VCPU Sizing Guidelines	• The combined VCPUs of worker hosts should be no less than 96 for deploying HPE Ezmeral Unified Analytics Software services and apps; otherwise, some services cannot start due to lack of resources. If the machine configuration has 16 VCPU and 64 GB of memory, HPE recommends using a minimum of six (6) machines. • When installing HPE Ezmeral Unified Analytics Software, the VCPU option (on the Installation Details screen) should be equal to or less than the total VCPUs of the combined capability of the worker hosts; otherwise, the installation will fail due to lack of resources.
DNS Configuration	The DNS configuration requires that: • All hosts have A records added to DNS. • The name resolution works forward and backward. • The FQDN is a maximum of 63 characters. • All hosts must be part of the DNS domain and be able to resolve the FQDNs.

Primary Storage

Starting in HPE Ezmeral Unified Analytics Software version 1.5, Unified Analytics no longer provides an internal data fabric for storage on worker nodes. Instead, Unified Analytics supports connections to a primary storage platform that you provide and configure. Primary storage is an external storage platform that Unified Analytics uses to store and process the data generated by the applications and frameworks that run in your Unified Analytics cluster. Unified Analytics connects to the external storage platform through a CSI driver. During Unified Analytics installation, you provide the information that the CSI driver needs to make the connection to the external storage platform. Currently, Unified Analytics supports HPE Ezmeral Data Fabric as primary storage.

Before you install Unified Analytics, the HPE Ezmeral Data Fabric administrator must prepare the HPE Ezmeral Data Fabric cluster. For instructions, see Preparing HPE Ezmeral Data Fabric to be Primary Storage for HPE Ezmeral Unified Analytics Software on page 78.

Object Storage

Starting in HPE Ezmeral Unified Analytics Software version 1.5, Unified Analytics no longer provides internal object storage. Instead, Unified Analytics supports connections to S3-compatible object storage that you provide and configure during installation. Currently, Unified Analytics supports the following S3-compatible object stores:

- HPE Ezmeral Data Fabric Object Store
- HPE GreenLake for File Storage
- AWS S3
- MinIO

Unified Analytics must have permission to create buckets, read from the bucks, and write to the buckets. When configuring object storage during installation, note the following points:

- For secure object stores (https), specify the FQDN in the object store endpoint TLS certificate. For example, if the object store is located at 192.168.0.10 and uses a TLS certificate with the subject name `myobjecstore.example.com`, you would specify the endpoint as:

  ```
  https://myobjecstore.example.com:9000
  ```

  Specifying https://192.168.0.10:9000 would cause a connection failure.

- You must provide the CA certificate for the object store. Unified Analytics validates the TLS certificate presented by the object store endpoint. If the object storage is within the same primary storage platform, such as HPE Ezmeral Data Fabric and HPE Ezmeral Data Fabric Object Store, use the same CA file that was used for the primary storage.

- The access key and secret key must correspond to an IAM account on the object store that has permission to create buckets, read from buckets, and write to buckets on the object store. For example, the following IAM policy is acceptable:

  ```
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "AllAccess",
        "Action": "s3:*",
        "Effect": "Allow",
        "Resource": [
           "arn:aws:s3:::*",
        ],
        "Principal": "*"
      }
    ]
  }
  ```

Operating System

HPE Ezmeral Unified Analytics Software supports RHEL 8.8. All machines serving as hosts must run the same OS. Both GPU and non-GPU hosts are supported with RHEL 8.8.

Hewlett Packard Enterprise strongly recommends using only dedicated hosts with clean OS installations on them. Installing HPE Ezmeral Unified Analytics Software 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 can cause HPE Ezmeral Unified Analytics Software to function unpredictably.

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

HPE Ezmeral Unified Analytics Software does not support upgrades between major OS versions. For example, if you are migrating from OS version 7.x to 8.x, you must perform a new installation (not an upgrade), and then install HPE Ezmeral Unified Analytics Software.

RHEL 8.8 Requirements

HPE Ezmeral Unified Analytics Software has the following RHEL requirements:

- RHEL systems must have active, valid subscriptions in order to access the RHEL RPM repositories.
- Firewall is supported only in `iptables` mode for RHEL 8.8.

TIP:

- The GPU operator does not support Ubuntu or Rocky.
- Rocky works in non-GPU environments.
- For information related to operating systems and operating-system version support, contact HPE Support.

Network

HPE Ezmeral Unified Analytics Software has the following network requirements:

Table

VM	DNS	Network	IP Address
Single vNIC, static IP	DNS server to resolve the FQDN	Single network connecting all machines.	Static: 2 controller machine + 3 worker machines

Port Access

See Ports Used by HPE Ezmeral Unified Analytics Software on page 100.

GPU (Optional)

HPE Ezmeral Unified Analytics Software supports GPU on user-provided hosts. For GPU configuration information, see GPU Support on page 137.

Preparing HPE Ezmeral Data Fabric to be Primary Storage for HPE Ezmeral Unified Analytics Software

Provides the steps that an HPE Ezmeral Data Fabric administrator (`mapr`) must complete before an HPE Ezmeral Unified Analytics Software administrator installs Unified Analytics and configures HPE Ezmeral Data Fabric as primary storage for Unified Analytics.

During installation, the HPE Ezmeral Unified Analytics Software administrator must provide the CSI driver (KDF-CSI) with the information needed to successfully connect to an external HPE Ezmeral Data Fabric cluster. The CSI driver requires the following information:

- List of CLDB hosts
- List of API servers
- Tenant ticket
- Username
- Password
- CA certificate
- Mount prefix

The HPE Ezmeral Data Fabric cluster administrator (`mapr`) can obtain this information while preparing the HPE Ezmeral Data Fabric cluster to be accessed by Unified Analytics and its users.

HPE Ezmeral Data Fabric preparation includes:

- Specifying user information for the Unified Analytics deployment

- Specifying the mount prefix for the Unified Analytics deployment
- Creating a new user in the HPE Ezmeral Data Fabric cluster
- Giving the new user permissions to access the HPE Ezmeral Data Fabric cluster
- Creating a dedicated volume for the new user
- Creating a tenant ticket for the new user
- Obtaining the root and signing CA certificates for the HPE Ezmeral Data Fabric cluster
- Obtaining a list of CLDB hosts in the HPE Ezmeral Data Fabric cluster
- Obtaining a list of API servers in the HPE Ezmeral Data Fabric cluster

The following section provides the preparation steps.

Preparing the HPE Ezmeral Data Fabric Cluster

As you complete the steps required to prepare the HPE Ezmeral Data Fabric cluster, take note of the following information, as this information is required during the installation of Unified Analytics:

- Username and password for the HPE Ezmeral Data Fabric user
- Mount prefix
- Contents of the tenant ticket
- Contents of the HPE Ezmeral Data Fabric CA certificate
- List of CLDB hosts
- List of API/REST servers

To prepare the HPE Ezmeral Data Fabric cluster, complete the following steps:

1. SSH in to one of the nodes in the external HPE Ezmeral Data Fabric cluster:

   ```
   ssh <node-ip-address>
   ```

2. Specify the user information for your Unified Analytics deployment:

   ```
   export USER=ezua
   export GROUP=ezua
   export USERID=7000
   export GROUPID=7000
   export PASSWORD=$(openssl rand -base64 12)
   ```

   TIP:

   - If you have multiple Unified Analytics deployments, HPE recommends having a dedicated user for each deployment.
   - If the password is user-provided instead of auto-generated, properly escape special characters or sequences, such as `$!`, to prevent the shell from replacing them and returning unexpected results.

3. Specify the mount prefix for your Unified Analytics deployment:

   ```
   export MOUNT_PREFIX=/ezua
   ```

   TIP:

   - If you have multiple Unified Analytics deployments, HPE recommends having a dedicated mount prefix for each deployment.
   - Do **not** use `/mapr` as the mount prefix, as `/mapr` denotes the global namespace and some tools, including the hadoop client, are configured to reference this directory for their operations.

4. Create a new HPE Ezmeral Data Fabric user by running the following commands on **all nodes** in the HPE Ezmeral Data Fabric cluster:

   ```
   sudo groupadd -g ${GROUPID?} ${GROUP?}
   sudo adduser -g ${GROUP?} -M -u ${USERID?} ${USER?}
   echo "${USER?}:${PASSWORD?}" | sudo chpasswd
   ```

   TIP:

   - Use the same password on all nodes. For example, run the `openssl` command one time.
   - For additional information, see User Accounts.
   - An alternative option for this step is to add this user to your LDAP.

5. Verify the password:

   ```
   echo ${PASSWORD}
   ```

6. Verify that you can log in as the new user:

   ```
   echo ${PASSWORD} | maprlogin password -user ${USER?}
   ```

7. Log in as the `mapr` administrative user:

   ```
   maprlogin password -user mapr
   ```

8. Assign the `create volume` ACL to the HPE Ezmeral Data Fabric user:

   ```
   maprcli acl edit -type cluster  -user ${USER?}:login,cv
   ```

9. Create a volume that this user can access under a dedicated prefix:

   ```
   maprcli volume create -name ezua-base-volume-${USER?} -path $
   {MOUNT_PREFIX?} \
   -createparent true -type rw -json -rootdiruser ${USER?} -rootdirgroup $
   {GROUP?}
   ```

10. Create a tenant ticket for this user:

    ```
    maprlogin generateticket -type tenant -user ${USER?} -out /tmp/
    maprtenantticket-${USER?}
    ```

    TIP: Unified Analytics and the CSI driver do not currently support rotating tickets; therefore, the system checks the ticket expiration date to verify that it is at least 100 years from the current date. By default, tenant tickets have LIFETIME duration (10000 years) to ensure that the ticket does not expire. For additional information, including how to set the duration, see maprlogin.

11. Inspect the tenant ticket:

    ```
    maprlogin print -ticketfile /tmp/maprtenantticket-${USER?}
    ```

12. Obtain the tenant ticket:

    ```
    cat /tmp/maprtenantticket-${USER?}
    ```

13. Obtain the root and signing CA of the HPE Ezmeral Data Fabric cluster:

    ```
    sudo cat /opt/mapr/conf/ca/chain-ca.pem
    ```

14. Obtain the endpoints of the HPE Ezmeral Data Fabric cluster:

    ```
    maprcli node list -columns hn,ip -filter svc==cldb
    ```

    TIP:

    - Filtering nodes using `svc==cldb` returns the nodes currently running the CLDB service. If the CLDB service is configured on a node, but not running for some reason, that node will not appear in the results. Alternatively, you can filter nodes using `csvc==cldb`, which returns a list of nodes configured with the CLDB service.
    - If MAPR_EXTERNAL is configured, the `maprcli node list` command returns an `extIp` column, which lists the external IP addresses of the nodes in the HPE Ezmeral Data Fabric cluster. Unified Analytics uses the external IP addresses to access the HPE Ezmeral Data Fabric cluster. When you provide Unified Analytics with the endpoints, use the external IP addresses; do not use the local hostnames.

      ```
      maprcli node list -columns hn,ip -filter svc==cldb
      hostname                      ip          extIp
      ip-10-0-0-100.ec2.internal   10.0.0.100
      10.10.100.110:5660,10.10.100.120:5692
      ```

      In this example, you would provide the `extIp` (`10.10.100.110`), not the `hostname` (`ip-10-0-0-100.ec2.internal`). For additional information, see MAPR_EXTERNAL Environment Variable.

    a. Obtain a list of the CLDB hosts and then append port `:7222` to each host in a comma-separated list:

       ```
       maprcli node list -columns hn,ip -filter svc==cldb
       ```

b. Obtain a list of API servers and then append port `:8443` to each host in a comma-separated list:

```
maprcli node list -columns hn,ip -filter svc==apiserver
```

IMPORTANT: Verify that the Unified Analytics nodes can access the HPE Ezmeral Data Fabric nodes. For example, verify that the firewall is not blocking the connections. See Port Information.

Post Installation Steps

Provides steps to complete after you install HPE Ezmeral Unified Analytics Software on user-provided hosts.

After you have successfully installed HPE Ezmeral Unified Analytics Software, complete the following post installation steps:

1. Note the ingress node IP address on the *Installation Status* screen and configure or update your DNS A records to point to this address.

   NOTE: If you do not see the IP address, refer to the User Interface on page 194 troubleshooting page to resolve the issue.

2. Update A records in your DNS server to resolve your domain name with the addresses provided. You can do this by pointing your DNS record to the two IP addresses that display on the *Installation Status* screen. It may take a few minutes for your DNS settings to propagate.

3. Access the HPE Ezmeral Unified Analytics Software home page by clicking the green bar that reads **Open HPE Ezmeral Unified Analytics Software**.

4. Note the Platform ID in the window. You need the Platform ID to activate the HPE Ezmeral Unified Analytics Software service.

5. To activate the HPE Ezmeral Unified Analytics Software service, follow the steps listed in Service Activation and Billing in Connected Environments on page 92.

6. (Air-gapped environments only) For a successful Airflow installation, manually set the HTTP proxy or configure Airflow to point to your internal GitHub repository, as described in Airflow DAGs Git Repository on page 355. This step is required in an air-gapped environment because Airflow is pre-configured to pull DAGs from an HPE GitHub repository. In air-gapped environments, Airflow cannot access the HPE repository.

7. Run the following command to update the SPIFFE CSI driver:

   ```
   kubectl -n spire set image ds spire-spiffe-csi-driver
   spiffe-csi-driver=ghcr.io/spiffe/spiffe-csi-driver:0.2.5
   ```

   For details, see Host (Node) Management on page 170.

Installing HPE Ezmeral Unified Analytics Software on OpenShift

Provides the prerequisites and steps for installing HPE Ezmeral Unified Analytics Software in an OpenShift cluster and also lists the current limitations.

To install HPE Ezmeral Unified Analytics Software in an OpenShift cluster, complete the following steps:

1. Complete the Prerequisites.

- A. Verify that the VMs (nodes) in the OpenShift cluster meet the installation requirements on page 83
- B. Apply labels to the storage nodes on page 85

- C. (Air-Gapped Only) Inject HPE Ezmeral Unified Analytics Software images into your local repository on page 86
- D. (Air-Gapped Only) Apply the image registry certificate on page 88
- E. Install the CertManager on page 89

2. Install HPE Ezmeral Unified Analytics Software.

- Currently, you can install HPE Ezmeral Unified Analytics Software through the Installer Web UI only. See Install HPE Ezmeral Unified Analytics Software with the Installer Web UI on page 89.
- The ability to install HPE Ezmeral Unified Analytics Software manually (through a CLI) is coming soon.

NOTE: Installation in an AWS cluster on OpenShift can fail during MLDE installation because there is no DNS mapping. For this scenario, see Creating a DNS Mapping on page 91 before installation.

NOTICE:

Temporary Limitations

Upcoming releases will address the following temporary limitations:

- Some Pods in the HPE Ezmeral Unified Analytics Software platform must run as `root`. To avoid permission-denied errors, the HPE Ezmeral Unified Analytics Software installation process sets the `anyuidsecurity` context in some namespaces to allow root-based execution.
- Some Pods in the HPE Ezmeral Unified Analytics Software platform require HostPath-based volume mounts, which OpenShift denies by default. These Pods need permission to mount HostPath volumes. Permission to mount these volumes is granted by the `securityContext` parameter, with `privileged` set to `true`.
- Read more about OpenShift security context constraints here.

A. Verify that the VMs (nodes) in the OpenShift cluster meet the installation requirements

The following table lists the requirements:

Prerequisite	Details
Operating System	RHEL8.8 based RHCOS
OpenShift	An OpenShift 4.12.x cluster must be dedicated to HPE Ezmeral Unified Analytics Software.

Prerequisite	Details
Storage	Minimum of 3 nodes, each with at least: • 32 vCPUs • 128 GB RAM • 2 additional disks with a minimum of 500 GB **NOTE:** Currently, HPE Ezmeral Unified Analytics Software does not natively support three-node compact cluster mode. HPE does not recommend using three-node compact cluster mode; however, it is possible with manual intervention.
GPU	• Install the NFD and GPU operators and then create instances of these through the OpenShift console. • Verify that the GPU is active. • For the supported GPU models, see GPU Models on page 205.

<table>
<tr><th>Prerequisite</th><th>Details</th></tr>
<tr><td>

Configure private image registry access

</td><td>

To configure access to the private image registry:

1. Update the CA certificate to make HTTPS access possible.
2. Append a new pull secret to the existing pull secret:
   a. To download the pull secret, run:

   ```
   oc get secret/pull-secret -n openshift-config --template='{{index .data ".dockerconfigjson" | base64decode}}' \
   <pull_secret_location>

   //<pull_secret_location> = path to the pull secret file
   ```

   b. To add the new pull secret, run:

   ```
   oc registry login --registry="<registry>" --auth-basic="<username>:<password>" --to=<pull_secret_location>

   //<registry> = new registry; you can enter multiple repositories w/in the same registry;
   //for example: --registry="<registry/my-namespace/my-repository>"
   //<username>:<password> = credentials for the new registry
   //<pull_secret_location> = path to the pull secret file
   ```

3. To update the global pull secret for your cluster, run:

   ```
   oc set data secret/pull-secret -n openshift-config --from-file=.dockerconfigjson=<pull_secret_location>

   //<pull_secret_location> = path to the new pull secret file
   ```

   This update occurs on all nodes and takes some time to complete.
4. On all nodes, view the `/var/lib/kubelet/config.json` file to verify that your private registry was added.

For additional information, see Using image pull secrets.

</td></tr>
</table>

B. Apply labels to the storage nodes

Tag your storage nodes (non-GPU worker nodes) with the `"hpe.com/dataplatform"="true"` label, as shown in the following example that uses generic DNS names:

```
kubectl label no worker0.user01.ezfab.local "hpe.com/dataplatform"="true"
kubectl label no worker1.user01.ezfab.local "hpe.com/dataplatform"="true"
kubectl label no worker2.user01.ezfab.local "hpe.com/dataplatform"="true"
kubectl label no worker3.user01.ezfab.local "hpe.com/dataplatform"="true"
```

NOTE: You need *at least* three storage nodes in your cluster that meet this requirement. This is not required for all nodes. Refer to the section A (above) for details about the required configuration for each storage node.

C. (Air-Gapped Only) Inject HPE Ezmeral Unified Analytics Software images into your local repository

For an **air-gapped** installation, you must inject the HPE Ezmeral Unified Analytics Software images into a local repository that you will use to bootstrap the installation process. For this purpose:

- HPE recommends having an empty dedicated image registry. You can also use an existing image registry with other pre-existing images.
- Run the HPE Ezmeral Airgap Utility from a connected host. The Airgap Utility connects to the HPE Greenlake image repository marketplace to download the images into your local registry.

  To inject images into your local repository, **create a local registry (optional)** and **download the images (required)**:

  (Optional) Create a local registry.

  You have many options to create a local registry. If you already have a registry or want to follow your own procedure to set one up, skip to step 2 (Download Images).

  The registry can be hosted on a container, virtual machine, or BareMetal. This document describes how to set up a registry inside a container using the `podmanutility`. The container OS is RHEL8.

  To create a local registry, complete the following steps:

  1. On a fresh RHEL BareMetal/VM, deploy all the utilities required to create the container:

     ```
     yum module enable -y container-tools:rhel8

     yum module install -y container-tools:rhel8
     ```

  2. Install the additional dependencies required for the process:

     ```
     yum install -y httpd-tools jq wget
     ```

  3. Create the following directories:

     - certs/: stores certificates to enable https access to the registry
     - auth/: authentication files for the registry
     - data/: location where the registry stores all the images

       To create the directories, run:

       ```
       mkdir -p /local_registry/{certs, auth, data}
       ```

       Later, you will mount these directories to the registry container.

4. (Optional) Create self-signed certificates. Complete this step to make your registry accessible through HTTPS. You can also use a company-wide certificate. In that case, simply copy your certificate to the `local_registry/certs` directory and skip to the next step.

   NOTE: You can use the same certificate across more than one registry.

   There are many ways to use `openssl` to create a self-signed certificate, for example:

   ```
   openssl req -newkey rsa:4096 -nodes -sha256 -keyout
   <$KEY_FILE_LOCATION> -x509 -days 365 -subj "/
   CN=<$CERTIFICATE_NAME>" -addext "subjectAltName =
   DNS:<$FULL_DNS>" -out <$CRT_FILE_LOCATION>

   //Example:
   openssl req -newkey rsa:4096 -nodes -sha256 -keyout /local_registry/
   certs/domain.key -x509 -days 365 -subj "/CN=Myname" -addext
   "subjectAltName = DNS:*.example.com" -out /local_registry/certs/
   domain.crt
   ```

   You must copy this certificate file to the standard location of the operating system. For RHEL, the standard cert location is `/etc/pki/ca-trust/source/anchors`:

   ```
   cp /local_registry/certs/domain.crt /etc/pki/ca-trust/source/anchors/
   ```

   After you copy the file, run:

   ```
   update-ca-trust
   ```

5. Create access credentials to the registry to keep it secure. You can skip this step for anonymous access.

   ```
   htpasswd -bBc /local_registry/auth/<$PASSWORD_FILENAME> <$USERNAME>
   <$PASSWORD>

   //Example:
   htpasswd -bBc /local_registry/auth/htpd user01 admin123
   ```

6. Expose the registry on port 5000. Add this rule to `firewalld` to open the port and make it available.

   ```
   firewall-cmd --zone=public --permanent --add-port=5000/tcp
   firewall-cmd reload
   ```

7. Create the container to use as local registry. In this example, `podman` is used to create the container; however, you can use any container utility that you prefer:

   ```
   podman run -d --name <$REGISTRY NAME> -p <$PORT>:<$PORT> \
   -v <$DATA_DIRECTORY>:/var/lib/registry:z \
   -v <$AUTH_DIRECTORY>:/auth:z \
   -v <$CERT_DIRECTORY>:/certs:z \
   -e "REGISTRY_AUTH=htpasswd" \
   -e "REGISTRY_AUTH_HTPASSWD_REALM=<$REALM_NAME>" \
   -e "REGISTRY_HTTP_SECRET=<$PHRASE_FOR_SECRET>" \
   -e "REGISTRY_AUTH_HTPASSWD_PATH=<$PATH_TO_AUTH_FILE>" \
   -e "REGISTRY_HTTP_TLS_CERTIFICATE=<$PATH_TO_CERT_FILE>" \
   -e "REGISTRY_HTTP_TLS_KEY=<$PATH_TO_KEY_FILE>" \
   <$REGISTRY_IMAGE>

   //Example:
   podman run -d --name local-registry -p 5000:5000 \
   -v /local_registry/data:/var/lib/registry:z \
   -v /local_registry/auth:/auth:z \
   -v /local_registry/certs:/certs:z \
   -e "REGISTRY_AUTH=htpasswd" \
   -e "REGISTRY_AUTH_HTPASSWD_REALM=my-local-registry" \
   -e "REGISTRY_HTTP_SECRET=ALongRandomSecretForLocalRegistry" \
   -e "REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpd" \
   -e "REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt" \
   -e "REGISTRY_HTTP_TLS_KEY=/certs/domain.key" \
   docker.io/library/registry:2
   ```

8. Use `curl` to access the registry and test that the registry is up and running:

   ```
   curl  -u <$USERNAME>:<$PASSWORD> -k -X GET https://$
   (hostname -f):5000/v2/_catalog

   //Example:
   curl -u user01:admin123 -k -X GET https://
   local-registry.example.com:5000/v2/_catalog
   ```

(Required) Download the images.

To download the images, refer to Using the Air Gap Utility on page 95 for information about pulling HPE Ezmeral Unified Analytics Software images into the local registry.

D. (Air-Gapped Only) Apply the image registry certificate

You can configure your **air-gapped** registry with HTTP or HTTPS (see previous steps). To make it accessible using the HTTPS protocol, you need to add a certificate to the registry. This certificate can be a self-signed certificate (see previous steps) or a company-wide common certificate. The same certificate can be used for multiple registries. If there are multiple registries and all of them are configured with different certificates, the OpenShift configuration should be updated with all the certificates. Follow this procedure to update the registry certificate on your OpenShift cluster.

Create a `config` map with all the certificates for accessing multiple registries. The following syntax shows how to create one `config` map with one registry and one certificate.

```
kubectl create -n openshift-config cm <$REGISTRY_CONFIG_NAME>
--from-file=<$REGISTRY_URL_WITHOUT_PROTOCOL>=<$CERTIFICATE_FILENAME>
```

If you have more than one registry and more than one certificate, run this instead:

```
kubectl create -n openshift-config cm <$REGISTRY_CONFIG_NAME> \
      --from-file=<$REGISTRY_URL_WITHOUT_PROTOCOL>= <$CERTIFICATE_FILENAME>
\
      --from-file=<$REGISTRY_URL_WITHOUT_PROTOCOL>= <$CERTIFICATE_FILENAME>

//Example:
kubectl create -n openshift-config cm
image-registry-config --from-file=image-registry.example.com=registry.crt
kubectl create -n openshift-config cm multiple-registry-config -\
      --from-file=image-registry.example.com=registry.crt \
      --from-file=image-registry.example.com..5000=registry.crt \
      --from-file=new-image-registry.example.com=newCert.crt
```

Once the `configmap` is available, patch that `configmap` with the existing OpenShift config:

```
kubectl patch image.config.openshift.io cluster --type merge -p '{"spec":
{"additionalTrustedCA":{"name":"<$REGISTRY_CONFIG_NAME>"}}}'

//Example:
kubectl patch image.config.openshift.io cluster --type merge -p '{"spec":
{"additionalTrustedCA":{"name":"multiple-registry-config"}}}'
```

E. Install the CertManager

Install the cert manager on the OpenShift cluster. The version should be higher than 1.10.

To install CertManager, run:

```
kubectl apply -f https://github.com/cert-manager/cert-manager/releases/
download/v1.13.1/cert-manager.yaml
```

Install HPE Ezmeral Unified Analytics Software with the Installer Web UI

To install HPE Ezmeral Unified Analytics Software on OpenShift through the Installer Web UI, complete the followign steps:

1. Run the installation script that was provided with the software bundle. The host on which you run this command must be connected to the internet (the Web UI image is public for the specific version of HPE Ezmeral Unified Analytics Software that you are installing) or must point to a local registry where you pre-pulled the Web UI image.

   Running the installation script opens the launcher that guides you through the prompts to start the Installer Web UI.

   - For a **connected** environment, run:

     ```
     ./start_ezua_installer_ui.sh
     ```

   - For an **air-gapped** environment, run the following command and provide the URL of the image repository that you configured as a prerequisite:

     ```
     ./start_ezua_installer_ui.sh --image <$PRIVATE_REGISTRY>/us.gcr.io/
     mapr-252711/hpe-ezua-installer-ui
     ```

2. Copy the OpenShift admin `kubeconfig` (certificate-based `kubeconfig`) to the UI installer.

NOTE:

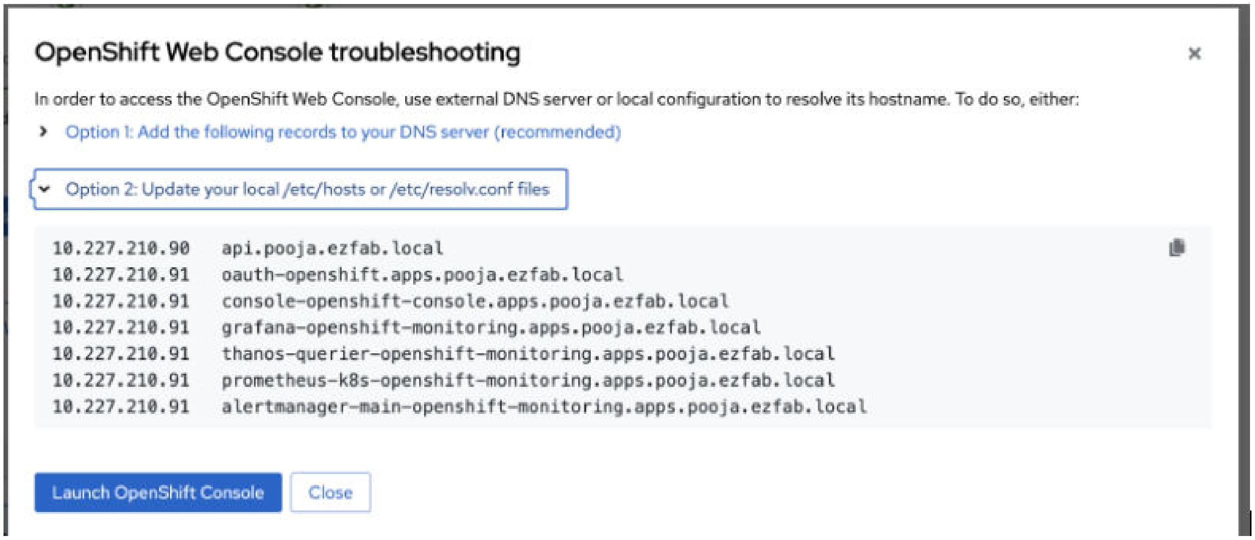
- The UI installer is a container that accesses the OpenShift cluster via `kubectl` commands. You must give the UI installer container `kubectl` access to the OpenShift cluster.
- In a **connected** environment, you can download the OpenShift admin `kubeconfig` from the OpenShift console.
- In an **air-gapped** environment, use the admin `kubeconfig` that was generated during installation.

a. Once you have the `kubeconfig`, run the following command to place it in the container running the HPE Ezmeral Unified Analytics Software Web UI Installer (located at `~/.kube/config`):

```
docker cp <$PATH_TO_ADMIN_KUBECONFIG><$CONTAINER_ID>:/root/.kube/
config
```

b. Update the hosts entries in the Web UI Installer so it can reach the OpenShift cluster.

- In a **connected environment**, you can find the hosts entries of your OpenShift cluster in the OpenShift console. In the OpenShift console, go to **Clusters** on the left and then select the cluster on which you are installing HPE Ezmeral Unified Analytics Software. Under the **Installation Progress** card, click **Not Able to Access the Web Console?**. In the dialog that opens, copy the list of hosts:



OpenShift Web Console troubleshooting

In order to access the OpenShift Web Console, use external DNS server or local configuration to resolve its hostname. To do so, either:

×
› Option 1: Add the following records to your DNS server (recommended)
⌄ Option 2: Update your local /etc/hosts or /etc/resolv.conf files

```
10.227.210.90    api.pooja.ezfab.local
10.227.210.91    oauth-openshift.apps.pooja.ezfab.local
10.227.210.91    console-openshift-console.apps.pooja.ezfab.local
10.227.210.91    grafana-openshift-monitoring.apps.pooja.ezfab.local
10.227.210.91    thanos-querier-openshift-monitoring.apps.pooja.ezfab.local
10.227.210.91    prometheus-k8s-openshift-monitoring.apps.pooja.ezfab.local
10.227.210.91    alertmanager-main-openshift-monitoring.apps.pooja.ezfab.local
```

Example: Screenshot from the OpenShift console that shows the hosts of an example OpenShift cluster.

- In an **air-gapped environment**, copy the DNS entries (used during installation) to the Web UI Installer:

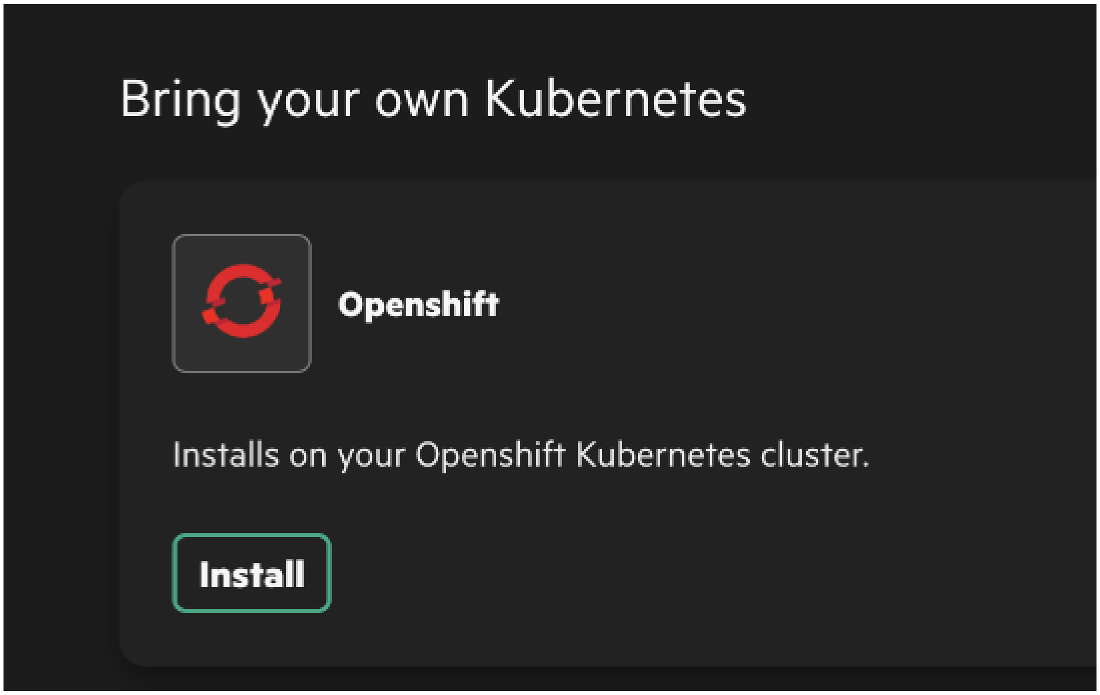
  1. To `exec` into the Web UI Installer container, run:

     ```
     docker exec --it <$CONTAINER_ID> bash
     ```

  2. Edit the `/etc/hosts` file and add the host entries.

3. Navigate back to the launcher that opened when you ran the installation script to start the Installer Web UI.

4. Select **Install** in the **OpenShift** tile.

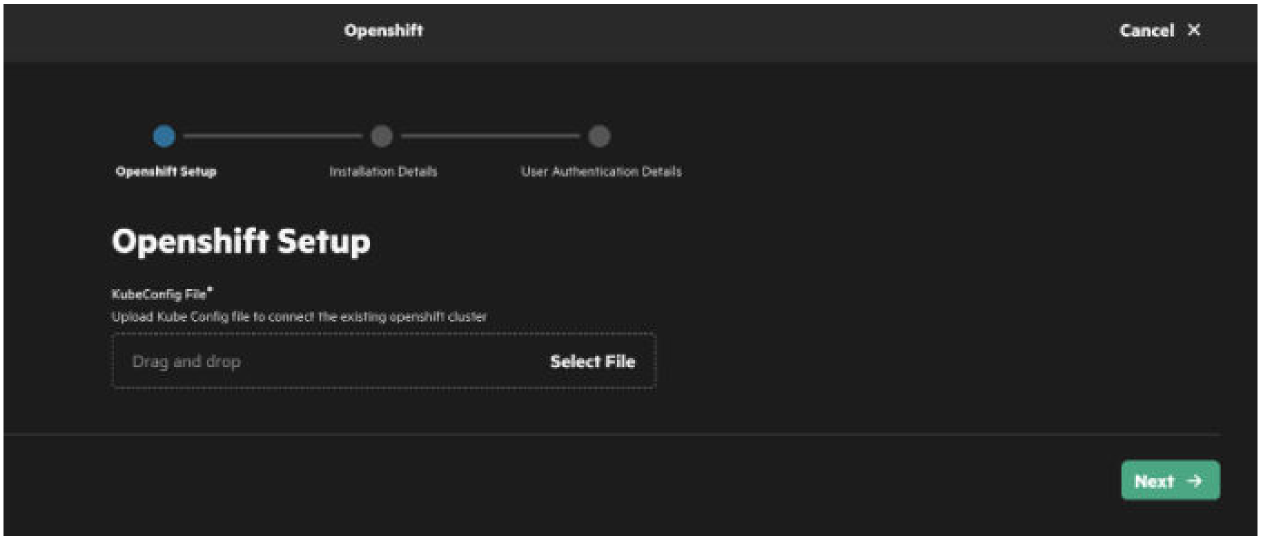


Bring your own Kubernetes

Openshift

Installs on your Openshift Kubernetes cluster.

5. On the **OpenShift Setup** screen, upload your **OpenShift Admin Kubeconfig** and then click **Next**.



Openshift Setup

KubeConfig File*

Upload Kube Config file to connect the existing openshift cluster

6. See Installing on User-Provided Hosts (Connected and Air-gapped Environments) on page 63 to continue installation, starting with Installation Details on page 67 on that page.

> **TIP:** If installation fails, you can access the Installer Web UI logs in the live container at `/root/ezua-installer-ui/log.`

Creating a DNS Mapping

Change the istio-ingressgateway service object in the istio-system namespace from NodePort to LoadBalancer. Get the external IP of the LoadBalancer and then register the external IP as a CNAME certificate in the domain or an A certificate with the alias.

To successfully create the DNS mapping, you must complete the following steps after the istio installation completes and within 45 minutes of the start of the MLDE installation.

1. To change the istio-ingressgateway service object in the istio-system namespace from NodePort to LoadBalancer, run:

   ```
   kubectl patch svc istio-ingressgateway  -p  '{"spec":
   {"type":"LoadBalancer"}}' -n istio-system
   ```

2. To get the external IP of the LoadBalancer, run:

   ```
   kubectl get svc -n istio-system
   ```

3. In your cloud domain service provider, use the external IP to create a DNS mapping using either of the following methods:

   - Register the external IP as a CNAME under the domain name.
   - Create an A certificate with the alias, as describe here for AWS and here for GCP.

   Note that the external IP is not an IPv4 address. An IPv4 address is typically used in A records.

HPE Ezmeral Unified Analytics Software Service Activation and Billing Processes

Provides post-installation steps required to activate HPE Ezmeral Unified Analytics Software in connected and air-gapped environments.

When you install HPE Ezmeral Unified Analytics Software through the installation wizard, you have the option to install in a connected environment or air-gapped environment. The activation and billing processes differ for each type of installation. In a connected environment, billing is an automated process. In an air-gapped environment, the billing process is manual and requires an activation code in addition to an activation key.

After you install and deploy HPE Ezmeral Unified Analytics Software, the system provides you with a URL to access Unified Analytics. The first time you go to the URL, the system prompts you for an activation key (and an activation code for air-gapped environments) to activate the product.

The following sections provide the information needed to get the activation key and activation code (for air-gapped environments). When you have those, you can return to the Unified Analytics URL and enter the activation key to activate Unified Analytics.

IMPORTANT: *HPE Ezmeral Unified Analytics Software services only work with a valid activation key and activation code (for air-gapped environments). Services are deactivated if the activation key and/or activation code become invalid, for example, if contractual obligations are not met.*

Service Activation and Billing in Connected Environments

Provides information for administrators about HPE Ezmeral Unified Analytics Software activation and billing in a connected environment, including activation steps.

An administrator needs the following information to activate Unified Analytics in a connected environment:

Information	Description
Platform ID	Unique, system-generated ID assigned to the HPE Ezmeral Coordinator instance during installation. The ID is displayed when you go to the Unified Analytics URL provided after installation.
Activation key	The license file that the administrator uploads to complete the installation of Unified Analytics. The administrator can download the activation key in their MY HPE SOFTWARE CENTER customer portal. The activation key file is a signed XML file.

Service activation and billing in connected environments is mostly automated. The only manual process that the administrator performs is going to MY HPE SOFTWARE CENTER and downloading the activation key file and then uploading the file into Unified Analytics to activate the product. The activation key is valid for the length of the contract, typically one, three, or five years unless the contract is made invalid, such as product cancellation or failure to meet the contractual agreement.

To activate Unified Analytics, an administrator completes the following steps:

1. Install and deploy Unified Analytics. For connected environments, select the Connected option during installation. The system provides the URL to access Unified Analytics.
2. Go to the Unified Analytics UI URL provided. The window displays a Platform ID and requests an activation key. You cannot proceed with activation until you provide the activation key file.
3. Copy the unique Platform ID.
4. After purchasing HPE Ezmeral Unified Analytics Software, the activation key is made available to you through the **Activate your products** button in the **HPE Subscription Electronic Receipt** email that you receive from HPE. This receipt directs you to MY HPE SOFTWARE CENTER where you can activate your product.
5. On the Activate EON page, enter the Platform ID (copied in step 3) in the Platform ID field.
6. Once activation is completed, download the Unified Analytics activation key file.
7. Return to the Unified Analytics URL and upload the activation key file.

Billing Process in Connected Environments

When the activation key is uploaded, the cluster registers with the HPE billing service. Consumption data is uploaded to the HPE billing service on an hourly basis. Consumption data is based on the vCPU used by applications every hour.

Service Activation and Billing in Air-Gapped Environments

Provides information for administrators about HPE Ezmeral Unified Analytics Software activation and billing in an air-gapped environment, including activation steps.

An administrator needs the following information to activate Unified Analytics in an air-gapped environment:

Information	Description
Platform ID	Unique, system-generated ID assigned to the HPE Ezmeral Coordinator instance during installation. The ID is displayed when you go to the Unified Analytics URL provided after installation.
Activation key	The license file that the administrator uploads to complete the installation of Unified Analytics. The administrator can download the activation key in their MY HPE SOFTWARE CENTER customer portal. The activation key file is a signed XML file.
Activation code	A unique code that HPE Ezmeral Customer Support gives to the administrator every 30 days to keep clusters in an active state. Automatically deactivated after 45 days (includes a 15-day grace period). The activation code file is a signed JSON file. See Billing Process in Air-Gapped Environments and Renewing the Activation Code on page 95.

Service activation and billing in an air-gapped environment requires an activation key file and an activation code. The activation code must be renewed on a monthly basis (every 30 days). See Billing Process in Air-Gapped Environments.

Getting the Activation Key File and Activation Code

To get the activation key:

1. Install and deploy Unified Analytics. For air-gapped deployments, select the Air-Gapped option during installation. The system provides the URL to access Unified Analytics.
2. Go to the Unified Analytics URL provided. The window displays a Platform ID and requests an activation key and activation code. You cannot proceed with the activation until you provide the activation key file and activation code.
3. Copy the unique Platform ID.
4. After purchasing HPE Ezmeral Unified Analytics Software, the activation key is made available to you through the **Activate your products** button in the **HPE Subscription Electronic Receipt** email that you receive from HPE. This receipt directs you to MY HPE SOFTWARE CENTER where you can activate your product.
5. On the Activate EON page, enter the Platform ID (copied in step 3) in the Platform ID field.
6. Once activation is completed, download the Unified Analytics activation key file.
7. Return to the Unified Analytics URL and upload the activation key file.

To get the first activation code to activate Unified Analytics:

1. To request the activation code, open a support case at https://support.hpe.com using the account you have on the HPE Support Center customer portal. The support ticket must include the following information:
   - Activation key
   - Platform ID
   - Cluster ID

   TIP: This is the same portal that you would use to create any kind of ticket related to your platform. If you do not have an account, you can create an account for free. When you create an account, you must link your support contract to the account. If you have never used the customer portal, refer to the KB article here to help you get your support portal account up and running.
2. When support notifies you that the activation code is available in your customer portal, go to the portal and get the code.
3. Return to the Unified Analytics URL and upload both the activation key and activation code files.

Billing Process in Air-Gapped Environments

Contracts for air-gapped installations must be validated with an activation code on a monthly basis.
The Unified Analytics cluster securely stores billing data. The Unified Analytics site administrator must download the billing data at the end of the billing cycle and then open an HPE Support Center customer support ticket to renew the activation code. The support ticket that the administrator opens must include the following information:

- Billing data (downloaded from the Billing tab in Unified Analytics)
- Cluster ID

HPE Support Center renews the certificate and credentials through the billing and registration system and then uploads the new activation code to your customer portal. This cycle continues on a monthly basis to keep clusters active. Failure to adhere to this process can result in cluster deactivation or service disruption. Unified Analytics provides regular updates and reminder alerts on the product screen.

Renewing the Activation Code

To get a new activation code (every 30 days), complete the following steps:

1. Sign in to Unified Analytics.
2. In the left navigation bar, select **Administration > Settings**.
3. Click the **Billing** tab.
4. On the **Billing** tab, download the billing data for the current billing cycle.
5. Open a support case at https://support.hpe.com using the account you have on the HPE Support Center customer portal and include the following information:
   - Cluster ID
   - Billing data file
6. When support updates the ticket, go to your customer portal to get the new activation code.
7. Return to Unified Analytics, and enter the activation code in the **Activation Code** field on the **Billing** tab.

   IMPORTANT: Failure to complete these steps monthly can result in access to the Unified Analytics applications and services being disabled.

Using the Air Gap Utility

Describes how to use the Air Gap Utility to download files in an air-gapped HPE Ezmeral Unified Analytics Software environment.

IMPORTANT: The README.txt file included with the product provides instructions for downloading and extracting the HPE Ezmeral Unified Analytics Software binaries that are required to install the product, including the Air Gap Utility. If you downloaded and extracted the files, as described in the README.txt file, you should have the Air Gap Utility.

Requirements

The Air Gap Utility has the following requirements:

Python	• 2.7 • 3.6 and above
Operating System	At minimum: • RHEL 8 • SLES 15 • Rocky Linux 8
Scopeo	At minimum: • For RHEL or Rocky Linux:

  - Skopeo 0.1.40
- For SLES:
  - Skopeo 0.1.41

About the Air Gap Utility

HPE Ezmeral Unified Analytics Software 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 installation.

To install the air gap utility package:

1. Install Skopeo. In the CLI, enter the following:
   - For RHEL:

     ```
     dnf install -y skopeo
     ```

   - For SLES:

     ```
     zypper install -y skopeo
     ```

2. Install the `hpeairgaputil` package:
   - PIP2:

     ```
     pip install hpeairgaputil-1.6.0-py2.py3-none-any.whl
     ```

   - PIP3:

     ```
     pip3 install hpeairgaputil-1.6.0-py2.py3-none-any.whl
     ```

   TIP: To uninstall `hpeairgaputil`, use:

   - PIP2:

     ```
     pip uninstall hpeairgaputil-1.6.0-py2.py3-none-any.whl
     ```

   - PIP3:

     ```
     pip3 uninstall hpeairgaputil-1.6.0-py2.py3-none-any.whl
     ```

Using Air Gap Utility Filters

After installing the air gap utility package, you can filter the available apps for a given HPE Ezmeral Unified Analytics Software version in a project.

You must provide one of the following mandatory arguments in each of your commands:

-
  ```
  --list_releases
  ```

-
  ```
  --release
  ```

  TIP: To display a list of options available in the `ezua-airgap-util`, use the following command:

  ```
  ezua-airgap-util --help
  ```

You can use filters to display the following information:

- **Release:** List all releases with the following command:

  ```
  ezua-airgap-util --list_releases
  ```

- **Images**: List all the images for a particular release:

  ```
  ezua-airgap-util --release <release-number>
  ```

  - List available images without headers:

    ```
    ezua-airgap-util --release <release-number> --noheaders
    ```

  - List all required images:

    ```
    ezua-airgap-util --release <release-number> --required
    ```

  - List all optional images:

    ```
    ezua-airgap-util --release <release-number> --optional
    ```

- **List components:** List all the components that are available for a particular release:

  ```
  ezua-airgap-util --list_components --release <release-number>
  ```

- **Component:** List all images for a particular component:

  ```
  ezua-airgap-util --release <release-number> --component <component>
  ```

- **Size:** Valid values include `b`, `kb`, `mb`, and `gb`.
  - Display images less than a certain size:

    ```
    ezua-airgap-util --release <release-number> --lessthan 1mb
    ```

  - Display images greater than a certain size:

    ```
    ezua-airgap-util --release <release-number> --greaterthan 5gb
    ```

  - Display images between two sizes:

    ```
    ezua-airgap-util --release <release-number> --lessthan
    6gb --greaterthan 5gb
    ```

- You can **combine filters** to provide a more customized query, for example:

  ```
  ezua-airgap-util --release <release-number> --component falco
  ```

- To filter for a specific name or string, you can use the options **–noheaders | grep <String>**:

  ```
  ezua-airgap-util --release <release-number> --noheaders | grep <String>
  ```

Downloading Air Gap Files

After Using Air Gap Utility Filters on page 96 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:

   ```
   ezua-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:

   ```
   ezua-airgap-util --release <release-number> --lessthan
   1mb --copy --dest_path images/ --force
   ezua-airgap-util --release <release-number> --lessthan
   1mb --copy --dest_path images/ --dest_compress --force
   ```

   - To copy **multiple images** to a local filesystem, run the following command. Provide the destination path where you want to store your files.

     ```
     ezua-airgap-util --release <release-number>
     <add-on_filters> --copy --dest_path <destination-path>
     ```

   - To copy a **single image** to a local filesystem, execute the following command. Provide the destination path where you want to store your files.

     ```
     ezua-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:

       ```
       ezua-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>
  ```

- To copy a **single image** to a remote container registry, execute the following command. Provide the destination URL and credentials for your container registry.

  ```
  ezua-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` argument.
- 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:

  ```
  ezua-airgap-util --release v1.5.0 | grep -i airflow
  ```

Using Skopeo `--options` with the Air Gap Utility

This section describes how to use Skopeo `--options` with the Air Gap Utility and provides usage examples.

The following examples show the Skopeo `--preserve-digests` and `--retry-times` options used with the Air Gap Utility:

```
ezua-airgap-util --release v1.5.0 --image
longhornio/livenessprobe:v2.9.0 --copy --dest_path
ezua-v1.5.0/ --options="--preserve-digests"
```

```
ezua-airgap-util --release v1.5.0 --image
longhornio/livenessprobe:v2.9.0 --copy --dest_path
ezua-v1.5.0/ --options="--retry-times 5"
```

You can use multiple Skopeo options with the Air Gap Utility.

The following example demonstrates how to use the Skopeo `--preserve-digests` and `--retry-times` options together:

```
ezua-airgap-util --release v1.5.0 --image
longhornio/livenessprobe:v2.9.0 --copy --dest_path
ezua-v1.5.0/ --options="--preserve-digests --retry-times 5"
```

Expediting the Image Download Process

HPE recommends copying the images to the remote container registry using the following command:

```
ezua-airgap-util --release <release-number>
<add-on-filters> --copy --dest_url <destination-url> --dest_creds
<username:password>
```

However, if this process is too time-intensive, for example if internet access is slow and downloading 250-300 GB of images takes more than a day to download at the container registry location, use this expedited method to download images.

To expedite the image download process, download the images to disk on a host at a location with a high-speed internet connection, transfer the downloaded images to an internal container registry location, and then upload the images to your private image registry.

To expedite the image download process, complete the following steps:

1. On a server with a high-speed connection, download the images to a package.

To download some or all of the images into a local directory, run the following `ezua-airgap-util` command:

```
ezua-airgap-util --release
v1.5.0 --copy --dest_path
<local-directory-path> --force

//Example: hpe-airgap-util --release
v1.5.0 --image
busybox:latest --copy --dest_path
images/ --force
```

This command downloads each image into an individual TAR file.

2. Transfer the package (TAR files) to a host that runs the container registry.

Transfer all of the TAR files to the server where you will run the Air Gap Utility to upload the images to the private image registry.

3. Use the Air Gap Utility to upload the package (TAR files) to the private image registry.

To upload the image files to the private image registry, run the following `ezua-airgap-util` command:

```
ezua-airgap-util --release
v1.5.0 --copy_from_dir
<image-dir-path> --dest_url
<destination-url> --dest_creds
<username:password>

//Example: ezua-airgap-util --release
v1.5.0 --copy_from_dir /home/mehul/
images --dest_creds
username:password  --dest_url
lr1-bd-harbor-registry.mip.storage.hpe
corp.net/mehul-test
```

Ports Used by HPE Ezmeral Unified Analytics Software

Lists and describes the ports used by HPE Ezmeral Unified Analytics Software.

Kubernetes Ports

TCP Ports	Purpose
80, 443	Ingress traffic into the cluster
6443	kube apiserver
2379-2380	etcd
10250	kubelet
10248	kubelet (healthz endpoint)
10249	kube-proxy (metrics)
10256	kube-proxy (health check)
10259	kube-scheduler
10257	kube-controller-manager
9099	calico-node
9100	Node exporter service
30000-32767	NodePort Services

UDP Ports

Ports	Purpose
8472	Flannel vxlan traffic between all Kubernetes hosts

Installer Host Ports

TIP: The installer automatically opens ports if the firewall is disabled.

TCP Ports	Purpose
8080	Installer UI
22	SSH connectivity between installer, control plane, and worker hosts for software installation

Upgrading HPE Ezmeral Unified Analytics Software

Provides the steps to upgrade HPE Ezmeral Unified Analytics Software.

Unified Analytics administrators can upgrade HPE Ezmeral Unified Analytics Software to the latest version of the product.

REMEMBER:

- To upgrade HPE Ezmeral Unified Analytics Software to version 1.5.0, contact HPE Support.
- To upgrade HPE Ezmeral Unified Analytics Software from version 1.5.0 to 1.5.2, follow the instructions outlined in this topic.

The upgrade process typically lasts up to 90 minutes. During the upgrade, the infrastructure services are upgraded first, followed by applications upgrade. You can monitor the upgrade as it progresses. While the upgrade is in progress, you cannot access the HPE Ezmeral Unified Analytics Software UI; however, the UI automatically loads and becomes accessible when the upgrade completes.

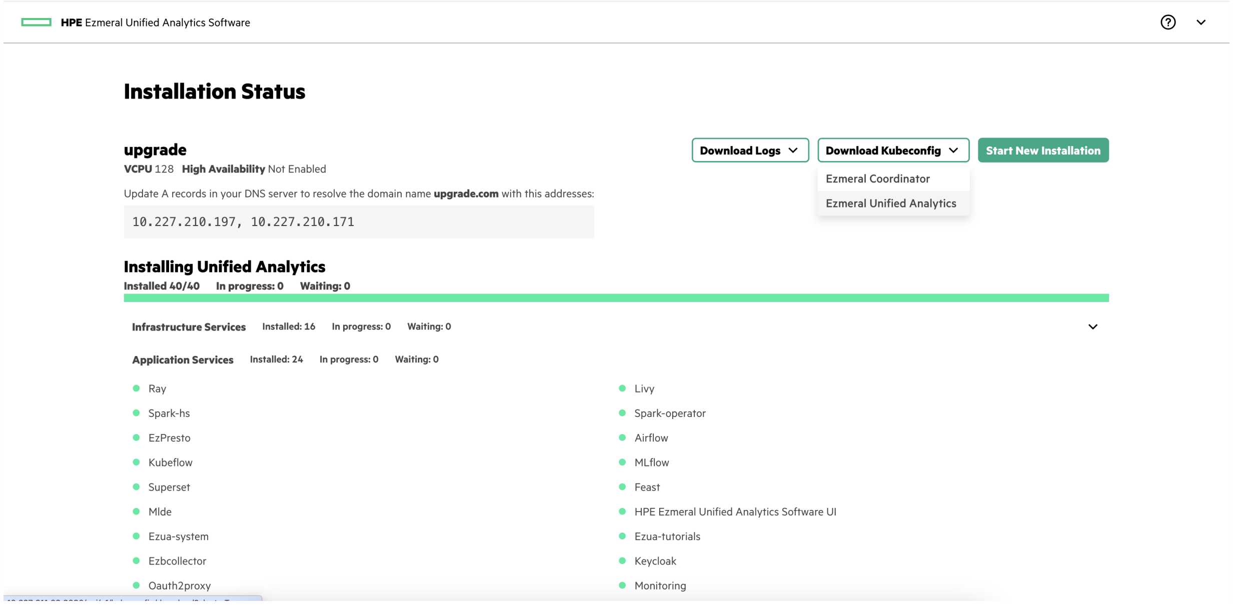
Complete the following steps to upgrade HPE Ezmeral Unified Analytics Software:

- Complete the prerequisites.
- Upgrade the HPE Ezmeral Coordinator cluster.
- Verify that the upgrade completed successfully for the HPE Ezmeral Coordinator cluster.
- Download the upgrade bundle for workload cluster.
- Trigger the upgrade for the workload cluster from the UI.
- Monitor the progress of the upgrade for the workload cluster.
- Verify that the upgrade completed successfully for the workload cluster.

Complete Prerequisites

Before upgrading HPE Ezmeral Unified Analytics Software to the new version, ensure the following prerequisites are met:

- You have administrative access to the cluster running the current version of HPE Ezmeral Unified Analytics Software.
- Download the HPE Ezmeral Coordinator upgrade package (`ezua-upgrade-package-to-<version-number>-commit-chksum.tgz`) from MY HPE SOFTWARE CENTER. If you cannot access this package, contact HPE Support.
- To upgrade, you need the `kubeconfig` files for the HPE Ezmeral Coordinator and workload clusters. You can get the `kubeconfig` files through the **Download Kubeconfig** button on the last UI installation screen from the previous installation.



Installation Status

upgrade

VCPU 128 **High Availability** Not Enabled

Update A records in your DNS server to resolve the domain name **upgrade.com** with this addresses:

```
10.227.210.197, 10.227.210.171
```

Installing Unified Analytics

Installed 40/40 In progress: 0 Waiting: 0

Infrastructure Services Installed: 16 In progress: 0 Waiting: 0

Application Services Installed: 24 In progress: 0 Waiting: 0

- Ray
- Spark-hs
- EzPresto
- Kubeflow
- Superset
- Mlde
- Ezua-system
- Ezbcollector
- Oauth2proxy
- Livy
- Spark-operator
- Airflow
- MLflow
- Feast
- HPE Ezmeral Unified Analytics Software UI
- Ezua-tutorials
- Keycloak
- Monitoring

Alternatively, you can SSH into the HPE Ezmeral Coordinator and workload nodes to get the `kubeconfig` files.

NOTE: If you need to identify the HPE Ezmeral Coordinator nodes, run the following command:

```
kubectl get nodes --kubeconfig <ezmeral-coordinator-kubconfig-file>
```

- Verify that pods are not in the *Pending* state. To check if the pods are in the *Pending* state, run the following command on the workload cluster:

  ```
  kubectl get pod -A | grep Pending
  ```

- Verify that the workload cluster has CPU and memory resources available, with approximately 20 free CPU units in the cluster. You can check the CPU and memory resources by running the following commands on the workload cluster:

Actions	Commands
Check the CPU and memory resources on a particular worker node.	`kubectl describe node ${NodeName}`
Check the CPU resources on all worker nodes.	`kubectl describe node -l node-role.kubernetes.io/worker | grep cpu | grep -v cpu-`
Check the CPU resources on a particular worker node.	`kubectl describe node ${NodeName} | grep cpu | grep -v cpu-`
Check the memory resources on all worker nodes.	`kubectl describe node -l node-role.kubernetes.io/worker | grep memory | grep -v MemoryPressure`
Check the memory resources on a particular worker node.	`kubectl describe node ${NodeName} | grep memory | grep -v MemoryPressure`

- Verify that the existing ezaddon resources in the Unified Analytics cluster are in the *Installed* state. To check the status of all the ezaddon resources, run the following command on the workload cluster:

  ```
  kubectl get ezad -A
  ```

NOTE:

If the verification for prerequisites fails, contact HPE support.

Upgrade HPE Ezmeral Coordinator Cluster

1. Use `scp` to manually copy the downloaded upgrade package (`ezua-upgrade-package-to-<version-number>-commit-chksum.tgz`) to the HPE Ezmeral Coordinator cluster control plane node.

2. Extract the upgrade package. For example:

```
[ouxi@m2-lr1-dev-vm210015 ~]$ tar xvf
ezua-upgrade-package-to-v1.5.2-364aa1349-90e173.tgz
upgrade/
upgrade/IMAGES
upgrade/ezfab-release.tgz
upgrade/ezfabricctl_darwin_amd64
upgrade/ezfabricctl_linux_amd64
upgrade/ezua-system-1.5.2.tgz
upgrade/ezfab-upgrade.sh
upgrade/VERSION
[ouxi@m2-lr1-dev-vm210015 ~]$ ls -al
total 278660
drwx------. 4 ouxi ldap       182 Sep 26 22:40 .
drwxr-xr-x. 4 root root        30 Sep 26 16:19 ..
-rw-------. 1 ouxi ldap        18 Sep 26 16:10 .bash_logout
-rw-------. 1 ouxi ldap       141 Sep 26 16:10 .bash_profile
-rw-------. 1 ouxi ldap       376 Sep 26 16:10 .bashrc
-rw-------. 1 ouxi ldap 285124183 Sep 26 22:40
ezua-upgrade-package-to-v1.5.2-364aa1349-90e173.tgz
drwx------. 3 ouxi ldap        33 Sep 26 16:21 .kube
-rw-------. 1 ouxi ldap    207894 Sep 26 16:21 mgmt-bootstrap.log.xtrace
drwx------. 2 ouxi ldap       176 Sep 25 18:34 upgrade
```

3. The upgrade directory includes the `ezfab-upgrade.sh` script, which is used to upgrade HPE Ezmeral Coordinator. To run the `ezfab-upgrade.sh` script, prepare the kubeconfig file at `$HOME/.kube/ezfab.config`. For example:

```
[ouxi@m2-lr1-dev-vm210015 ~]$ cd .kube/
[ouxi@m2-lr1-dev-vm210015 .kube]$ ls
cache  config
[ouxi@m2-lr1-dev-vm210015 .kube]$
[ouxi@m2-lr1-dev-vm210015 .kube]$
[ouxi@m2-lr1-dev-vm210015 .kube]$ cp config ezfab.config
[ouxi@m2-lr1-dev-vm210015 .kube]$
[ouxi@m2-lr1-dev-vm210015 .kube]$
[ouxi@m2-lr1-dev-vm210015 .kube]$ ls -al
total 16
drwx------. 3 ouxi ldap   53 Sep 26 22:40 .
drwx------. 4 ouxi ldap  182 Sep 26 22:40 ..
drwx------. 4 ouxi ldap   35 Sep 26 16:21 cache
-rw-------. 1 ouxi ldap 5671 Sep 26 16:21 config
-rw-------. 1 ouxi ldap 5671 Sep 26 22:40 ezfab.config
```

4. Navigate to the upgrade directory and manually run the `ezfab-upgrade.sh` script to trigger the HPE Ezmeral Coordinator upgrade. For example:

```
[ouxi@m2-lr1-dev-vm210015 .kube]$ cd
[ouxi@m2-lr1-dev-vm210015 ~]$ ls
ezua-upgrade-package-to-v1.5.2-364aa1349-90e173.tgz
mgmt-bootstrap.log.xtrace  upgrade
[ouxi@m2-lr1-dev-vm210015 ~]$ cd upgrade/
[ouxi@m2-lr1-dev-vm210015 upgrade]$ ls
ezfab-release.tgz  ezfabricctl_darwin_amd64  ezfabricctl_linux_amd64
ezfab-upgrade.sh  ezua-system-1.5.2.tgz  IMAGES  VERSION
[ouxi@m2-lr1-dev-vm210015 upgrade]$
[ouxi@m2-lr1-dev-vm210015 upgrade]$ ./ezfab-upgrade.sh
Check OS ...

Running ezfabricctl orchestrator upgrade...

No resources found
INFO Upgrading Ezmeral coordinator cluster/components
INFO ezfabricctl version: 1.3.0-1a6b8d0
INFO Build date: 2024-09-25T23:56:04Z
INFO Release Package: ezfab-release.tgz
INFO Deploy Target:
INFO Deploy Env:
INFO Running precheck
INFO Upgrading Ezmeral coordinator cluster
...
INFO upgrade addons: [longhorn]
INFO Started monitoring the status of service addons (EzkfUpgrade)
INFO Service Addons (EzkfUpgrade) are not upgraded yet. Waiting...
INFO Service Addons (EzkfUpgrade) are not upgraded yet. Waiting...
INFO Service Addons (EzkfUpgrade) are not upgraded yet. Waiting...
INFO Service Addons (EzkfUpgrade) are not upgraded yet. Waiting...
INFO Service Addons (EzkfUpgrade) are not upgraded yet. Waiting...
INFO Upgraded service addons in Ezmeral coordinator cluster
INFO Installing EzUA plugin CR: ezua-plugin-1.5.2-75f3ef1.yaml
INFO Successfully upgraded Ezmeral coordinator cluster/components
```

The output of the `ezfab-upgrade.sh` script displays the upgrade progress. A successful upgrade is indicated by the message: *"INFO Successfully upgraded Ezmeral coordinator cluster/components"*. Some pods may not enter the **Running** state immediately after the script finishes.

5. Wait and monitor all pods to verify they are all in the **Running** state. For example:

```
[ouxi@m2-lr1-dev-vm210015 upgrade]$ kubectl get pod -A
NAMESPACE
NAME                                                     READY
STATUS              RESTARTS       AGE
capi-system
capi-controller-manager-69b84d5d98-knns6                 1/1
Running             5 (22m ago)    6h59m
cert-manager
cert-manager-5b99f586f8-rnw6t                            1/1
Running             0              37m
cert-manager
cert-manager-cainjector-b8cdfd94f-cqjnw                  1/1
Running             0              37m
cert-manager
cert-manager-webhook-58c8649c5c-vnm2n                    1/1
Running             0              37m
ezaddon-system
ezaddon-controller-manager-7865c6f86b-ktcxx              1/1
Running             1 (22m ago)    39m
ezkf-mgmt
op-v2-upgrade-addonproc-ezkf-mgmt-f42q7                  1/1
Running             0              3m13s
ezkf-system
ezfabric-controller-manager-6b95849759-nq67n             2/2
Running             3 (22m ago)    35m
ezkf-system
ezkf-bootstrap-controller-manager-687875bc97-tlt7k       2/2
Running             3 (22m ago)    6h59m
ezkf-system
ezkf-control-plane-controller-manager-7bd7f9f876-85bl7   2/2
Running             4 (22m ago)    6h58m
ezkf-system
ezkf-infrastructure-manager-57b6675545-82kc8             2/2
Running             4 (24m ago)    6h58m
ezua-demo
ezua-demo-controller-manager-76d8dc858f-nbmhh            2/2
Running             2 (5h4m ago)   6h28m
ezua-demo
op-clustercreate-ezua-demo                               1/1
Running             0              6h29m
ezua-demo
op-plugin-upgrade-ezua-demo                              0/1
ContainerCreating   0              27s
ezua-demo
w-op-workload-deploy-ezua-demo                           1/1
Running             0              6h28m
kube-system
calico-kube-controllers-5696576df5-s99dl                 1/1
Running             0              7h3m
kube-system
calico-master-dgq5n                                      1/1
Running             0              7h3m
kube-system
calico-typha-66c46875c4-5xktv                            1/1
Running             0              7h3m
kube-system
coredns-d4544c989-5vlbq                                  1/1
Running             0              7h2m
kube-system
coredns-d4544c989-hr2zj                                  1/1
Running             0              7h2m
```

```
longhorn-system
csi-attacher-67578d8dc7-q9pqd                            1/1
Running             0              19m
longhorn-system
csi-provisioner-59fd5dfcff-bwvgl                         1/1
Running             0              18m
longhorn-system
csi-resizer-8686444cc9-66c4v                             1/1
Running             0              17m
longhorn-system
csi-snapshotter-c49d97966-99txb                          1/1
Running             1 (15m ago)    16m
longhorn-system
engine-image-ei-6435403c-sl8x6                           1/1
Running             0              27m
longhorn-system
instance-manager-0ee8f243f69bf3a6841efc3b74cdcf05        1/1
Running             0              23m
longhorn-system
longhorn-csi-plugin-lsm7v                                3/3
Running             0              15m
longhorn-system
longhorn-driver-deployer-56b8686647-v9nts                1/1
Running             0              28m
longhorn-system
longhorn-manager-95r5t                                   1/1
Running             0              27m
longhorn-system
longhorn-ui-7dbff4bb69-wwbkn                             1/1
Running             0              28m
mgmt-cluster-ops-system
mgmt-cluster-ops-controller-manager-7596655d45-kfqml     2/2
Running             3 (22m ago)    33m
[ouxi@m2-lr1-dev-vm210015 upgrade]$
```

6. Verify that all ezaddons are in the **installed** state. For example:

```

[ouxi@m2-lr1-dev-vm210015 ~]$ kubectl get ezad -A
NAMESPACE        NAME                      CLUSTER      INSTALL
STATE       PHASE   REASON   PROGRESS
ezaddon-system   cert-manager              in-cluster   true
installed
ezaddon-system   ezkf                      in-cluster   true
installed
ezaddon-system   in-cluster-csi-detector   in-cluster   true
installed
ezaddon-system   longhorn                  in-cluster   true
installed
[ouxi@m2-lr1-dev-vm210015 ~]$
```

Verify HPE Ezmeral Coordinator Cluster Upgrade Success

- Review the logs stored in the `./log` directory to ensure there are no errors or warnings.
- Verify that there are no errors or warnings in the console output during the upgrade process.
- Verify that the `ezkf-agent` status is in the **active (running)** state. To verify, run:

```
systemctl status ezkf-agent
```

- Verify that the Kubernetes node status is in the **Ready** state. To verify, run:

  ```
  kubectl get nodes
  ```

If the upgrade fails, contact HPE Support.

Download Workload Cluster Upgrade Bundle

You can either automatically or manually download the upgrade bundle required to upgrade Unified Analytics workload cluster. In a connected environment, you can enable an automatic download or perform a manual download. In an air-gapped environment, you must manually download the bundle.

Manual Download

(Air-gapped environment only) Before you complete the following steps, use the air-gap utility to download all the images required for the Unified Analytics version you want to upgrade to and push those images into the air-gap registry. To learn about how to use the air-gap utility, see Using the Air Gap Utility on page 95.

To download the upgrade bundle for both the air-gapped and connected environment, complete the following steps:

1. In the left navigation bar, click **Administration Settings**.
2. Click **Updates** **Download Updates**.
3. Enter **Image Name**.

   To find the the upgrade bundle images, see Upgrade Bundle Images on page 206.

   An example of the image name for:

   - An air-gapped environment: `<RegistryURL>/ezua/ezua-system-bundle:<image-tag>`
   - The connected environment: `marketplace.us1.greenlake-hpe.com/ezua/ezua/ezua-system-bundle:<image-tag>`.
4. Click **Download**.
5. View the downloaded image in the table.

   NOTE: To remove an image, click **Delete**.
6. After successfully downloading a new image, wait for a few minutes (at most five minutes) to see the upgrade bundle on the **Available Updates** table as **EZUA System**.
7. Once you see the **EZUA System** upgrade bundle, continue to the Trigger Workload Cluster Upgrade from UI on page 109 section.

Automatic Download (Using EzCentral)

In a connected environment, you can upgrade Unified Analytics by enabling the automatic downloads of the upgrade bundle. After the upgrade bundle is downloaded, you can view it on the **Available Updates** table.

To automatically download the upgrade bundle, complete the following steps:

1. In the left navigation bar, click **Administration Settings**.
2. Click **Updates** **Download Updates**.
3. To enable the automatic download of the upgrade bundle, toggle **Enable**.

   NOTE: To disable the automatic download of the upgrade bundle, toggle **Disable**. When you disable the automatic download of the upgrade bundle in the connected environment, you must manually download the upgrade bundle to upgrade to the latest version of the product.
4. After enabling the automatic downloads, when an upgrade is available for a new version, HPE Ezmeral Unified Analytics Software displays the upgrade bundle on the **Available Updates** table as **EZUA System**.
5. Once you see the **EZUA System** upgrade bundle, continue to the Trigger Workload Cluster Upgrade from UI on page 109 section.

Trigger Workload Cluster Upgrade from UI

Once you see the **EZUA System** upgrade bundle, you can click the bundle to view details. In the **Details** dialog box, you can see the name, description, the current version of the Unified Analytics, and the new available version for Unified Analytics. Once you see the new available versions for the upgrade, you can perform the following actions by clicking on the **Actions** menu.

Update Now

To upgrade Unified Analytics immediately, complete the following steps:

1. Click **Update Now** in the **Actions** menu. This will open an **Update Now** dialog box and you can compare the current and new available versions of Unified Analytics for upgrade.
2. Click **Update Now** to immediately start the Unified Analytics upgrade. Wait for the upgrade to be in the **In Progress** status.

   NOTE: You cannot cancel the upgrade once it is in the **In Progress** status.
3. To monitor the upgrade process, see Monitor Workload Cluster Upgrade Progress on page 110.

Schedule Update

To schedule the upgrade for later, complete the following steps:

1. Click **Schedule Update** in the Actions menu. This will open a **Schedule Update** dialog box and you can compare the current and new available versions of Unified Analytics for upgrade.

2. Select a date and time to schedule the update. Wait for the upgrade to be in the **Scheduled** status.

3. Once an upgrade is in the **Scheduled** status, you can perform the following actions from the **Actions** menu.

   Cancel

   You can cancel the scheduled upgrade any time before upgrade starts or if upgrade is not in the **In Progress** status yet.

   Reschedule

   You can reschedule the scheduled upgrade any time before upgrade starts or if upgrade is not in the **In Progress** status yet.

   Update

   You can upgrade Unified Analytics immediately even though it has been scheduled for a later date and time.

4. To monitor the upgrade process, see Monitor Workload Cluster Upgrade Progress on page 110.

Monitor Workload Cluster Upgrade Progress

You can use the workload cluster to monitor the upgrade as it progresses. You can also use the HPE Ezmeral Coordinator cluster to debug errors and collect log files.

On Workload Cluster

To monitor the upgrade progress, run the following commands on the Unified Analytics workload cluster:

Actions	Commands	Descriptions
View the upgrade CR.	`kubectl get EzkfWorkloadUpgrade -A -o yaml`	• Retrieves the `EzkfWorkloadUpgrade` custom resource (CR) in YAML format. • Allows you to view detailed information about the `EzkfWorkloadUpgrade` CR, including its configuration and current state. • Enables you to troubleshoot and debug a failure reason.

Actions	Commands	Descriptions
List the ezaddon resources.	`kubectl get eza -A`	• Retrieves a list of all the ezaddon resources from all namespaces. • Allows you to view the installed version of each ezaddon resource.
Monitor the current status of the ezaddon resources.	`kubectl get ezad -A`	• Retrieves a list of all the ezaddon resources from all namespaces. • Allows you to view the current status of each ezaddon resource, such as installed, upgrading, failed, and others.
Monitor the status of the ezaddon resources in real-time.	`kubectl get ezad -A -w`	• Retrieves a list of all the ezaddon resources from all namespaces and continuously watches for any changes or updates to the ezaddon resources. • Allows you to monitor the status and changes of the ezaddon resources in real-time.
Monitor the bootstrap pod.	`kubectl get pod -n ezaddon-system`	• Monitor the bootstrap pod in the `ezaddon-system` namespace. • Allows you to track the progress of the upgrade and identify any potential issues or failures early on.

Actions	Commands	Descriptions
Monitor the total time for the upgrade process.	`kubectl get EzkfWorkload Upgrade -A -o yaml | grep Time`	• Extracts the time required to complete the upgrade process from the YAML output.

On HPE Ezmeral Coordinator Cluster

To debug and collect log files, run the following commands on the HPE Ezmeral Coordinator cluster:

1. To get a list of pods in the HPE Ezmeral Coordinator cluster, run:

   ```
   kubectl get pod -A
   ```

2. In the list of pods, locate the upgrade pod named `upgrade-ezua-upgrader-<a-b-c-wxyz-namespace>`.

3. To get log files, run:

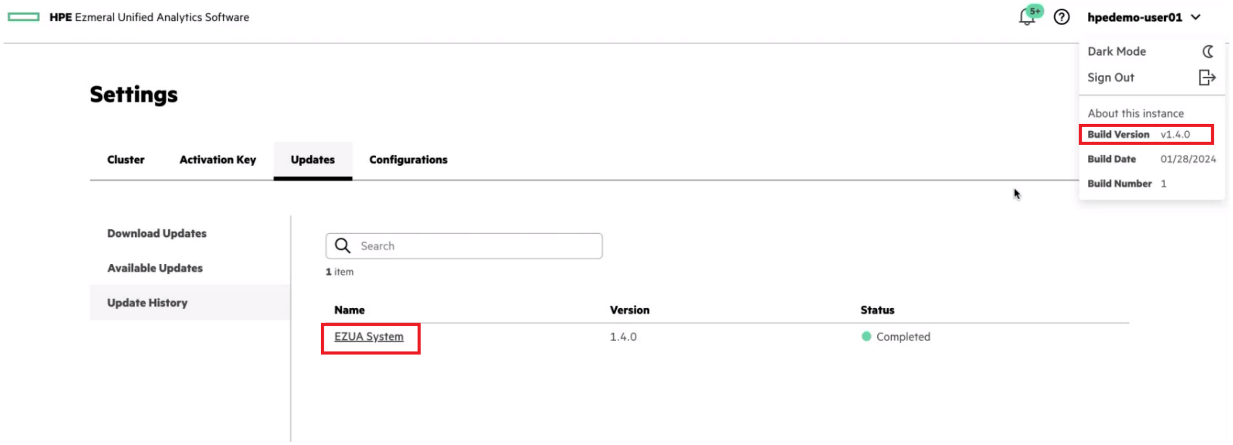
   ```
   kubectl logs -f
   upgrade-ezua-upgrader-<a-b-c-wxyz-namespace> -n <namespace>
   ```

Verify Workload Cluster Upgrade Success

If the upgrade is successful, the HPE Ezmeral Unified Analytics Software UI loads automatically. If it does not load automatically, manually reload and sign in.

To verify the sucessful upgrade:

- Click the **<username>** dropdown at the top right. You will see that the **Build Version** for the Unified Analytics cluster has been updated to the new version.



Build Version v1.4.0
Build Date 01/28/2024
Build Number 1

Name	Version	Status
EZUA System	1.4.0	● Completed

- View the upgrade details in the **Update History** table. You can click **EZUA System** to view the upgrade details. In the **Details** dialog box, you can see the name, description, and versions of Unified Analytics.

NOTE:

If you are unable to access the HPE Ezmeral Unified Analytics Software UI after 90 minutes, monitor the upgrade progress and check the logs from the HPE Ezmeral Coordinator cluster. For details, see Monitor Workload Cluster Upgrade Progress on page 110.

If the upgrade fails, contact HPE Support.

Expanding the Cluster

Describes how to add additional user-provided hosts to the management cluster to increase resource capacity and how to expand the cluster to include the additional user-provided hosts.

Expand the cluster when applications cannot run due to resource limitations, such as lack of vCPU.

When applications do not have enough resources to run, the system raises an alarm to alert you of the issue. In such cases, the HPE Ezmeral Unified Analytics Software administrator and system administrator can work together to add additional user-provided hosts to the pool of machines in the management cluster (control plane nodes) and workload cluster to increase the processing capacity of the cluster.

The following steps outline the cluster expansion process:

1. An application triggers an alert to users that it does not have sufficient resources to run.
2. Users contact the system administrator to request additional resources (add additional user-provided hosts to the management cluster).
3. A system administrator adds user-provided hosts to the cluster, as described in the section Adding User-Provided Hosts to the Cluster on page 113.
4. After the system administrator adds user-provided hosts to the cluster, the HPE Ezmeral Unified Analytics Software administrator signs into the HPE Ezmeral Unified Analytics Software UI and expands the cluster, as described in the section Expanding the Cluster on page 115.

Adding User-Provided Hosts to the Cluster

Use the `ezfab-addhost.sh` script to add control plane hosts and workload hosts to the `ezfabric-host-pool`. After you add hosts, you can expand the cluster, as described in the following section, Expanding the Cluster on page 115.

You can only add user-provided hosts to the cluster. User-provided hosts are machines that meet the installation prerequisites, as described in Installation Prerequisites on page 73.

TIP:

- If you want to use the high-availabilty (HA) feature when you expand the cluster, note that HA requires three master nodes. You must add two hosts to the `ezfabric-host-pool` with the `controlplane` role.
- If you want to increase the VCPU or VGPU resources when you expand the cluster, you must add worker hosts or GPU hosts with enough resources (VCPU or VGPU) to `ezfabric-host-pool` with the `worker` role.

To add user-provided hosts to the `ezfabric-host-pool`, complete the following steps:

1. From a CLI, sign in to the HPE Ezmeral Coordinator host.

2. Download the `ezfab-addhost-tool-1-4-x.tgz` file at https://github.com/HPEEzmeral/troubleshooting/releases/download/v1.4.0/ezfab-addhost-tool-1-4-x.tgz.

   Use one of the following commands to download the file:

   ```
   curl -L -O https://github.com/HPEEzmeral/troubleshooting/releases/
   download/v1.4.0/ezfab-addhost-tool-1-4-x.tgz
   ```

   ```
   wget https://github.com/HPEEzmeral/troubleshooting/releases/download/
   v1.4.0/ezfab-addhost-tool-1-4-x.tgz
   ```

3. Untar the `ezfab-addhost-tool-1-4-x.tgz` file:

   ```
   tar -xzvf ezfab-addhost-tool-1-4-x.tgz
   ```

4. Go to the `ezfab-addhost-tool` directory and view its contents:

   ```
   cd ezfab-addhost-tool

   ls -al
   ```

   The command returns results similar to the following:

   ```
   total 50504
   drwxr-xr-x. 2  501 games      149 Feb  2 09:57 .
   dr-xr-x---. 9 root root      4096 Feb  2 16:19 ..
   -rw-r--r--. 1  501 games     1211 Jan 26 18:16
   controlplane_input_template.yaml
   -rwxr-xr-x. 1  501 games     2687 Feb 22 10:54 ezfab-addhost.sh
   -rwxr-xr-x. 1  501 games 51695616 Jan 26 14:05 ezfabricctl
   -rw-r--r--. 1  501 games      360 Jan 26 18:24 input_example.yaml
   -rw-r--r--. 1  501 games     1205 Jan 26 18:17
   worker_input_template.yaml
   ```

   > **TIP:** You should see the `ezfab-addhost.sh` listed, as well as three YAML files (`controlplane_input_template.yaml`, `worker_input_template.yaml`, and `input_example.yaml`) that you can use as guides. Use the `cat` command to view the YAML files, for example:
   >
   > ```
   > cat controlplane_input_template.yaml
   > ```

5. Using the provided YAML files as a guide, create a YAML file.

6. Run the `ezfab-addhost.sh` script:

   ```
   ./ezfab-addhost.sh
   ```

   When you run the script, the system returns the supported options:

   ```
   Check OS ...
   Parse options ...
   Please provide the input yaml file that includes the hosts info

   USAGE: ./ezfab-addhost.sh <options>

   Options:
         -i/--input: the input yaml file that includes the hosts info.
         -k/--kubeconfig: the coordinator's kubeconfig file(optional).
   ```

7. Run the `ezfab-addhost.sh` script with the `-i` and `-k` options, as shown:

   ```
   ./ezfab-addhost.sh -i  <your-input-file>.yaml -k ~/.kube/config
   ```

8. After the `ezfab-addhost.sh` script successfully completes, run the following command to check the new hosts in `ezfabric-host-pool`:

   ```
   kubectl get ezph -A
   ```

   TIP:

   - If the `ezfab-addhost.sh` script fails, check the logs in the log directory.
   - If the failure is due to the wrong username/password or some transient error, run the following command to delete the hosts in the error state and then retry:

     ```
     ./ezfabricctl poolhost destroy --input
     $INPUT_YAML_FILE --kubeconfig $KUBECONFIG_FILE
     ```

     Note that the `INPUT_YAML_FILE` is different from the YAML file in step 7, as it only includes the failed host. After the failed hosts have been deleted, modify the `<your-input-file>.yaml` from step 7 and then complete step again 7 to re-add the failed hosts.

9. Go to the Expanding the Cluster on page 115 section (below) and follow the steps to trigger the cluster expansion from the HPE Ezmeral Unified Analytics Software UI.

Expanding the Cluster

In a user-provided host configuration, the hosts within the pool (namespace) must have enough vCPUs and vGPUs for the cluster expansion to succeed. If you request more vCPUs and vGPUs than are available, the cluster expansion will fail.

ATTENTION: If repeated attempts to expand the cluster fail with an "*already complete*" message, delete any existing EzkfOpsExpand custom resources on the workload cluster before you expand the cluster.

To identify the EzkfOpsExpand custom resources, run the following command:

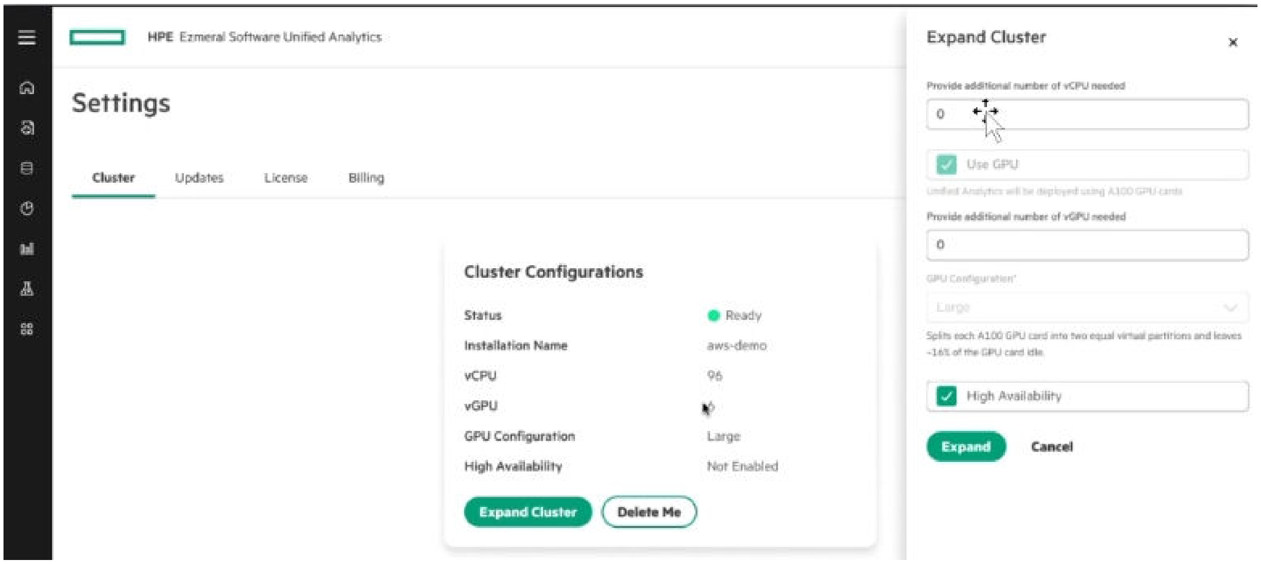
```
kubectl get ezkfopsexpand -A
# (lists the Expand CR names and namespaces)
```

For each of the EzkfOpsExpand custom resources listed in the output, run the following command:

```
kubectl delete ezkfopsexpand -n <expand_CR_namespace> <expand_CR_name>
```

To expand the cluster, complete the following steps:

1. In the left navigation bar, select **Administration > Settings**.
2. On the **Cluster** tab, select **Expand Cluster**.



Cluster Configurations

Status	● Ready
Installation Name	aws-demo
vCPU	96
vGPU	
GPU Configuration	Large
High Availability	Not Enabled

Expand Cluster

Provide additional number of vCPU needed: 0

☑ Use GPU

Unified Analytics will be deployed using A100 GPU cards

Provide additional number of vGPU needed: 0

GPU Configuration*: Large

Splits each A100 GPU card into two equal virtual partitions and leaves ~16% of the GPU card idle

☑ High Availability

3. In the **Expand Cluster** drawer that opens, enter the following information:
   a. Number of additional vCPU to allocate. For example, if the current vCPU is 96 and you add 4 vCPU, the vCPU increases to a total of 100 vCPU.
   b. Select **Use GPU** if you want to use GPU and it is not already selected. If **Use GPU** was selected during installation of HPE Ezmeral Unified Analytics Software, this option cannot be disabled and stays selected by default.
   c. Indicate the additional number of vGPU to allocate.
   d. For GPU configuration, if a size was selected during HPE Ezmeral Unified Analytics Software installation, you cannot change the size. However, if no vGPU size was selected during installation, you can select a size now. For additional information, see GPU Support on page 137.
   e. If HA was selected during HPE Ezmeral Unified Analytics Software installation, you cannot disable it. If it was not selected during installation, you can select it now. Currently HA is available for the workload cluster only. You cannot set HA for the management cluster.
   f. Click **Expand**.

Configuring HPE MLDE for Added GPU Nodes

If you add GPU nodes to the cluster after installing HPE MLDE, you must perform additional steps to ensure HPE MLDE works on these nodes. For details, see Configuring HPE MLDE for Added GPU Nodes on page 415.

Importing Frameworks and Managing the Application Lifecycle

Describes how to import, manage, and secure tools and frameworks in HPE Ezmeral Unified Analytics Software.

The Administrator can import, run, and manage customized Kubernetes applications and frameworks in HPE Ezmeral Unified Analytics Software. The Administrator can manage imported applications as well as the applications that were included with HPE Ezmeral Unified Analytics Software at the time of installation.

Imported and included applications appear in the **Tools & Frameworks** screen in HPE Ezmeral Unified Analytics Software. You can access **Tools & Frameworks** in the left navigation bar. A tile is displayed for each application. A yellow **Imported** label on a tile indicates that the application was imported.

Importing Custom Kubernetes Applications

You can import your custom Kubernetes applications through the Unified Analytics UI or an API.

To import a Kubernetes application, you upload a Helm chart with a `tar.gz` file extension and specify configuration parameters. After you import your Kubernetes applications, you can also manage them in HPE Ezmeral Unified Analytics Software. Unified Analytics supports SSO for imported applications.

- To import applications through the Unified Analytics UI, see Importing Frameworks on page 117.
- To import applications using the API, see https://github.com/HPEEzmeral/byoa-tutorials.

Managing Tools and Frameworks

HPE Ezmeral Unified Analytics Software provides the following options to manage tools and frameworks:

- Configure
- Delete
- Update for imported applications (Managing Imported Tools and Frameworks on page 122)
- Automatic and manual upgrade for included applications (Upgrading Included Frameworks on page 125

For detailed instructions, see the following:

- Managing Imported Tools and Frameworks on page 122
- Configuring Included Frameworks on page 124
- Upgrading Included Frameworks on page 125

Importing Frameworks

Describes how to import frameworks in HPE Ezmeral Unified Analytics Software.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software as Administrator.

- Configure Istio Virtual Service to expose the endpoint.

  Virtual Service Example

  ```
  apiVersion: networking.istio.io/v1alpha3
  kind: VirtualService
  metadata:
    name: {{ include "test-app.fullname" . }}
    namespace: {{ .Release.Namespace }}
    labels:
      {{- include "test-app.labels" . | nindent 4 }}
  spec:
    gateways:
      - {{ .Values.ezua.virtualService.istioGateway }}
    hosts:
      - {{ .Values.ezua.virtualService.endpoint }}
    #The following virtualService options are specific and depend on the
  application implementation.
    #This example is a simple application with single service and simple
  match routes.
    #The URL should point to the corresponding service.
    #Kubernetes provides an internal DNS mapping for services using the
  format <ServiceName>.<ServiceNamespace>.svc.cluster.local.
    http:
      - match:
          - uri:
              prefix: /
        rewrite:
          uri: /
        route:
          - destination:
              host: {{ include "test-app.fullname" . }}.
  {{ .Release.Namespace }}.svc.cluster.local
              port:
                number: {{ .Values.service.port }}
  ```

  Configure the `values.yaml` file of your application chart as follows:

  ```
  ezua:
    ... #other EZUA options

    virtualService:
      endpoint: "test-app.hpe-staging-ezaf.com"
      istioGateway: "istio-system/ezaf-gateway"
  ```

- Add the following configuration options for Kyverno policy to your application chart.

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: {{ printf
"add-vendor-app-labels-%s-%s" .Release.Name .Chart.Name }}
  annotations:
    "helm.sh/hook": pre-install
    "helm.sh/hook-weight": "-5"
    "helm.sh/hook-delete-policy": before-hook-creation
spec:
  background: false
  rules:
  - name: add-vendor-app-labels
    match:
      any:
      - resources:
          # list all namespaces defined by the chart here
          # if there are no namespace resource defined leave
only .Release.Namespace
          namespaces:
          - {{ .Release.Namespace }}
          kinds:
          - Pod
    mutate:
      patchStrategicMerge:
        metadata:
            labels:
              hpe-ezua/type: vendor-service
```

- Configure SSO for the applications you want to import. See SSO Support for Imported Frameworks on page 121.
- All the applications must be deployed as Helm charts. You must have the `tar.gz` file created from the Helm chart for the application you want to import.

About this task

In HPE Ezmeral Unified Analytics Software, you can bring your own Kubernetes customized runtimetools and frameworks. To start importing applications, follow these steps:

1. Click the **Tools & Frameworks** icon on the left navigation bar.
2. Click the **Import Framework** button on the top-right of the **Tools & Frameworks** screen. Navigate through each step within the **Import Framework** wizard:
    a. **Framework Details:** Set the following boxes on the **Framework Details** step:

Framework Name:	Enter the framework name.
Version:	Enter the framework version.
Description:	Enter the application description.
Category:	Select the application category from Data Engineering, Analytics, or Data Science.
Framework Icon:	Click **Select File** and browse the logo image for your application.

b. **Framework Chart:** Set the following boxes on the **Framework Chart** step:

Helm Chart:

Select **Upload New Chart** to import a new application. A list of all previously imported applications appears in the dropdown. If you deleted the previously imported application and you want to import the same application again, you can choose that application option from the dropdown.

NOTE:

If you are using a bitnami helm chart for your imported applications in HPE Ezmeral Unified Analytics Software, you must set the `volumePermissions` to `true` in the `values.yaml` file.

```
volumePermissions:

  enabled: true
```

When Bitnami starts up, it creates a directory inside the container.

When you set this value to `true`, it initiates the start of an init container that changes the owner of the PersistentVolume mount point.

When you set this value to `false`, the permissions remain unchanged, which prevents the creation of the directory, thus causing the container to fail.

Upload Helm Package tar.gz file:

Click **Select File** and browse the `tar.gz` of your application Helm chart.

Namespace:

Enter the namespace for framework.

Release Name:

Enter the name for this specific installation of Helm Chart.

Wait:

To wait until all the necessary services, volumes, pods, are in ready state before successfully importing the applications, check **Wait**.

Debug:

To get detailed information in error status, check **Debug**.

c. **Framework Values:** Configure the `override values` file of your application by using the **Helm Values (YAML)** box.

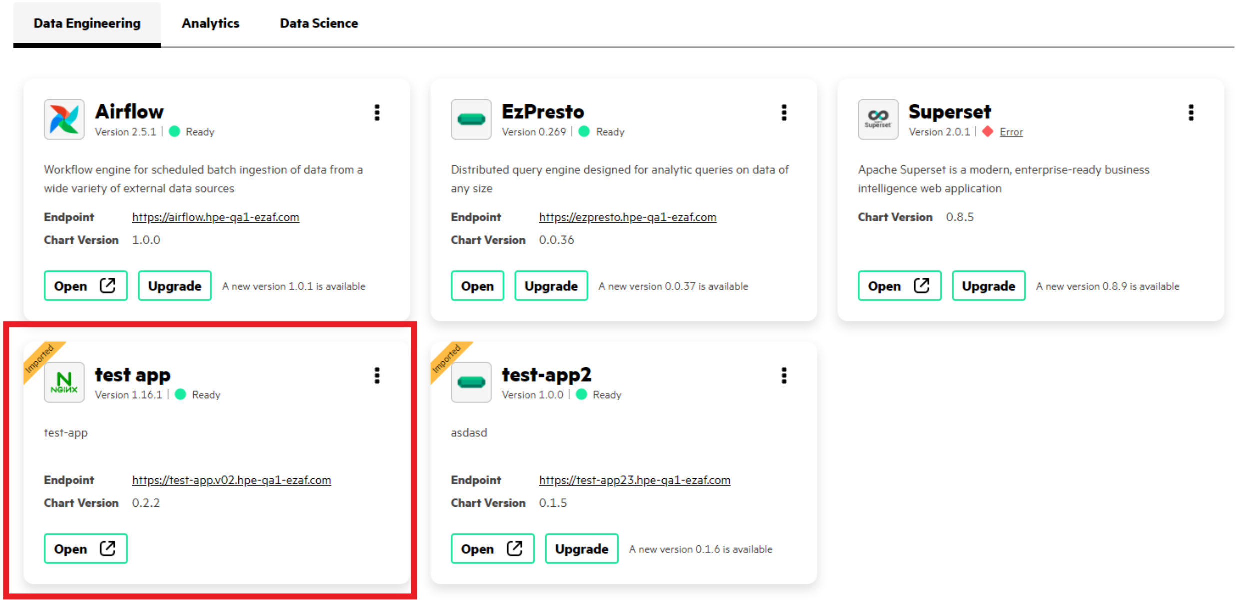
d. **Review:** Review the framework details. Click the **pencil** icon in each section to navigate to the specific step to change the framework configuration.

3. To import the framework, click **Submit** on the bottom right of the **Review** step.

Results

The application of your choice is imported and installed. You can view it on the **Tools & Frameworks** screen underneath your chosen application category.

For e.g.: If you imported **test-app** application under the **Data Engineering** category, you can view test-app on the **Tools & Frameworks** screen underneath the **Data Engineering** category.



Airflow
Version 2.5.1 | Ready
Workflow engine for scheduled batch ingestion of data from a wide variety of external data sources
Endpoint https://airflow.hpe-qa1-ezaf.com
Chart Version 1.0.0
A new version 1.0.1 is available

EzPresto
Version 0.269 | Ready
Distributed query engine designed for analytic queries on data of any size
Endpoint https://ezpresto.hpe-qa1-ezaf.com
Chart Version 0.0.36
A new version 0.0.37 is available

Superset
Version 2.0.1 | Error
Apache Superset is a modern, enterprise-ready business intelligence web application
Chart Version 0.8.5
A new version 0.8.9 is available

test app
Version 1.16.1 | Ready
test-app
Endpoint https://test-app.v02.hpe-qa1-ezaf.com
Chart Version 0.2.2

test-app2
Version 1.0.0 | Ready
asdasd
Endpoint https://test-app23.hpe-qa1-ezaf.com
Chart Version 0.1.5
A new version 0.1.6 is available

SSO Support for Imported Frameworks

Describes SSO support for imported frameworks integrated with native authentication and applications configured with authentication proxy.

Native Authentication Integrated Applications

Add the placeholders like `%%OIDC_ISSUER%%` and `%%LDAP_XXXX%%` in `values.yaml` file. HPE Ezmeral Unified Analytics Software automatically susbtitutes these placeholders with suitable values.

Authentication Proxy Configured Applications

Configure SSO with AuthorizationPolicy:

1. Configure the istio security AuthorizationPolicy before importing the application.

   Example of AuthorizationPolicy:

   ```
   apiVersion: security.istio.io/v1beta1
   kind: AuthorizationPolicy
   metadata:
     name: {{ .Release.Name }}-auth-policy
     namespace: {{ .Values.ezua.authorizationPolicy.namespace }}
   spec:
     action: CUSTOM
     provider:
       name: {{ .Values.ezua.authorizationPolicy.providerName }}
     rules:
       - to:
           - operation:
               hosts:
               - {{ .Values.ezua.virtualService.endpoint }}
     selector:
       {{- with .Values.ezua.authorizationPolicy.matchLabels }}
       matchLabels:
         {{- toYaml . | nindent 6 }}
       {{- end }}
   ```

2. Configure the `values.yaml` file of your application chart as follows:

   ```
   ezua:
     oidc:
       client_id: "${OIDC_CLIENT_ID}"
       client_secret: "${OIDC_CLIENT_SECRET}"
       domain: "${OIDC_DOMAIN}"

     domainName: "${DOMAIN_NAME}"
     #Use next options in order to configure the application endpoint.
     #Example of a VirtualService is here:
     virtualService:
       endpoint: "test-app.${DOMAIN_NAME}"
       istioGateway: "istio-system/ezaf-gateway"

     authorizationPolicy:
       namespace: "istio-system"
       providerName: "oauth2-proxy"
       matchLabels:
         istio: "ingressgateway"
   ```

Managing Imported Tools and Frameworks

Describes how to configure, delete, and update imported tools and frameworks in HPE Ezmeral Unified Analytics Software.

Prerequisites

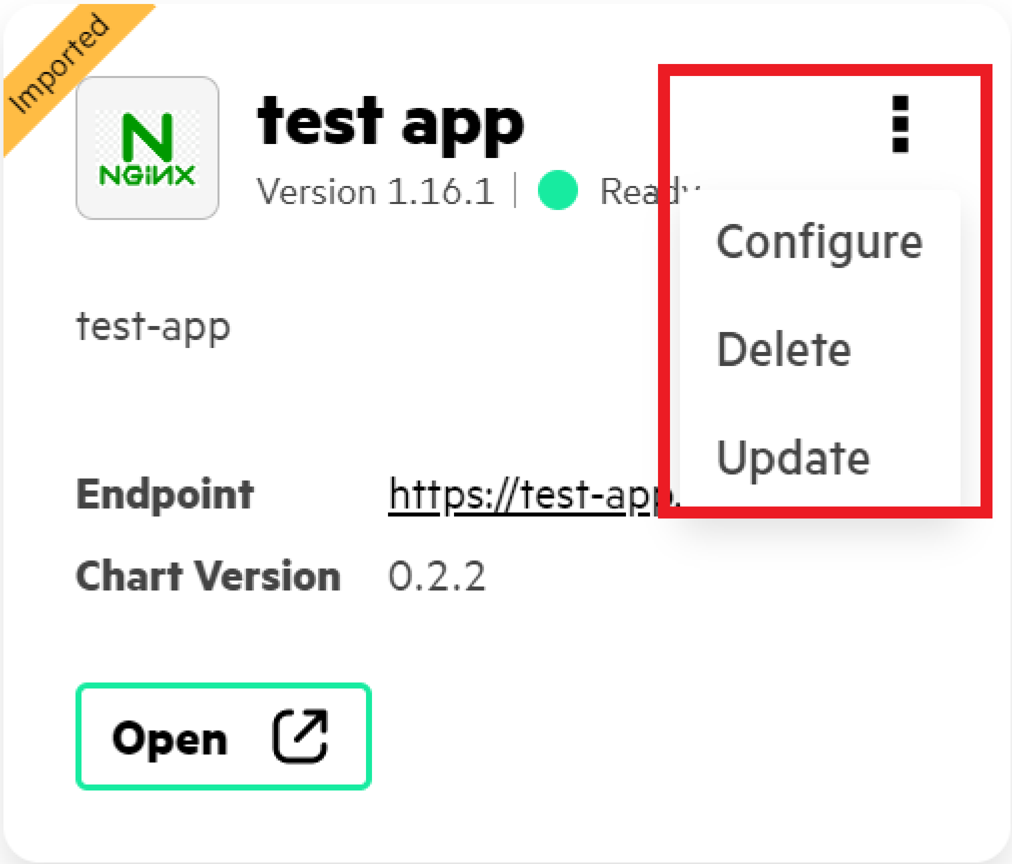
- Sign in to HPE Ezmeral Unified Analytics Software as Administrator.

About this task

You can configure, delete, or update imported applications and frameworks. Tiles for imported tools and frameworks display a yellow Imported label.

Procedure

1. In the left navigation bar, click **Tools & Frameworks**.
2. Click the **three-dots** on the tile of the application you want to manage.



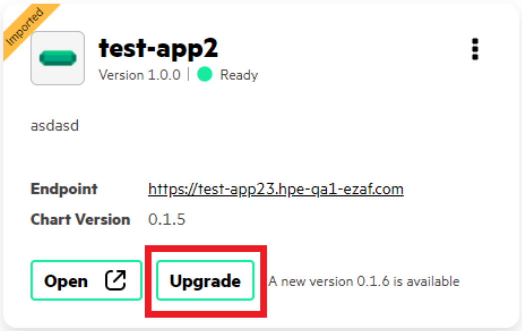
Perform one of the following tasks:

Configure	**a.** Select **Configure**. **b.** In the editor that opens, modify the application `values.yaml` file. **c.** Click **Configure** to apply the changes or **Cancel** to discard the changes.
Delete	To delete the application, select **Delete**. You can delete imported applications only. You cannot delete the applications that were installed with HPE Ezmeral Unified Analytics Software.
Update	! **ATTENTION:** You cannot undo the update action. **a.** Select **Update**. This **Update** option is only available for imported applications.

b. Browse to the location where the Helm chart is stored and select the Helm chart.

c. Click **Upload**. Clicking **Upload** enables the **Upgrade** button in the application tile.

d. To upgrade the application, click **Upgrade**.


Imported
test-app2
Version 1.0.0 | Ready
asdasd
Endpoint https://test-app23.hpe-qa1-ezaf.com
Chart Version 0.1.5
Open
Upgrade
A new version 0.1.6 is available

NOTE: A chart from the Chartmuseum is automatically deleted when an `ezappconfig` custom resource (CR) is deleted. This feature simplifies the management of imported tools and frameworks by ensuring that associated configurations and resources are removed seamlessly.

More information

Configuring Included Frameworks on page 124
Describes how to configure tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

Upgrading Included Frameworks on page 125
Describes how to upgrade tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

Configuring Included Frameworks

Describes how to configure tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

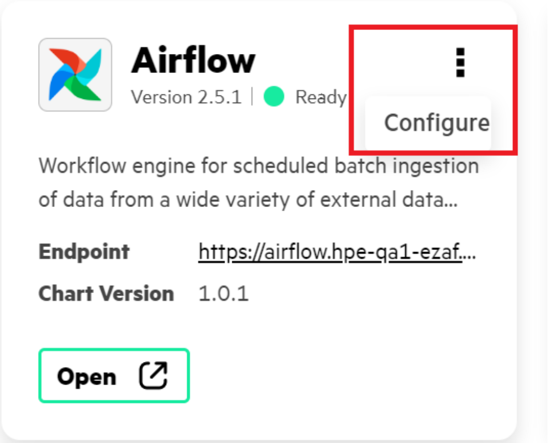
Prerequisites: Sign in to HPE Ezmeral Unified Analytics Software as Administrator.

You can configure the tools and frameworks that were installed with HPE Ezmeral Unified Analytics Software from the Tools & Frameworks screen or Settings screen.

Tools & Frameworks

To configure the included frameworks from the **Tools & Frameworks** screen, follow these steps:

1. In the left navigation bar, click **Tools & Frameworks**.
2. On the application tile, click the **three-dots** button.



Airflow

Version 2.5.1 | Ready

Workflow engine for scheduled batch ingestion of data from a wide variety of external data...

Endpoint https://airflow.hpe-qa1-ezaf...

Chart Version 1.0.1

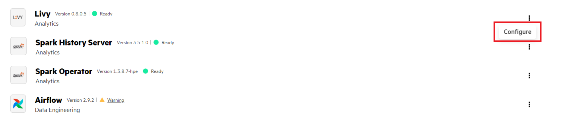
3. Select **Configure** to open the editor.
4. In the editor, modify the `values.yaml` file.
5. To apply the changes, click **Configure**, or to close the editor without any changes, click **Cancel**.

NOTE: When you use the **Configure** option to make configuration changes to included frameworks, the framework tile and associated tabs for that framework are not displayed while the framework is in the **Updating** state. Once the configuration is complete, the framework tile and associated tabs are displayed.

Settings

To configure the included frameworks from the **Settings** screen, follow these steps:

1. In the left navigation bar, navigate to **Administration > Settings > Tools & Frameworks**.
2. Select the framework you want to configure.
3. Select **Configure** to open the editor.



Livy Version 0.8.0.5 | Ready
Analytics

Spark History Server Version 3.5.1.0 | Ready
Analytics

Spark Operator Version 1.3.8.7-hpe | Ready
Analytics

Airflow Version 2.9.2 | Warning
Data Engineering

4. In the editor, modify the `values.yaml` file.
5. To apply the changes, click **Configure**, or to close the editor without any changes, click **Cancel**.

Upgrading Included Frameworks

Describes how to upgrade tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

Prerequisites: Sign in to HPE Ezmeral Unified Analytics Software as Administrator.

You can upgrade frameworks installed with HPE Ezmeral Unified Analytics Software when a new version is available.

You can upgrade the included frameworks in two different ways. They are:

- **Automatic downloads of framework updates**: If you are using a connected (non-air-gapped) environment, you can upgrade the included frameworks by enabling automatic downloads. To learn more, see Automatic Downloads of Framework Updates on page 126.
- **Manual downloads of framework updates**: If you are using a disconnected (air-gapped) environment, you must manually upgrade the included frameworks. However, you can also manually download framework updates in the connected environment. To learn more, see Manual Downloads of Framework Updates on page 128.

Automatic Downloads of Framework Updates

Describes how to automatically upgrade tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

Prerequisites:

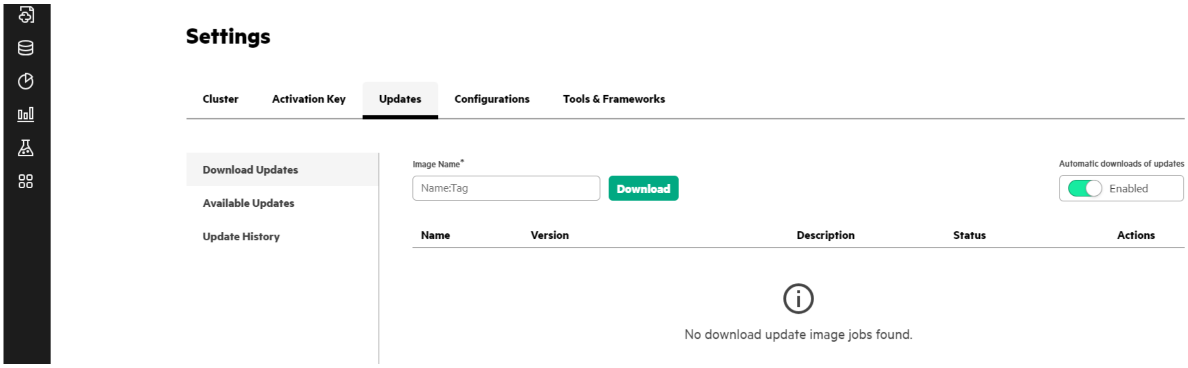
- Sign in to HPE Ezmeral Unified Analytics Software as Administrator.

You can simultaneously upgrade frameworks by enabling automatic downloads of updates when new versions are available. If frameworks have updates available for a new version, every hour HPE Ezmeral Unified Analytics Software bundles those updates and displays the update bundle on the **Available Updates** table as **Tools & Frameworks**.

To upgrade frameworks by enabling automatic downloads, follow these steps:

1. In the left navigation bar, click **Administration Settings**.
2. Click **Updates Download Updates**.
3. To enable the automatic download of framework updates, toggle **Automatic downloads of updates** to **Enabled**.

   NOTE: To disable automatic download of framework updates, toggle **Automatic downloads of updates** to **Disabled**.



Automatic downloads of updates: Enabled

No download update image jobs found.

4. Once you see the **Tools & Frameworks** update bundle in the **Available Updates** table, click the bundle to view details. In the **Details** dialog box, you can see the name, description, the current version of the framework and chart, and the new available version for the framework and chart. Once you see the new available versions for the update, you can perform the following actions by clicking on the **Actions** menu.

Update	To batch update frameworks immediately, follow these steps:

<table>
<tr>
<td></td>
<td>

a. Click **Update** in the **Actions** menu. This will open an **Update Now** dialog box and you can compare the current and new available versions of frameworks for upgrade.

b. Click **Update Now** to immediately start framework updates. Wait for framework updates to be in the **In Progress** status.

NOTE: You cannot cancel framework updates once it is in the **In Progress** status.

Result: You can navigate to the **Tools & Frameworks** screen to see frameworks are now in the **Upgrading** status.

</td>
</tr>
<tr>
<td>**Schedule**</td>
<td>

To schedule batch framework updates for later, follow these steps:

a. Click **Schedule** in the Actions menu. This will open a **Schedule Update** dialog box and you can compare the current and new available versions of frameworks for upgrade.

b. Select a date and time to schedule the update. Wait for framework updates to be in the **Scheduled** status.

c. Once the framework updates are in the **Scheduled** status, you can perform the following actions from the **Actions** menu.

<table>
<tr><td>**Cancel**</td><td>You can cancel the scheduled updates any time before update starts or if updates are not in the **In Progress** status yet.</td></tr>
<tr><td>**Reschedule**</td><td>You can reschedule the scheduled updates any time before update starts or if updates are not in the **In Progress** status yet.</td></tr>
<tr><td>**Update**</td><td>You can update frameworks immediately even though it has been scheduled for a later date and time.</td></tr>
</table>

</td>
</tr>
</table>

5. Once your updates are complete, the **Tools & Frameworks** update details will be displayed in the Update History table. You can click **Tools & Frameworks** to view details. In the **Details** dialog box, you can see the name, description, the current version of the framework and chart, and the new available version for the framework and chart.

Failure and Rollback

When you are upgrading frameworks, if one of the framework updates fails, the application tile for that framework will be in the **Error** status and the failed application will be rollbacked to the previous version from which you were upgrading to the new version.

For example: In HPE Ezmeral Unified Analytics Software, if you upgraded ten frameworks and nine frameworks are upgraded and in the **Ready** status, and if one farmework upgrade failed and is in the **Error** status with a warning message on the framework tile, then only that failed application is rolled back to the previous version whereas nine frameworks are successfully upgraded to new versions.

If for some reason the rollback fails and the framework is in the error state, you must contact HPE support to resolve this issue.

Manual Downloads of Framework Updates

Describes how to manually upgrade tools and frameworks included with the HPE Ezmeral Unified Analytics Software installation.

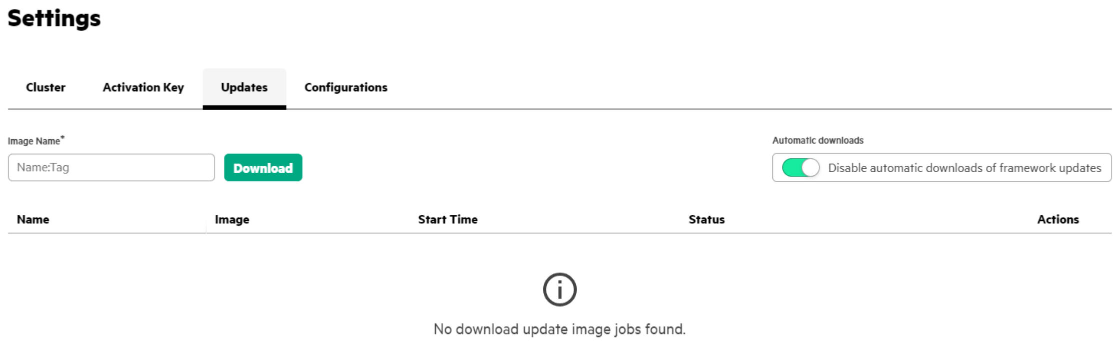
Prerequisites: Sign in to HPE Ezmeral Unified Analytics Software as Administrator.

You can upgrade the applications by getting the upgrade bundle (Docker container image) from HPE Support.

(Air-gapped environment only) Before you complete the following steps, use the air-gap utility to download the upgrade bundle (Docker container image) and then push those images into the air-gap registry. To learn about how to use the air-gap utility, see Using the Air Gap Utility on page 95.

To manually upgrade the included application, follow these steps:

1. In the left navigation bar, click **Administration Settings**.
2. Click the **Updates** tab.
3. Enter the **Image Name**.
4. Click **Download**.
5. View the downloaded image in the table.



Disable automatic downloads of framework updates

Name	Image	Start Time	Status	Actions

No download update image jobs found.

NOTE: To remove an image, click **Delete**.

6. After successfully downloading a new image, wait for a few minutes to see the **Tools & Frameworks** update bundle on the **Available Updates** table.
7. Once you see the **Tools & Frameworks** update bundle in the **Available Updates** table, click the bundle to view details. In the **Details** dialog box, you can see the name, description, the current version of the framework and chart, and the new available version for the framework and chart. Once you see

the new available versions for the update, you can perform the following actions by clicking on the **Actions** menu.

Update

To batch update frameworks immediately, follow these steps:

a. Click **Update** in the **Actions** menu. This will open an **Update Now** dialog box and you can compare the current and new available versions of frameworks for upgrade.

b. Click **Update Now** to immediately start framework updates. Wait for framework updates to be in the **In Progress** status.

> **NOTE:** You cannot cancel framework updates once it is in the **In Progress** status.

Result: You can navigate to the **Tools & Frameworks** screen to see frameworks are now in the **Upgrading** status.

Schedule

To schedule batch framework updates for later, follow these steps:

a. Click **Schedule** in the Actions menu. This will open a **Schedule Update** dialog box and you can compare the current and new available versions of frameworks for upgrade.

b. Select a date and time to schedule the update. Wait for framework updates to be in the **Scheduled** status.

c. Once the framework updates are in the **Scheduled** status, you can perform the following actions from the **Actions** menu.

Cancel

You can cancel the scheduled updates any time before update starts or if updates are not in the **In Progress** status yet.

Reschedule

You can reschedule the scheduled updates any time before update starts or if updates are not in the **In Progress** status yet.

Update

You can update frameworks immediately even though it has been scheduled for a later date and time.

8. Once your updates are complete, the **Tools & Frameworks** update details will be displayed in the Update History table. You can click **Tools & Frameworks** to view details. In the **Details** dialog box, you can see the name, description, the current version of the framework and chart, and the new available version for the framework and chart.

Installing Included Frameworks Post Unified Analytics Installation

Describes how to install included frameworks post HPE Ezmeral Unified Analytics Software installation.

Prerequisites

1. Sign in to HPE Ezmeral Unified Analytics Software as Administrator.
2. Ensure you have the required CPU and memory resources to install the framework.

About this task

You can now select specific frameworks to exclude during the HPE Ezmeral Unified Analytics Software installation. Even if you choose not to install them during the initial installation, you can install these frameworks afterward.

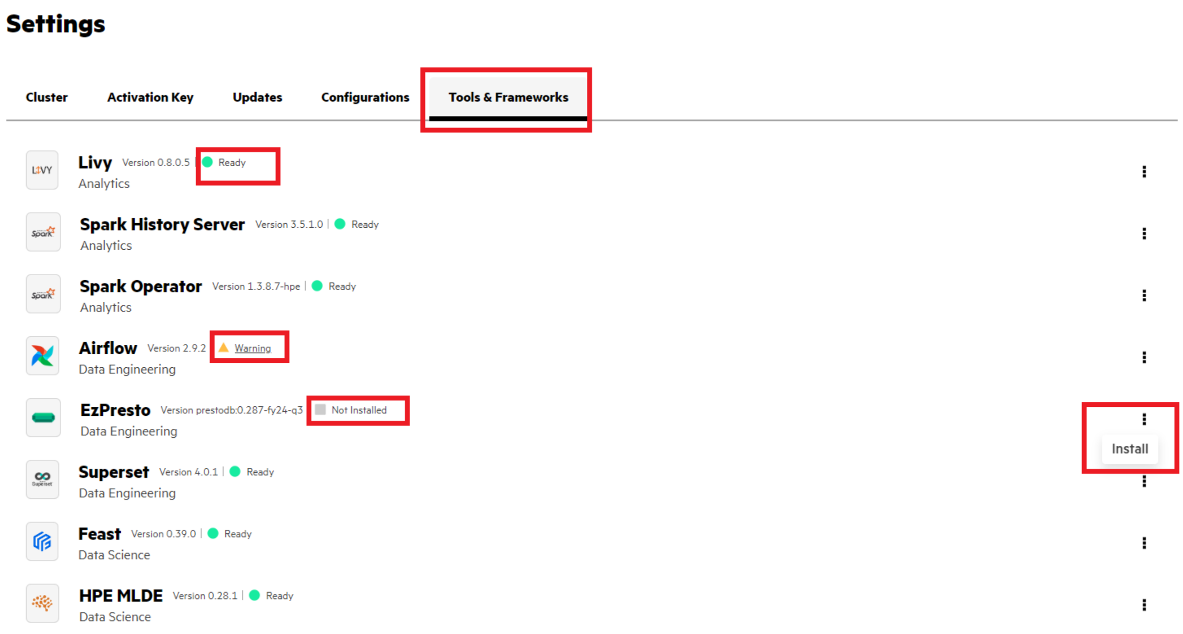
The following frameworks can be excluded during the Unified Analytics installation and installed later as needed:

- Superset
- EzPresto
- Livy
- Feast
- HPE MLDE

To install frameworks post-installation, follow these steps:

Procedure

1. Navigate to **Administration > Settings > Tools & Frameworks**.
2. Select the framework you want to install and select **Install** from the menu icon. Confirm the prompt in the dialog box as required.


Settings
Cluster
Activation Key
Updates
Configurations
Tools & Frameworks
Livy Version 0.8.0.5 | Ready
Analytics
Spark History Server Version 3.5.1.0 | Ready
Analytics
Spark Operator Version 1.3.8.7-hpe | Ready
Analytics
Airflow Version 2.9.2 | Warning
Data Engineering
EzPresto Version prestodb:0.287-fy24-q3 | Not Installed
Data Engineering
Install
Superset Version 4.0.1 | Ready
Data Engineering
Feast Version 0.39.0 | Ready
Data Science
HPE MLDE Version 0.28.1 | Ready
Data Science

NOTE: If the framework installation fails, select **Retry**.

Results

Once the framework is successfully installed, it will have a **Ready** status. You can navigate to the **Tools & Frameworks** screen to view the tile for the newly installed framework.

Connecting to External S3 Object Stores

Describes how to connect HPE Ezmeral Unified Analytics Software to external S3 object storage in AWS, MinIO, and HPE Ezmeral Data Fabric Object Store.

Administrators can connect HPE Ezmeral Unified Analytics Software to object storage in AWS S3, MinIO, HPE Ezmeral Data Fabric Object Store, and HPE GreenLake for File Storage. Users can then access data in the connected data sources through clients, such as Spark and Kubeflow notebooks, without providing an access or secret key.

When you configure the data source connection, you provide HPE Ezmeral Unified Analytics Software with the access credentials (access key and secret key); the user does not need the access credentials because HPE Ezmeral Unified Analytics Software uses a proxy to communicate with clients.

Clients talk to the HPE Ezmeral Unified Analytics Software proxy through the data source endpoint URL and pass JWT tokens to authenticate users. Users configure clients to talk to the connected object store. Users provide the client with the data source name and endpoint URL (as they appear on the data source tile in the HPE Ezmeral Unified Analytics Software UI), as well as the bucket they want the client to access.

How to Connect HPE Ezmeral Unified Analytics to Object Storage

Regardless of which object store you connect to, the general steps are the same with the exception of a few connection parameters.

IMPORTANT: You can create multiple object store connections. Each object store connection that you create must have a unique name.

To connect to an object store:

1. Sign in to HPE Ezmeral Unified Analytics Software.

2. In the left navigation bar, select **Data Engineering > Data Sources**.
3. On the **Data Sources** screen, select the **Object Store Data** tab.

   NOTE: By default, a local-s3 Ezmeral Data Fabric tile is displayed. This Ezmeral Data Fabric version of S3 is a local S3 version used internally by HPE Ezmeral Unified Analytics Software and cannot be deleted. Do not connect to this data source.

4. Click **Add New Data Source**.
5. Click the **Add…** button in one of the tiles (HPE Ezmeral Data Fabric Object Store, Amazon, MinIO, or HPE GreenLake for File Storage).
6. In the drawer that opens, enter the connection properties:

HPE Ezmeral Data Fabric Object Store

To connect to HPE Ezmeral Data Fabric Object Store, provide the following information:

- **Name** - Enter a unique name for the data source.
- **Endpoint** - Enter the HPE Ezmeral Data Fabric Object Store URL, for example:

  ```
  https://<ip-address>:9000
  ```

  To connect to a secured HPE Ezmeral Data Fabric Object Store, enter the fully qualified domain name (FQDN) of the external HPE Ezmeral Data Fabric Object Store node, for example:

  ```
  https://
  <FQDN-of-external-DF-s3-node>:9000
  ```

- **Access Key** - Enter the HPE Ezmeral Data Fabric Object Store access key.
- **Secret Key** - Enter the HPE Ezmeral Data Fabric Object Store secret key.
- **Insecure** - Only select this option for POCs or demos; do not select for production environments. If you do not select this option, you must add the root CA certificate for a secured connection.

  For a secure HPE Ezmeral Data Fabric Object Store connection, enter the path to the root CA certificate on the node that you specified as the endpoint. Typically, the root CA certificate path is:

  ```
  /opt/mapr/conf/ca/chain-ca.pem
  ```

AWS S3

To connect to AWS S3, provide the following information:

- **Name** - Enter a unique name for the data source.
- **Endpoint** - Enter the AWS S3 URL, for example `https://s3.us-east-20.amazonaws.com`.

	• **Access Key** - Enter the AWS S3 access key. **TIP:** The access key and secret key are associated with the IAM user in AWS. The IAM policy associated with the user should permit access to buckets. For example, the IAM policy should grant the user read, write, and/or create access on buckets. • **Secret Key** - Enter the AWS S3 secret key. • **AWS Region** - Enter the AWS region.
MinIO	To connect to MinIO, provide the following information: • **Name** - Enter a unique name for the data source. • **Endpoint** - Enter the MinIO URL. • **Access Key** - Enter the MinIO access key. • **Secret Key** - Enter the MinIO secret key. • **Insecure** - Only select this option for POCs or demos; do not select for production environments. When the option is not selected, you must add the root CA certificate for a secured connection. • **Root Certificate** - This is a TLS mode configuration. Add the root CA certificate bundle.

7. Click **Add**. The data source is connected and a new tile for the data source displays on the **Data Sources** screen.

   IMPORTANT: The data source *name* and *endpoint URL* display on the tile. Users need this information to connect their clients to the data source. Users can navigate to the **Data Sources** screen to get the information. See Accessing Data in External S3 Object Stores on page 292.

HPE GreenLake for File Storage

To connect to HPE GreenLake for File Storage, provide the following information:

- **Name** - Enter a unique name for the data source.
- **Endpoint** - Enter the MinIO URL.
- **Access Key** - Enter the MinIO access key.
- **Secret Key** - Enter the MinIO secret key.
- **Insecure** - Only select this option for POCs or demos; do not select for production environments. When the option is not selected, you must add the root CA certificate for a secured connection.
- **Root Certificate** - This is a TLS mode configuration. Add the root CA certificate bundle.

Connecting to HPE Ezmeral Data Fabric

Describes how to connect HPE Ezmeral Unified Analytics Software to an external HPE Ezmeral Data Fabric cluster.

To connect HPE Ezmeral Unified Analytics Software to an HPE Ezmeral Data Fabric cluster, you must provide the following information for the HPE Ezmeral Data Fabric cluster:

CLDB nodes (hostnames or IP addresses)

To get a list of the CLDB hosts, run the following command on the HPE Ezmeral Data Fabric cluster:

```
maprcli node listcldbs -cluster
<cluster name> -json
```

mapruserticket

When you connect HPE Ezmeral Unified Analytics Software to an external HPE Ezmeral Data Fabric cluster, you must provide the `mapruserticket` to create a connection that enables users to access HPE Ezmeral Data Fabric from HPE Ezmeral Unified Analytics Software.

To get the `mapruserticket`, run the following command on the HPE Ezmeral Data Fabric cluster:

```
sudo cat /opt/mapr/conf/
mapruserticket
```

Volume path

This is the path to the volume in HPE Ezmeral Data Fabric that you want to connect to from HPE Ezmeral Unified Analytics Software. For information about volumes, see Managing Data with Volumes.

Complete the following steps to connect HPE Ezmeral Unified Analytics Software to an external HPE Ezmeral Data Fabric cluster:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation panel, select **Data Engineering > Data Sources**.
3. On the **Data Sources** page, select the **Data Volumes** tab.
4. Click **New Volume**.
5. On the **Data Volumes** page, click **Add HPE Ezmeral Data Fabric** in the HPE Ezmeral Data Fabric tile.
6. In the drawer that opens, enter the following required information:

Field	Description
Name	Enter a name for the HPE Ezmeral Data Fabric connection. Each HPE Ezmeral Data Fabric connection that you create must have a unique name.
CLDB Hosts	List one or more CLDB hostnames or IP addresses with the port number. If entering more than one CLDB host, use a comma to separate each host name or IP address, for example: `cldb.node.01:7222,cldb.node.02:7222,cldb.node.03:7222`
Service Ticket	Paste the `mapruserticket` into the field.
Volume Path	Enter the path to the mounted volume in the HPE Ezmeral Data Fabric cluster.

7. Click **Add**. The HPE Ezmeral Data Fabric connection is listed on the **Data Volumes** tab on the **Data Sources** page. **Status** indicates the connection status.

   TIP: When you connect HPE Ezmeral Unified Analytics Software to an external HPE Ezmeral Data Fabric cluster, it can take one to two minutes for the synchronization with the cluster to complete. Once synchronized, the HPE Ezmeral Data Fabricconnection **Status** column displays **Ready** (green light) and the HPE Ezmeral Data Fabric name changes to a clickable hyperlink. Click the hyperlink to browse directories and files in the connected volume.

Connecting to HPE GreenLake for File Storage

Describes how to connect Unified Analytics to an external HPE GreenLake for File Storage cluster.

Complete the following steps to connect HPE Ezmeral Unified Analytics Software to an external HPE GreenLake for File Storage cluster:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation panel, select **Data Engineering > Data Sources**.
3. On the **Data Sources** page, select the **Data Volumes** tab.
4. Click **New Volume**.
5. On the **Data Volumes** page, click **Add HPE GreenLake for File Storage** in the HPE GreenLake for File Storage tile.
6. In the drawer that opens, enter the following required information:

Field	Description
Name	Enter a unique name that identifies the HPE GreenLake for File Storage connection in your Unified Analytics cluster.
Storage	Enter the amount of storage available to Unified Analytics through the HPE GreenLake for File Storage connection.
Server	Enter the HPE GreenLake for File Storage endpoint that you want Unified Analytics to connect to.
Volume Share	Enter the volume path. The volume path correlates with a View in HPE GreenLake for File Storage.

7. Click **Add**. The HPE GreenLake for File Storage connection is listed on the **Data Volumes** tab on the **Data Sources** page. **Status** indicates the connection status.

   TIP: When you connect HPE Ezmeral Unified Analytics Software to an external HPE GreenLake for File Storage cluster, it can take one to two minutes for the synchronization with the cluster to complete. Once synchronized, the HPE GreenLake for File Storageconnection **Status** column displays **Ready** (green light) and the HPE GreenLake for File Storage name changes to a clickable hyperlink. Click the hyperlink to browse directories and files in the connected storage volume.

Configuring Endpoints

Describes the endpoints in HPE Ezmeral Unified Analytics Software and how to configure them.

Configure endpoints in HPE Ezmeral Unified Analytics Software by going to **Administration > Settings** and selecting the **Configurations** tab.

The following sections provide details for each type of endpoint on the **Configurations** tab:

OTel Endpoint

The OTel endpoint is the target URL where HPE Ezmeral Unified Analytics Software OTel exporter sends metrics. The OTel endpoint enables other OTel collectors to receive cluster metrics in OTel format.

When you register an OTel endpoint, the cluster OTel collector exports metric data to the customer OTel collector hosted at the OTel endpoint. This includes Prometheus metrics about cluster performance, billing/metering related data, and app-based metrics for Kubeflow, Spark, and Ray. You can also export the incoming data to tools, such as Grafana or Elasticsearch.

OTEL is the standard format for metrics collection. Data only persists for 60 days in prometheus.

Use the following OTel endpoint format:

```
<host>:<port>
```

The OTel endpoint format:

- Must be a valid HTTPS host
- May contain a port
- Should contain a path
- Cannot contain other parts, such as a query string or fragment

JDBC Endpoint

The JDBC endpoint is automatically created when you install and configure HPE Ezmeral Unified Analytics Software.

To connect EzPresto to external applications, see Connecting External Applications to EzPresto via JDBC on page 345.

EzCentral Forwarding

NOTE: EzCentral Forwarding is not supported for air-gapped (disconnected) environments.

In HPE Ezmeral Unified Analytics Software, you can opt-in to forward your metrics collected by Prometheus to EzCentral via OTEL in real-time. EzCentral is a platform managed by HPE that can monitor your HPE Ezmeral Unified Analytics Software clusters when you enable EzCentral Forwarding.

The metrics forwarding to EzCentral has the following benefits:

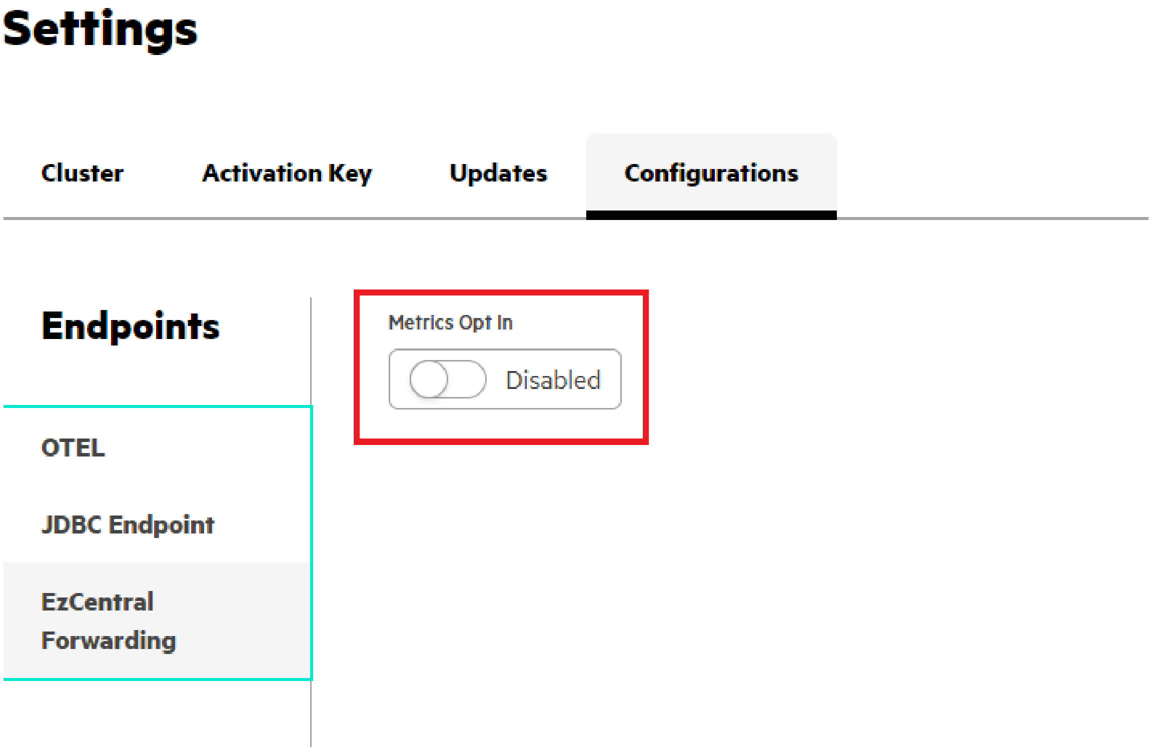
- Fast resolution of cluster issues by HPE through efficient cluster management and administration.
- Provides real-time alerts to HPE enabling immediate incident resolution.
- Enables HPE to warn you regarding unhealthy clusters and take actions to resolve issues.
- Provides valuable metrics to fix bugs and improve user experience.

The metrics forwarding to EzCentral is enabled by default.

The forwarded metrics include:

- Node – CPU or Memory or Disk Usage
- Container – CPU or Memory
- GPU usage

To disable metrics forwarding to EzCentral, toggle off the Metrics Opt In button as follows:



GPU Support

Provides information about support for NVIDIA GPU, MIG partitioning, preparing hosts for GPU-enabled environment, adding hosts and enabling GPU in HPE Ezmeral Unified Analytics Software.

GPUs provide essential computational power and parallel processing capabilities to accelerate the training and inference processes of deep learning models, reading and processing data frames, processing SQL queries within Spark, and running experiments using Jupyter notebooks integrated with GPUs.

The hundreds or thousands of smaller cores working in parallel enable GPUs to process massive amounts of data in a short period of time.

HPE Ezmeral Unified Analytics Software supports homogenous configuration deployment where the GPU is split into N equal parts with the same amount of memory and CUDA cores. All GPU models on the same Kubernetes cluster must operate in the same configuration mode. HPE Ezmeral Unified Analytics Software does not support any mixed configuration across multiple GPU models.

Supported GPU Models

To see the GPU models supported by HPE Ezmeral Unified Analytics Software, see GPU Models on page 205 .

MIG Partitioning

HPE Ezmeral Unified Analytics Software supports single-access multi-instance GPU. You can use MIG GPU when there are multiple applications that require GPU acceleration. By using MIG, you can achieve higher resource utilization and cost efficiency.

In HPE Ezmeral Unified Analytics Software, GPU partitions are presented as whole devices by using the MIG mechanism. When an application requests one GPU, the application receives a partition. Only one GPU device is visible to the application. To learn more, see CUDA visible devices.

During the installation of HPE Ezmeral Unified Analytics Software, you must specify GPU partition size (Whole, Large, Medium, and Small) and request the number of GPU instances required for the workload.

A100

For A100 GPU, the partition size maps to the following profiles:

Unified Analytics vGPU Size	No. of Unified Analytics vGPUs per physical GPU	MIG Profile - A100-40 GB	MIG Profile - A100-80 GB	Description
Whole	1–100%	No MIG - entire physical GPU	No MIG - entire physical GPU	A100 GPU models are not split into any partitions. You will get the entire physical GPU. In this configuration, applications can use only one virtual GPU at a time.
Large	2 – 42% each	3g.20gb	3g.40gb	A100 GPU models are split into two equal partitions. In this configuration, 16% of the GPU will remain idle.

Unified Analytics vGPU Size	No. of Unified Analytics vGPUs per physical GPU	MIG Profile - A100-40 GB	MIG Profile - A100-80 GB	Description
Medium	3 – 28% each	2g.10gb	2g.20gb	A100 GPU models are split into three equal partitions. In this configuration, 16% of the GPU will remain idle.
Small	7 – 14% each	1g.5gb	1g.10gb	A100 GPU models are split into seven equal partitions. In this configuration, 2% of the GPU will remain idle.

A30

For A30 GPU, the partition size maps to the following profiles:

Unified Analytics vGPU Size	No. of Unified Analytics vGPUs per physical GPU	MIG Profile - A30-24GB	Description
Whole	1–100%	No MIG - entire physical GPU	A30 GPU models are not split into any partitions. You will get the entire physical GPU. In this configuration, applications can use only one virtual GPU at a time.
Large	2 – 50% each	all-2g.12gb	A30 GPU models are split into two equal partitions. This configuration will utilize 100% of the GPU.
Small	4 – 25% each	all-1g.6gb	A30 GPU models are split into four equal partitions. This configuration will utilize 100% of the GPU.

H100-NVL

For H100-NVL GPU, the partition size maps to the following profiles:

Unified Analytics vGPU Size	No. of Unified Analytics vGPUs per physical GPU	MIG Profile - H100-NVL	Description
Whole	1–100%	No MIG - entire physical GPU	H100 GPU models are not split into any partitions. You will get the entire physical GPU. In this configuration, applications can use only one virtual GPU at a time.
Large	2 – 45% each	3g-47gb	H100 GPU models are split into two equal partitions. In this configuration, 10% of the GPU will remain idle.
Medium	3 – 24% each	2g.24gb	H100 GPU models are split into three equal partitions. In this configuration, 28% of the GPU will remain idle.
Small	7 – 12% each	1g.12gb	H100 GPU models are split into seven equal partitions. In this configuration, 16% of the GPU will remain idle.

H100-PCI

For H100-PCI GPU, the partition size maps to the following profiles:

NOTE: To use the H100-PCI GPU, manually update the MIG profile values in the `gpunodeconfig` CR.

Unified Analytics vGPU Size	No. of Unified Analytics vGPUs per physical GPU	MIG Profile - H100-PCI	Description
Whole	1–100%	No MIG - entire physical GPU	H100 GPU models are not split into any partitions. You will get the entire physical GPU. In this configuration, applications can use only one virtual GPU at a time.
Large	2 – 40% each	3g-40gb	H100 GPU models are split into two equal partitions. In this configuration, 20% of the GPU will remain idle.
Medium	3 – 26% each	2g.20gb	H100 GPU models are split into three equal partitions. In this configuration, 22% of the GPU will remain idle.

Unified Analytics vGPU Size	No. of Unified Analytics vGPUs per physical GPU	MIG Profile - H100-PCI	Description
Small	7 – 12% each	1g.10gb	H100 GPU models are split into seven equal partitions. In this configuration, 16% of the GPU will remain idle.

NOTE: If you are using a GPU host on GCP (Google Cloud Platform), you must reboot after deployment to perform MIG partitioning. If you do not reboot, MIG will not be enabled.

To learn about MIG profile names, see MIG Device Names.

Preparing the GPU Environment

HPE Ezmeral Unified Analytics Software 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 Unified Analytics Software.

HPE Ezmeral Unified Analytics Software supports user-provided deployment.

Preparing hosts to use GPU in the user-provided host model:

If you want to use GPU in HPE Ezmeral Unified Analytics Software, you will have to prepare your hosts. The following are the requirements for preparing your hosts before installation to use GPU in HPE Ezmeral Unified Analytics Software:

- The host can be bare metal or VM with GPU pass-through, or an AWS EC2 instance.
- Install the latest version of the supported operating system. To learn about the supported operating system versions for GPU in HPE Ezmeral Unified Analytics Software, see Operating System on page 205.

  NOTE: Do not use operating systems with pre-installed NVIDIA drivers. HPE Ezmeral Unified Analytics Software does not support operating systems with pre-installed NVIDIA drivers. The GPU operator automatically installs NVIDIA drivers when the host is added to HPE Ezmeral Unified Analytics Software.

- Disable SELinux on the host before adding the host to HPE Ezmeral Unified Analytics Software. This is the NVIDIA limitation, see GPU Operator with RHEL8/SELinux.

  NOTE: After successfully adding the host to HPE Ezmeral Unified Analytics Software cluster and the successfull NVIDIA driver install through the GPU operator, you can enable SELinux on that host and set it to enforcing mode.

To learn more about user-provided hosts, see Installing on User-Provided Hosts (Connected and Air-gapped Environments) on page 63.

Environments	Description
vSphere VM	Configure the VMs in the GPU pass-through setup by following the steps in VMware setting up GPU pass-through documentation. Add hosts to the HPE Ezmeral Unified Analytics Software.
AWS	Use the AWS account with access to provision GPU-based instances (p4d.24xlarge, and p4de.24xlarge EC2 instances). Deploy the A100 EC2 instance (P4d instance) with the AMI image in the supported operating system. Add hosts to the HPE Ezmeral Unified Analytics Software.

Adding Hosts and Enabling GPU Environment

After you have prepared hosts to work in the GPU-enabled environment, you must add them to the HPE Ezmeral Unified Analytics Software during the installation or during cluster expansion. After adding the host, the GPU is enabled automatically.

Adding Hosts and Selecting GPU Environment During Installation

To add hosts and select the GPU environment in the cluster during installation, follow these steps :

1. Perform the installation instructions provided in the installation documentation for your deployment target until you reach the **Installation Details** step in the installation wizard. See Installation on page 63.
2. In the Installation Details step, to enable the GPU, check **Use GPU**.
   a. **vGPU:** Specify the vGPU instances for your cluster.

      The number of vGPUs allocated depends on the GPU configuration partition size, the number of added GPU worker hosts, and the number of GPU cards per host. The number of allocated vGPUs may be less than the number of requested vGPUs.

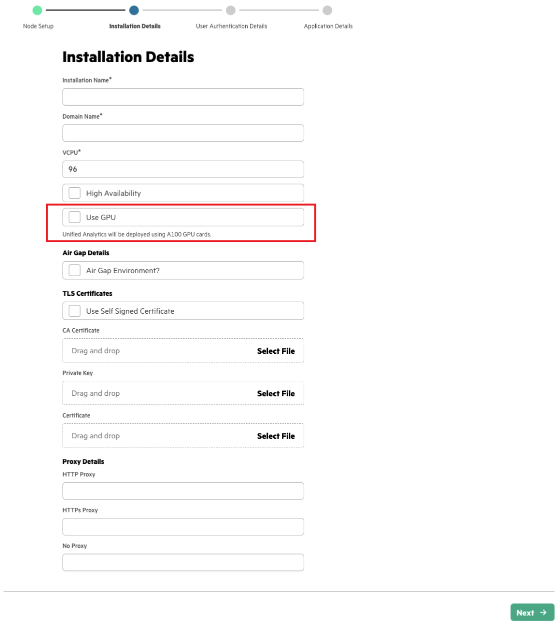
      For example: If one A100 GPU host is added with two GPU cards with the following configurations:

      - vGPU request: 10 vGPUs
      - vGPU configuration: large

      Then the number of allocated vGPUs is as follows:

      - vGPUs allocated: 2 x 2 large per GPU card = 4

   b. **GPU Configuration:** Specify the GPU partition size.



Installation Details

Installation Name*

Domain Name*

VCPU*
96

□ High Availability

□ Use GPU
Unified Analytics will be deployed using A100 GPU cards.

Air Gap Details
□ Air Gap Environment?

TLS Certificates
□ Use Self Signed Certificate

CA Certificate
Drag and drop **Select File**

Private Key
Drag and drop **Select File**

Certificate
Drag and drop **Select File**

Proxy Details
HTTP Proxy

HTTPs Proxy

No Proxy

As each node is added to the HPE Ezmeral Unified Analytics Software inventory node pool, HPE Ezmeral Unified Analytics Software configures the MIG profile if it detects MIG-capable devices (e.g., A100).

3. To specify the details for other boxes or options in the **Installation Details** step and to complete the cluster installation, refer to the installation documentation for your deployment target. See Installation on page 63.

Adding Hosts and Selecting GPU Environment During Cluster Expansion

To add hosts and select the GPU environment in the cluster during cluster expansion, follow these steps:

1. Perform the steps to expand the cluster until you reach the **Expand Cluster** screen. See Expanding the Cluster on page 113.

2. To enable the GPU, in the **Expand Cluster** screen, check **Use GPU**.

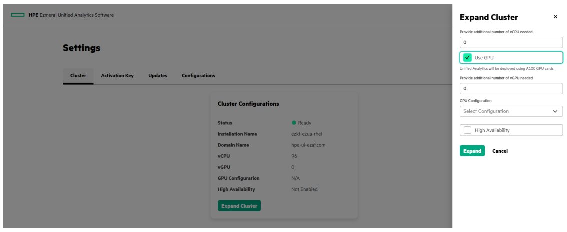
   NOTE: If you enabled the **Use GPU** option during the cluster installation, then that means GPU is already enabled and you cannot disable the **Use GPU** option while expanding the cluster.

a. **vGPU:** Specify the additional vGPU instances for your cluster.

   NOTE: The number of additional vGPUs allocated depends on the GPU configuration partition size, the number of added GPU worker hosts, and the number of GPU cards per host. The number of allocated vGPUs may be less than the number of requested vGPUs.

b. **GPU Configuration:** Specify the GPU partition size.

   NOTE: If you selected the partition size during the cluster installation, you can not update the partition size while expanding the cluster.



Cluster Configurations

Status	● Ready
Installation Name	ezkf-ezua-rhel
Domain Name	hpe-ui-ezaf.com
vCPU	96
vGPU	0
GPU Configuration	N/A
High Availability	Not Enabled

Expand Cluster

Provide additional number of vCPU needed: 0

☑ Use GPU

Unified Analytics will be deployed using A100 GPU cards

Provide additional number of vGPU needed: 0

GPU Configuration: Select Configuration

□ High Availability

3. To specify the details for other boxes or options in the **Expand Cluster** screen and to complete the cluster expansion, see Expanding the Cluster on page 113.

Viewing GPU Model Information

To retrieve the information about the GPU model installed in HPE Ezmeral Unified Analytics Software for your operating system, run:

```
lspci | grep -i nvidia
```

To learn more about supported operating systems for GPUs in HPE Ezmeral Unified Analytics Software, see Operating System on page 205.

Updating GPU Partition Size Post Installation

To update the GPU partition size, modify the `gpunodeconfig` CR as follows:

1. Verify that the GPU workloads are not running in the cluster. To verify, SSH into the GPU node, run the following command:

   ```
   chroot /run/nvidia/driver/ nvidia-smi
   ```

   Then, look for any processes consuming the GPU. If no processes are shown, the GPU is not in use. Ensure this is consistent across all GPU nodes in the cluster.

2. To fetch the resource name, SSH into the workload control plane node and run the following command:

   ```
   gpunodeconfig=$(kubectl get gpunodeconfig -n hpecp-gpunodeconfig -o
   jsonpath='{.items[].metadata.name}')
   ```

3. To update the partition size, update the `enabled` field with the partition size of your choice.

   For example, to update the partition size to `large`, run:

   ```
   kubectl patch gpunodeconfig $gpunodeconfig -n hpecp-gpunodeconfig --type
   merge -p '{"spec":{"enabled":"large"}}'
   ```

4. To verify that the new partition size is applied in your cluster, run the following command inside the driver daemon pod in the `hpecp-gpu-operator` namespace:

   ```
   nvidia-smi
   ```

Integrating GPU with Tools and Frameworks

In HPE Ezmeral Unified Analytics Software, both imported and included tools and frameworks support GPU. With a MIG configuration, only one GPU is assigned per application. Applications request GPUs using the `nvidia.com/gpu` resource specifier.

NOTE: HPE Ezmeral Unified Analytics Software does not support MIG specifier `nvidia.com/mig-Xg.YYgb`.

The following applications and frameworks support GPU in HPE Ezmeral Unified Analytics Software:

- Kubeflow Kale or KFP. See Enabling Kale Extension in Kubeflow Notebook on page 442.
- Kubeflow KServe. See Enabling GPU Support on Kubeflow Kserve Model Serving on page 411.
- Kubeflow Notebooks. See Creating GPU-Enabled Notebook Servers on page 438.
- Ray. See GPU Support for Ray on page 428.
- Spark. See Enabling GPU Support for Spark on page 399.

GPU Resource Management

Describes the GPU idle reclaim policy used for GPU resource management.

GPU resource management enables you to optimize the analytical workloads by distributing the GPU resources to various workloads so that each workload receives the necessary computing power.

HPE Ezmeral Unified Analytics Software implements the GPU idle reclaim feature to maximize GPU utilization by dynamically allocating and deallocating resources to different frameworks and workloads as needed. This prevents overallocation and underutilization of the GPU resources and increases efficiency.

GPU resource management uses a priority policy to ensure that critical workloads get the resources they need. The priority policy also allows lower-priority workloads to utilize the GPU when it is available.

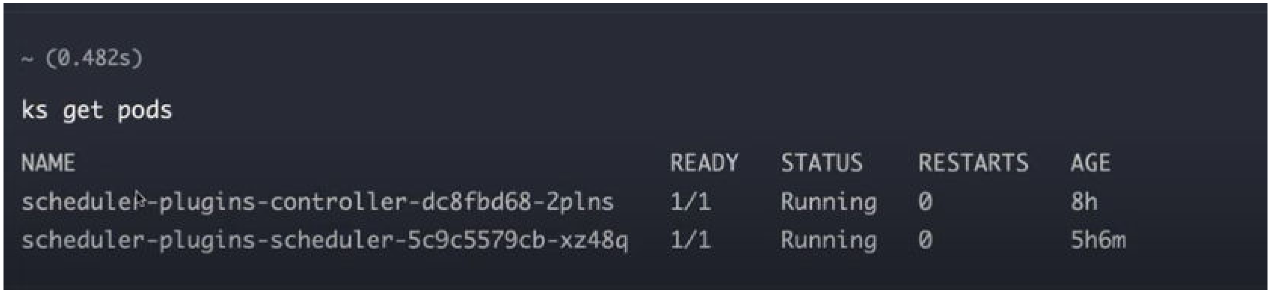
When a workload or framework is finished using its GPU resources, HPE Ezmeral Unified Analytics Software initiates GPU resource reclamation. This involves deallocating the resources and making them available for other workloads.

Custom Scheduler

HPE Ezmeral Unified Analytics Software runs its own scheduler which functions independently and is not connected to the default Kubernetes scheduler.

Note that the default Kubernetes scheduler is still available alongside this custom scheduler. The custom scheduler is an enhanced version of the default Kubernetes scheduler that includes the GPU idle reclaim plugins and the preemption tolerance.

The custom scheduler plugin governs all GPU workloads and is installed in the `hpe-scheduler-plugins` namespace. This namespace consists of a controller and a scheduler module. The scheduler is responsible for scheduling and reclaiming.

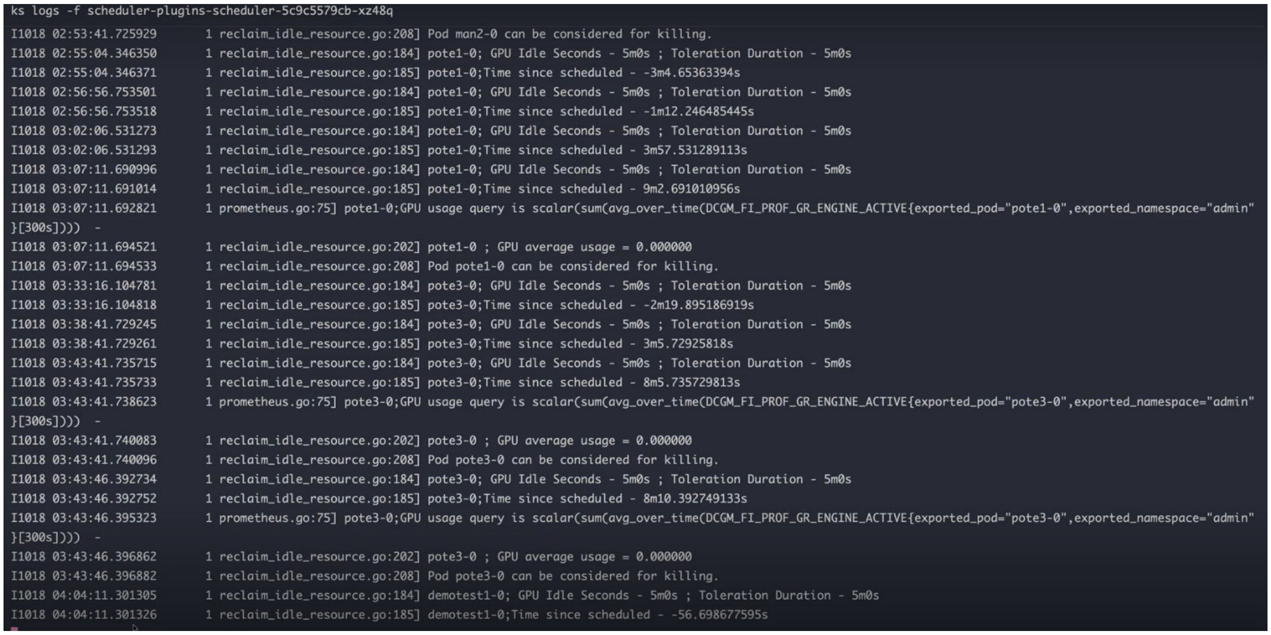


```
~ (0.482s)

ks get pods

NAME                                            READY   STATUS    RESTARTS   AGE
scheduler-plugins-controller-dc8fbd68-2plns     1/1     Running   0          8h
scheduler-plugins-scheduler-5c9c5579cb-xz48q    1/1     Running   0          5h6m
```

There are two pods in the scheduler namespace.
They are `scheduler-plugins-controller-dc8fbd68-2plns` and `scheduler-plugins-scheduler-5c9c5579cb-xz48q`. To view details of the GPU reclamation and pod preemption, see the logs for the `scheduler-plugins-scheduler-5c9c5579cb-xz48q` pod.

To see the logs, run:

```
kubectl logs -f scheduler-plugins-scheduler-5c9c5579cb-xz48q -n
hpe-scheduler-plugins
```



```
ks logs -f scheduler-plugins-scheduler-5c9c5579cb-xz48q
I1018 02:53:41.725929      1 reclaim_idle_resource.go:208] Pod man2-0 can be considered for killing.
I1018 02:55:04.346350      1 reclaim_idle_resource.go:184] pote1-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 02:55:04.346371      1 reclaim_idle_resource.go:185] pote1-0;Time since scheduled - -3m4.65363394s
I1018 02:56:56.753501      1 reclaim_idle_resource.go:184] pote1-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 02:56:56.753518      1 reclaim_idle_resource.go:185] pote1-0;Time since scheduled - -1m12.246485445s
I1018 03:02:06.531273      1 reclaim_idle_resource.go:184] pote1-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 03:02:06.531293      1 reclaim_idle_resource.go:185] pote1-0;Time since scheduled - 3m57.531289113s
I1018 03:07:11.690996      1 reclaim_idle_resource.go:184] pote1-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 03:07:11.691014      1 reclaim_idle_resource.go:185] pote1-0;Time since scheduled - 9m2.691010956s
I1018 03:07:11.692821      1 prometheus.go:75] pote1-0;GPU usage query is scalar(sum(avg_over_time(DCGM_FI_PROF_GR_ENGINE_ACTIVE{exported_pod="pote1-0",exported_namespace="admin"
}[300s])))  -
I1018 03:07:11.694521      1 reclaim_idle_resource.go:202] pote1-0 ; GPU average usage = 0.000000
I1018 03:07:11.694533      1 reclaim_idle_resource.go:208] Pod pote1-0 can be considered for killing.
I1018 03:33:16.104781      1 reclaim_idle_resource.go:184] pote3-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 03:33:16.104818      1 reclaim_idle_resource.go:185] pote3-0;Time since scheduled - -2m19.895186919s
I1018 03:38:41.729245      1 reclaim_idle_resource.go:184] pote3-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 03:38:41.729261      1 reclaim_idle_resource.go:185] pote3-0;Time since scheduled - 3m5.72925818s
I1018 03:43:41.735715      1 reclaim_idle_resource.go:184] pote3-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 03:43:41.735733      1 reclaim_idle_resource.go:185] pote3-0;Time since scheduled - 8m5.735729813s
I1018 03:43:41.738623      1 prometheus.go:75] pote3-0;GPU usage query is scalar(sum(avg_over_time(DCGM_FI_PROF_GR_ENGINE_ACTIVE{exported_pod="pote3-0",exported_namespace="admin"
}[300s])))  -
I1018 03:43:41.740083      1 reclaim_idle_resource.go:202] pote3-0 ; GPU average usage = 0.000000
I1018 03:43:41.740096      1 reclaim_idle_resource.go:208] Pod pote3-0 can be considered for killing.
I1018 03:43:46.392734      1 reclaim_idle_resource.go:184] pote3-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 03:43:46.392752      1 reclaim_idle_resource.go:185] pote3-0;Time since scheduled - 8m10.392749133s
I1018 03:43:46.395323      1 prometheus.go:75] pote3-0;GPU usage query is scalar(sum(avg_over_time(DCGM_FI_PROF_GR_ENGINE_ACTIVE{exported_pod="pote3-0",exported_namespace="admin"
}[300s])))  -
I1018 03:43:46.396862      1 reclaim_idle_resource.go:202] pote3-0 ; GPU average usage = 0.000000
I1018 03:43:46.396882      1 reclaim_idle_resource.go:208] Pod pote3-0 can be considered for killing.
I1018 04:04:11.301305      1 reclaim_idle_resource.go:184] demotest1-0; GPU Idle Seconds - 5m0s ; Toleration Duration - 5m0s
I1018 04:04:11.301326      1 reclaim_idle_resource.go:185] demotest1-0;Time since scheduled - -56.698677595s
```

Custom Scheduler Configurations

HPE Ezmeral Unified Analytics Software sets the default configurations for the tools and frameworks supporting GPU workloads so that the custom scheduler is used by default. The following tools and frameworks support GPU workloads:

- Kubeflow
- Spark
- Livy
- Ray
- HPE MLDE

Every GPU workload for Kubeflow, Spark, Livy, Ray, and HPE MLDE has the following configurations set as part of the pod spec to use the custom scheduler by default.

- `schedulerName: scheduler-plugins-scheduler`
- `priorityClass: <app_name>-<component_name>-gpu`
  - For example,
    - For Kubeflow notebooks: `kubeflow-notebook-gpu`
    - For Spark: `spark-gpu` (Note: There is no component name for Spark)

Only pods with their `spec.schedulerName` set to `scheduler-plugins-scheduler` are considered for reclaiming.

Do not modify these configurations for the GPU reclamation. If your GPU pod spec is not set to `scheduler-plugins-scheduler`, the default Kubernetes scheduler will operate instead of the custom scheduler.

The scheduler runs a cron job every 5-10 minutes. Every 5-10 minutes, the scheduler looks at the running pods and determines the feasibility of reclaiming pods based on their GPU usage and the annotation values set in the priority class attached to the pod. If the pod is eligible for preemption, the GPU is reclaimed, and the pending pods are granted resources. Pods without any GPU usage or idle pods grant their resources to the pending pods.

NOTE: Workloads with an idle GPU will not be preempted unless there is a pending request from another workload for GPU.

GPU Configurations

In HPE Ezmeral Unified Analytics Software, you can configure the *priority level* and *idle time threshold* from the **GPU Control Panel** screen. However, you cannot configure the *toleration* seconds and *GPU usage threshold* for workloads.

To learn more about the GPU control panel, see Configuring GPU Idle Reclaim on page 150.

Priority class and priority level

HPE Ezmeral Unified Analytics Software attaches priority classes as pod specs to the deployed pods to prioritize pods. The *priority class* has a number called *priority level* that determines the importance of a pod.

The custom scheduler determines the priority based on this priority level. The default *priority level* for all pods is 8000.

You can set the priority level from 8000 to 10000 where 8000 is the lowest priority level and 10000 is the highest priority level. You can update the priority level for your applications and workloads from the GPU Control Panel screen.

Idle time threshold

You can also set the *idle time threshold* for a GPU from the GPU Control Panel screen. The *idle time threshold* is the maximum amount of time the GPU can remain idle without running any workloads. If a GPU remains idle for a duration exceeding this threshold, the GPU on those workloads can be reclaimed to make the GPU available for other workloads.

Toleration seconds

Toleration seconds is the minimum number of seconds the pod or workload needs to run before it can be

	preempted. The default toleration seconds is set to 300 seconds.
GPU usage threshold	The *GPU usage threshold* is the level of GPU utilization. The default usage threshold is set to 0.0. If any pod has a GPU usage of greater than 0 in the last 300 seconds, it cannot be preempted. For any pods to be preempted, the usage must be 0.0.

Configuring GPU Idle Reclaim

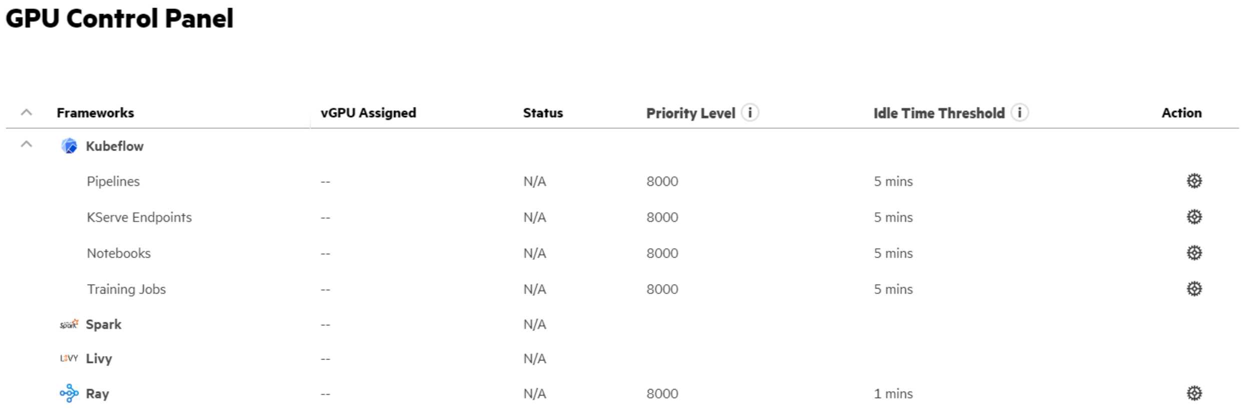
Describes how to configure the GPU idle reclaim, view pod details, and view GPU usage.

You can view frameworks, the number of vGPUs assigned, framework status, priority level, and the idle time threshold in the **GPU Control Panel** screen. You can also view the pod details and the GPU utilization chart.

To navigate to the **GPU Control Panel** screen,

1. Sign in to HPE Ezmeral Unified Analytics Software as Administrator.
2. In the left navigation bar, click **Administration Resource Management**.

You are now in the **GPU Control Panel** screen.



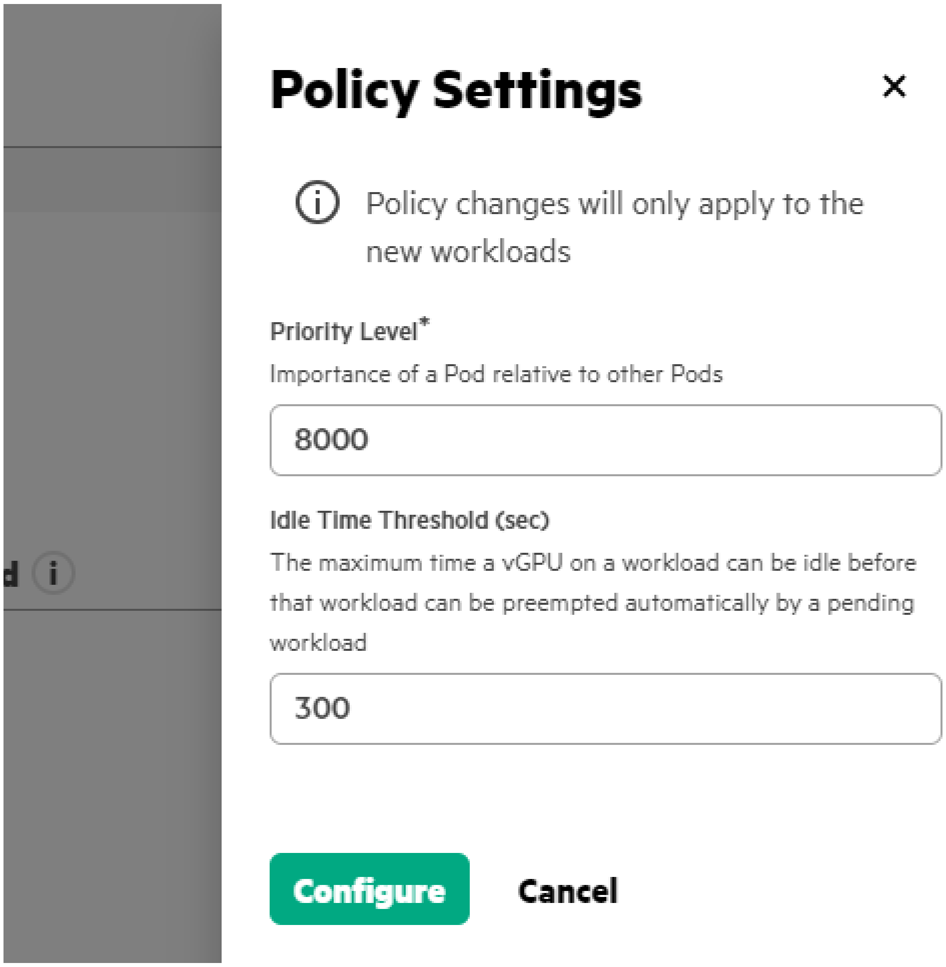
GPU Control Panel

Frameworks	vGPU Assigned	Status	Priority Level	Idle Time Threshold	Action
Kubeflow					
Pipelines	--	N/A	8000	5 mins	⚙
KServe Endpoints	--	N/A	8000	5 mins	⚙
Notebooks	--	N/A	8000	5 mins	⚙
Training Jobs	--	N/A	8000	5 mins	⚙
Spark	--	N/A			
Livy	--	N/A			
Ray	--	N/A	8000	1 mins	⚙

In this screen, you can configure the policy settings, view the pod details and GPU usage as follows:

Configuring the Policy Settings

To set the policy settings (priority level and idle time threshold) for your framework and workload, click the **Actions** menu.



Policy Settings

Policy changes will only apply to the new workloads

Priority Level*

Importance of a Pod relative to other Pods

8000

Idle Time Threshold (sec)

The maximum time a vGPU on a workload can be idle before that workload can be preempted automatically by a pending workload

300

In the **Policy Settings** screen, set the following boxes:

Priority Level	Set the priority level in the range of 8000-10000 where 8000 is the lowest priority and 10000 is the highest priority. For example, a pod with the 8000 priority level will have a low priority compared to the pod with the 10000 priority level. • Default priority level: 8000
Idle Time Threshold	Set the maximum amount of time a vGPU on a workload can be idle before that workload can be preempted (deallocated) automatically by a pending workload. • Minimum idle time threshold: 60 seconds • Default idle time threshold: 300 seconds

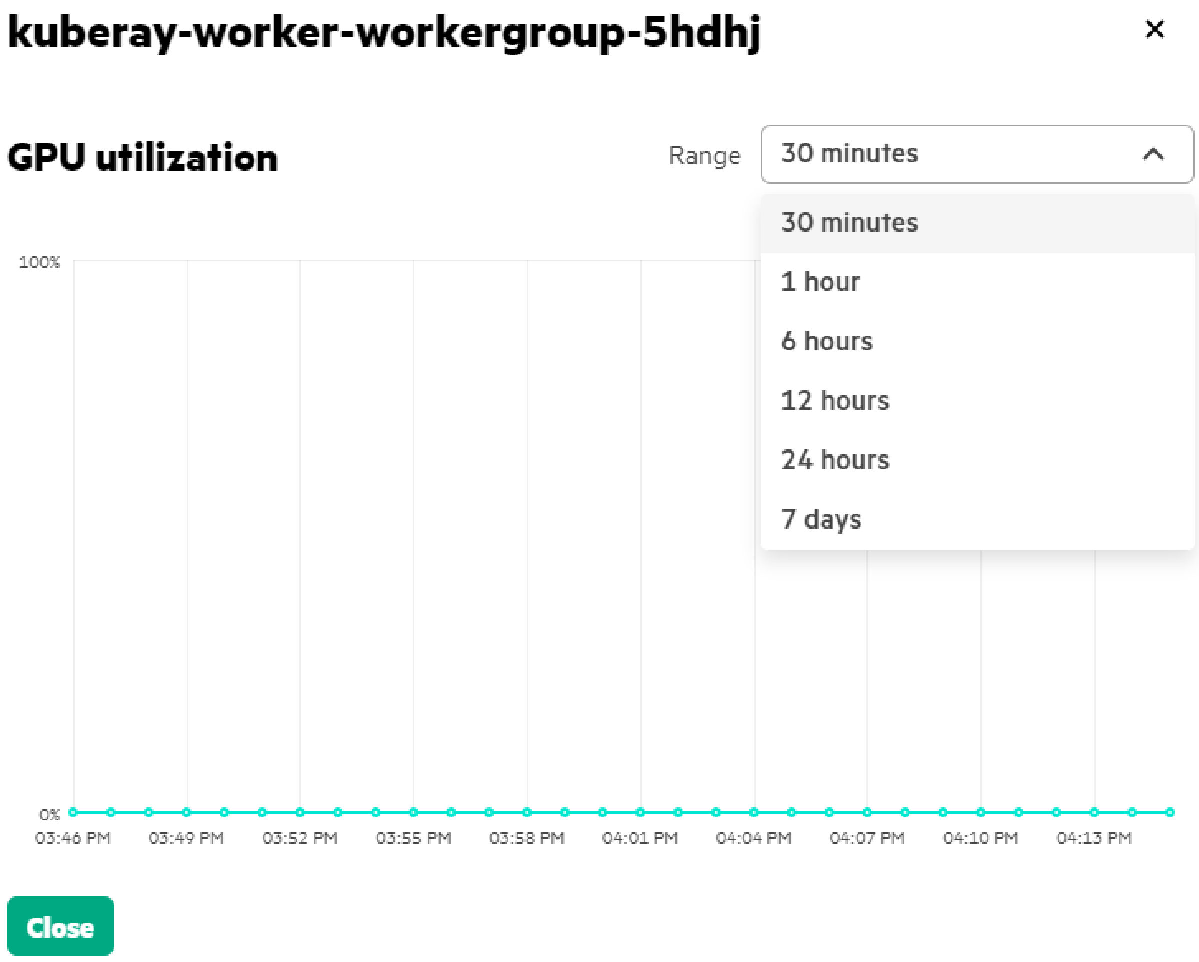
The new policy settings will not be applied to the pods that are currently in the **Running** or **Idle** status. These new policy settings will be applied to the new workloads.

Viewing the Pod Details

To view the pod details, click frameworks that are in the Idle or Running status. This will open a pod detail screen. Here, you can see a list of pods, vGPU assigned, status, age of pods, and the GPU utilization chart.

Viewing the GPU Usage

To view the GPU usage, click the **GPU utilization chart** icon under **Actions**. In the **GPU utilization screen**, you can view the GPU usage for the selected period.



kuberay-worker-workergroup-5hdhj

GPU utilization

Range: 30 minutes

GPU Scheduling Workload Scenarios

Describes GPU scheduling workload scenarios and the notebook example for GPU idle reclaim.

In HPE Ezmeral Unified Analytics Software, you can encounter the following GPU scheduling workload scenarios during the GPU idle reclamation.

GPU Idle Reclaim

In HPE Ezmeral Unified Analytics Software, consider two GPU workloads, denoted as `Workload1` and `Workload2`. Currently, `Workload1` is running and is in an idle state while `Workload2` is pending due to lack of available GPU resources. In this scenario, if the idle duration of `Workload1` exceeds an idle time threshold, `Workload1` is preempted in favor of `Workload2`. Following the preemption, `Workload1` goes into a pending state, while `Workload2` is allocated GPU resources and starts running.

Active GPU Usage

In HPE Ezmeral Unified Analytics Software, consider two GPU workloads, denoted as `Workload1` and `Workload2`. Currently, `Workload1` is running and is using GPU resources while `Workload2` is pending due to lack of available GPU resources. The custom scheduler runs a cron job every 5-10 minutes to determine the eligibility of reclaiming pods based on their GPU usage and the annotation values set in the priority class attached to the pod.

If the GPU usage for `Workload1` is greater than 0.0, `Workload1` cannot be preempted in favor of `Workload2`. In this scenario, `Workload1` will continue to run and utilize the GPU resources without interruption.

If the GPU usage for `Workload1` is equal to 0.0 and if the idle duration of `Workload1` exceeds an idle time threshold, `Workload1` is preempted in favor of `Workload2`. Following the preemption, `Workload1` goes into a pending state, while `Workload2` is allocated GPU resources and starts running.

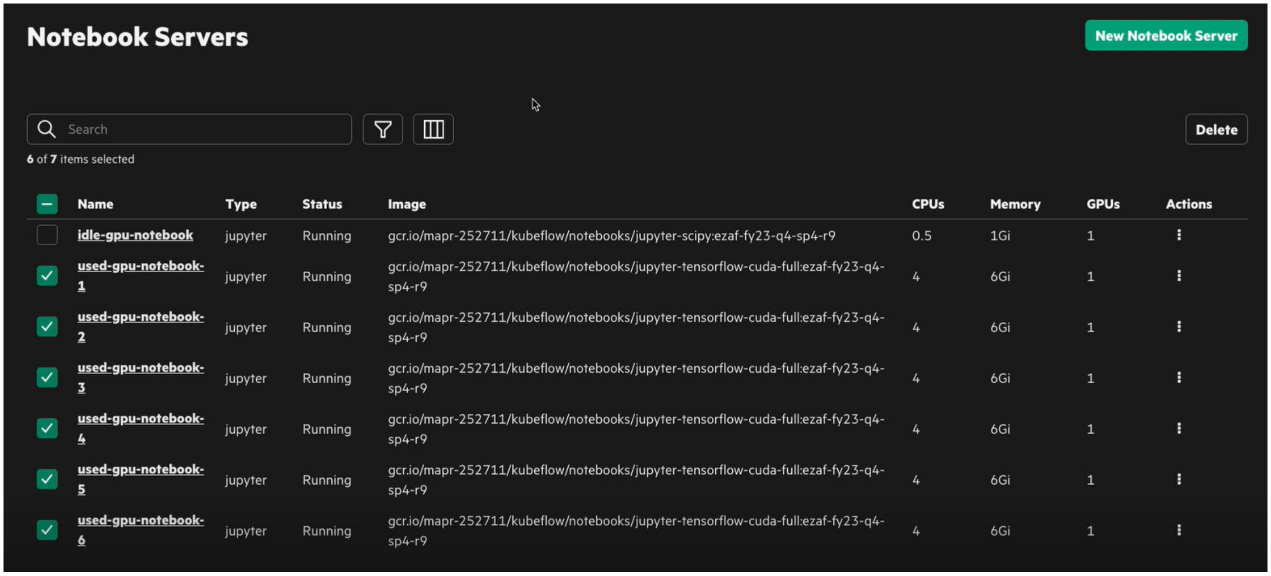
Priority Scheduling

In HPE Ezmeral Unified Analytics Software, consider three GPU workloads, denoted as `Workload1`, `Workload2`, and `Workload3`. Currently, `Workload1` is running and is in an idle state, `Workload2` is pending due to lack of available GPU resources, and `Workload3` has the highest priority among the three workloads and is pending due to lack of available GPU resources. In this scenario, if the idle duration of `Workload1` exceeds an idle time threshold, `Workload1` is preempted in favor of `Workload3`. Following the preemption, `Workload1` goes into a pending state, `Workload3` is allocated GPU resources and starts running, and `Workload2` will continue to be in the pending state.

Notebook Example for GPU Idle Reclaim

Consider a scenario in which HPE Ezmeral Unified Analytics Software is configured with a single physical GPU. In this scenario, you have chosen the small vGPU size, which includes 7 vGPUs. Each application will always have a maximum of one vGPU assigned to it.

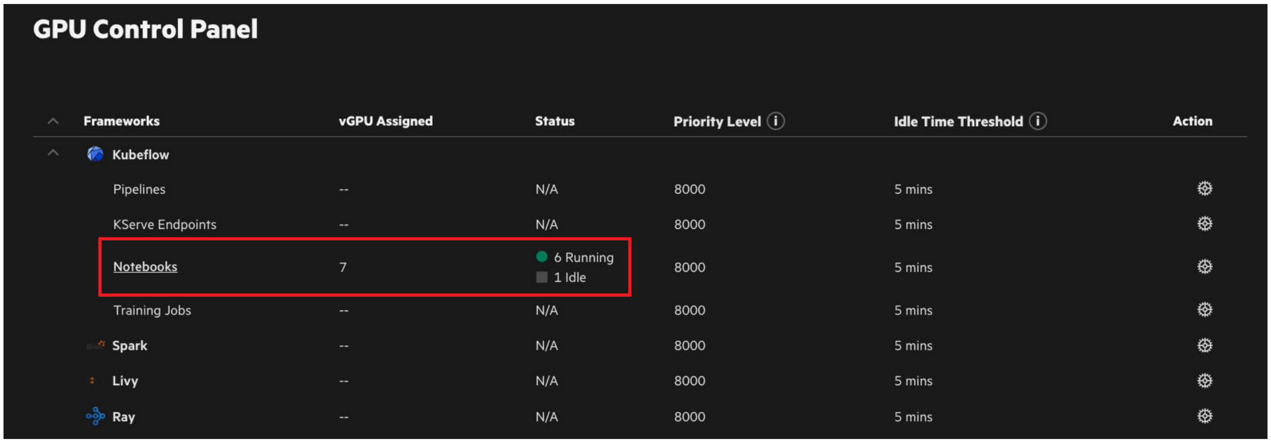
Now, assume you have seven notebook servers, denoted as `idle-gpu-notebook`, `used-gpu-notebook-1`, `used-gpu-notebook-2`, `used-gpu-notebook-3`, `used-gpu-notebook-4`, `used-gpu-notebook-5`, and `used-gpu-notebook-6`. In this scenario, the `idle-notebook-gpu` notebook server has an idle GPU with no GPU usage while the six other notebook servers are actively using GPU resources.



6 of 7 items selected

☐	Name	Type	Status	Image	CPUs	Memory	GPUs	Actions
☐	idle-gpu-notebook	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezaf-fy23-q4-sp4-r9	0.5	1Gi	1	⋮
☑	used-gpu-notebook-1	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	4	6Gi	1	⋮
☑	used-gpu-notebook-2	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	4	6Gi	1	⋮
☑	used-gpu-notebook-3	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	4	6Gi	1	⋮
☑	used-gpu-notebook-4	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	4	6Gi	1	⋮
☑	used-gpu-notebook-5	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	4	6Gi	1	⋮
☑	used-gpu-notebook-6	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	4	6Gi	1	⋮

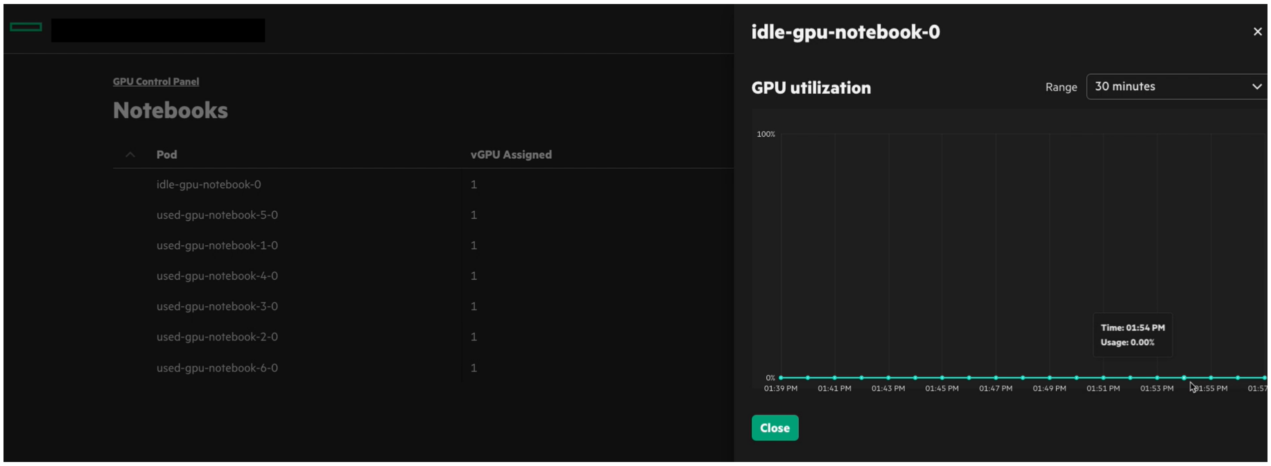
You can navigate to the **GPU Control Panel** screen to check the status of these notebook servers. There you can see that one notebook server has an `Idle` status and the six others have a `Running` status.



GPU Control Panel

Frameworks	vGPU Assigned	Status	Priority Level	Idle Time Threshold	Action
Kubeflow					
Pipelines	--	N/A	8000	5 mins	⚙
KServe Endpoints	--	N/A	8000	5 mins	⚙
Notebooks	7	● 6 Running ■ 1 Idle	8000	5 mins	⚙
Training Jobs	--	N/A	8000	5 mins	⚙
Spark	--	N/A	8000	5 mins	⚙
Livy	--	N/A	8000	5 mins	⚙
Ray	--	N/A	8000	5 mins	⚙

You can click **Notebooks** to view the details of each notebook server. You can confirm that the idle notebook has no GPU usage, and six others have an active GPU usage by clicking the **GPU utilization chart** icon in the **Actions** menu.



[redacted]

GPU Control Panel

Notebooks

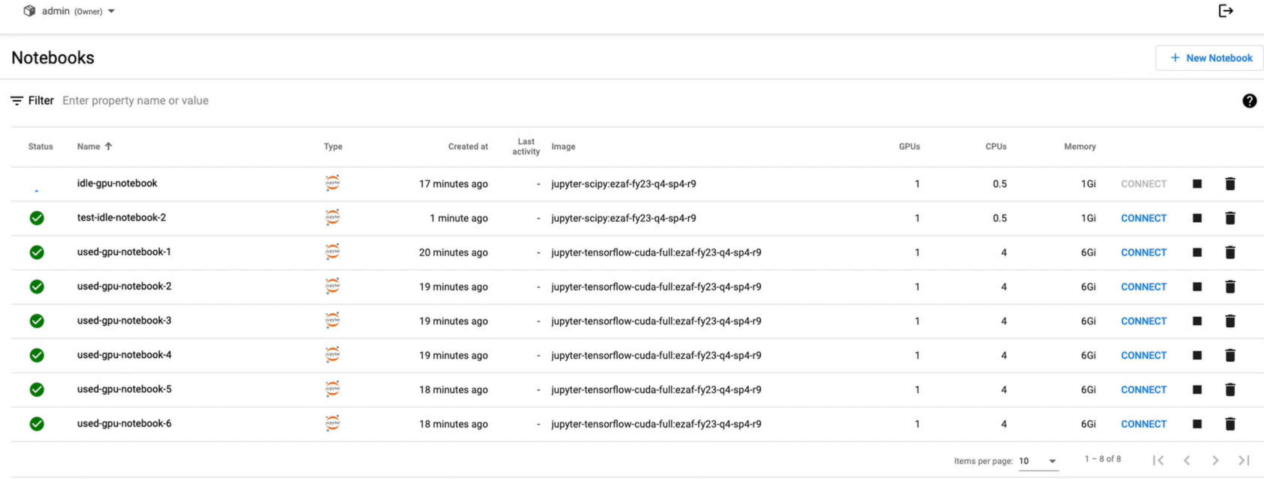
Pod	vGPU Assigned
idle-gpu-notebook-0	1
used-gpu-notebook-5-0	1
used-gpu-notebook-1-0	1
used-gpu-notebook-4-0	1
used-gpu-notebook-3-0	1
used-gpu-notebook-2-0	1
used-gpu-notebook-6-0	1

idle-gpu-notebook-0

GPU utilization

Range: 30 minutes

Consider creating another GPU-enabled notebook server, denoted as `test-idle-notebook-2`. As the GPU usage for `idle-gpu-notebook` is equal to 0.0, as soon as the idle duration of `idle-gpu-notebook` exceeds an idle time threshold, `idle-gpu-notebook` is preempted in favor of `test-idle-notebook-2`. Following the preemption, `idle-gpu-notebook` goes into a pending state, while `test-idle-notebook-2` is allocated GPU resources and starts running.



Notebooks

Status	Name ↑	Type	Created at	Last activity	Image	GPUs	CPUs	Memory			
	idle-gpu-notebook	jupyter	17 minutes ago	-	jupyter-scipy:ezaf-fy23-q4-sp4-r9	1	0.5	1Gi	CONNECT	■	🗑
✓	test-idle-notebook-2	jupyter	1 minute ago	-	jupyter-scipy:ezaf-fy23-q4-sp4-r9	1	0.5	1Gi	CONNECT	■	🗑
✓	used-gpu-notebook-1	jupyter	20 minutes ago	-	jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	1	4	6Gi	CONNECT	■	🗑
✓	used-gpu-notebook-2	jupyter	19 minutes ago	-	jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	1	4	6Gi	CONNECT	■	🗑
✓	used-gpu-notebook-3	jupyter	19 minutes ago	-	jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	1	4	6Gi	CONNECT	■	🗑
✓	used-gpu-notebook-4	jupyter	19 minutes ago	-	jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	1	4	6Gi	CONNECT	■	🗑
✓	used-gpu-notebook-5	jupyter	18 minutes ago	-	jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	1	4	6Gi	CONNECT	■	🗑
✓	used-gpu-notebook-6	jupyter	18 minutes ago	-	jupyter-tensorflow-cuda-full:ezaf-fy23-q4-sp4-r9	1	4	6Gi	CONNECT	■	🗑

Items per page: 10 1 – 8 of 8

Troubleshooting

Describes how to identify and debug issues in HPE Ezmeral Unified Analytics Software.

To run `kubectl` commands and perform the admin-related tasks described in these topics, sign in to HPE Ezmeral Unified Analytics Software as an administrator.

Airflow

Describes how to identify and debug issues for Airflow.

Airflow UI

Cannot access Airflow UI or cannot see DAGs.

- Ensure that the Git repository is configured properly. See Airflow DAGs Git Repository on page 355.
- The administrator can refer to the logs from the `git-sync` container in the `scheduler` pod in the `airflow-hpe` namespace.

Cannot sign in to Airflow or other issues in Airflow UI.

Check the logs from the `af-cluster-airflowui-0` pod in the `airflow-hpe` namespace. Run:

```
kubectl logs -n airflow-hpe
af-cluster-airflowui-0
```

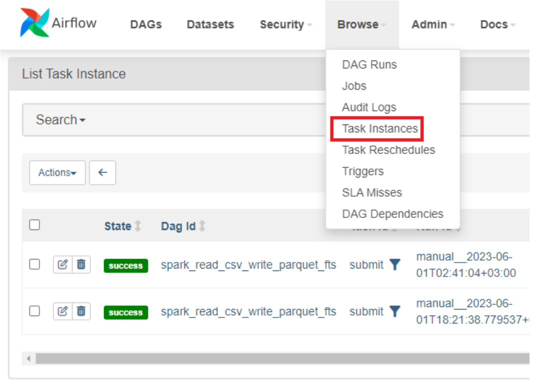
NOTE: If more than one user needs to access the same browser, the logged-in user must explicitly log out before another user can access the UI. Failure to explicitly log out results in caching and dashboard permission issues if multiple users try to access the same UI.

Airflow DAG

Airflow DAG is failing.

If Airflow DAG is failing, you can check the logs in the following three ways:

- To check the logs of the failed task in the Airflow UI page, follow these steps:
  1. Sign in to HPE Ezmeral Unified Analytics Software.
  2. Click the **Applications & Frameworks** icon on the left navigation bar. Navigate to the **Airflow** tile under the **Data Engineering** tab and click **Open**.
  3. Click **Browse** and select **Task Instances**.



State	Dag Id		
success	spark_read_csv_write_parquet_fts	submit	manual__2023-06-01T02:41:04+03:00
success	spark_read_csv_write_parquet_fts	submit	manual__2023-06-01T18:21:38.779537+

4. Select the failed task from the list.

5. Scroll horizontally to the right until you find the **Log Url** button.

6. Click on the **Log Url** button to view the logs associated with the failed task.

- To check the logs from the pod of a task by its name in the `airflow-hpe` namespace, run:

```
kubectl logs -n airflow-hpe
<pod_name_associated_with_the_task>
```

- To check the logs from the `scheduler` pod in the `airflow-hpe` namespace, run:

```
kubectl logs -n airflow-hpe
af-cluster-scheduler-0
```

Airflow `scheduler` Pod

The `scheduler` pod is not coming up.

If the `scheduler` pod is not coming up, follow these steps:

NOTE: Performing the next steps will result in the deletion of Airflow metadata. Proceed with caution.

1. Delete the PVC in the `airflow-hpe` namespace without waiting for the deletion.

```
kubectl delete pvc -n airflow-hpe
<pvc_name>
```

2. Delete the PostgreSQL database StatefulSet in the `airflow-hpe` namespace.

```
kubectl delete
statefulset -n airflow-hpe
<postgres_db_statefulset_name>
```

3. Restart the `scheduler` pod.

   ```
   kubectl rollout restart sts -n
   airflow-hpe af-cluster-scheduler
   ```

EzPresto

Describes how to identify and debug issues for EzPresto .

Cannot create Iceberg connections with hadoop catalog type from the UI

HPE Ezmeral Unified Analytics Software supports Iceberg connections with the hadoop catalog type. However, you cannot create the Iceberg connection through the HPE Ezmeral Unified Analytics Software UI. You must create the connection from the command line using a `curl` command with a JSON configuration.

To create an Iceberg connection with catalog type hadoop:

1. Create the JSON configuration for your storage type, replacing all values in angle brackets (<>) with values for your environment:

   IMPORTANT:

   - For the `iceberg.hadoop.config.resources` property, you must upload the XML configurations file and then click **Encode file to Base64**. Copy the encoded string and paste it as the value for `iceberg.hadoop.config.resources`.
   - If you run the `curl` command from a Unified Analytics 1.5.2 cluster, you must include a refresh token in the JSON configuration. To get a refresh token, go to the HPE Ezmeral Unified Analytics Software refresh token URL to download a refresh token. The following example shows the URL format to use to get a refresh token:

     ```
     https://token-service.<your-ua-cluster-domain>/
     refresh-token-download
     ```

   - **HPE Ezmeral Data Fabric Object Store**

     ```
     {
         "catalogName": "<catalog_name>",
         "connectorName": "iceberg",
         "properties": {
             "iceberg.catalog.type": "hadoop",
             "iceberg.catalog.warehouse": "<S3 Warehouse Location>",
             "iceberg.catalog.cached-catalog-num": "10",
             "hive.s3.aws-access-key": "<S3 Access key>",
             "hive.s3.aws-secret-key": "<S3 Secret Key>",
             "hive.s3.endpoint": "<S3 End Point>",
             "hive.s3.path-style-access": true,
             "hive.s3.ssl.enabled": false
         },
         "fileProperties": {
             "iceberg.hadoop.config.resources": [
                 "<Base64 encoded string for config resource file (e.g.
     hdfs-site.xml)>"
             ]
         }
     }
     ```

- **HPE Ezmeral Data Fabric File Store**

  ```
  {
    "catalogName": "<catalog_name>",
    "connectorName": "iceberg",
    "properties": {
      "iceberg.catalog.type": "hadoop",
      "hive.hdfs.authentication.type": "MAPRSASL",
      "df.cluster.details": "<DF Cluster Details>",
      "hive.hdfs.df.ticket":"<DF Cluster Ticker Details>",
      "iceberg.catalog.warehouse": "<MAPR FS Warehouse Location>"
    },
    "fileProperties": {
      "iceberg.hadoop.config.resources": [
        "<Base64 encoded string for config resource file (e.g.
  hdfs-site.xml)>"
      ]
    }
  }
  ```

- **Local or mounted file system that is locally accessible**

  ```
  {
      "catalogName": "<catalog_name>",
      "connectorName": "iceberg",
      "properties": {
          "iceberg.catalog.type": "hadoop",
          "iceberg.catalog.warehouse": "<Locally Mounted Warehouse
  Location>"
      }
  }
  ```

2. To call the EzPresto backend API and post the JSON, run:

   ```
   curl -u <username>:<password> --location '<EzPresto Endpoint>/v1/
   catalog' --header 'Content-Type: application/json' --insecure --data
   '<JSON DATA>'

   //<username>:<password> (Replace with your Unified Analytics username
   and password.)
   //<EZPresto End Point> (Go to Tools&Frameworks>Data Engineering>EzPresto
   and copy the endpoint URL).
   //<JSON DATA> (Enter the JSON config from the previous step.)
   ```

   NOTE: You can run the `curl` command from any machine that can access the Unified Analytics cluster endpoint, such as `https://<your-ua-cluster-domain>.com/v1/catalog`.

   You should now see the Iceberg connection in Unified Analytics by going to **Data Engineering > Data Sources** and clicking on tab that correlates with the data source type, such as **Structured Data**.

EzPresto installation fails due to mysql pod entering CrashLoopBackOff state

During EzPresto deployment, the HPE Ezmeral Unified Analytics Software installation fails due to slow disk I/O, which leads to the mysql pod in EzPresto entering a CrashLoopBackOff state.

When the mysql pod is deployed, a lifecycle hook expects the pod to be ready within thirty seconds. If the pod is not ready within thirty seconds, Kubernetes continuously tries to restart the pod which leads to the pod being in a CrashLoopBackOff state.

To resolve this issue, complete the following steps:

1. Stop the mysql pod:

   ```
   kubectl scale deployment ezpresto-dep-mysql --replicas=0 -n ezpresto
   ```

2. Edit the mysql deployment:

   ```
   kubectl edit deployment ezpresto-dep-mysql -n ezpresto
   ```

3. Remove the following lifecycle hook:

   ```
   lifecycle:
             postStart:
               exec:
                 command:
                   - "sh"
                   - "-c"
                   - >
                     sleep 30 ;
                     mysql -u root -p$MYSQL_ROOT_PASSWORD -e "GRANT ALL
   PRIVILEGES ON *.* TO '$MYSQL_USER'@'%' WITH GRANT OPTION";
   ```

4. Delete the mysql pvc:

   ```
   kubectl delete pvc ezpresto-pvc-mysql -n ezpresto
   ```

5. Create a file named `mysql.pvc` and copy the following content into the file:

   ```
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     annotations:
       meta.helm.sh/release-name: ezpresto
       meta.helm.sh/release-namespace: ezpresto
       volume.beta.kubernetes.io/storage-provisioner: com.mapr.csi-kdf
       volume.kubernetes.io/storage-provisioner: com.mapr.csi-kdf
     labels:
       app.kubernetes.io/managed-by: Helm
     name: ezpresto-pvc-mysql
     namespace: ezpresto
   spec:
     accessModes:
     - ReadWriteMany
     resources:
       requests:
         storage: 5Gi
     storageClassName: edf
     volumeMode: Filesystem
   ```

6. Create a mysql pvc:

   ```
   kubectl apply -f mysql.pvc -n ezpresto
   ```

7. Start the mysql pods:

   ```
   kubectl scale deployment ezpresto-dep-mysql --replicas=1 -n ezpresto
   ```

8. Restart the web service pods:

   ```
   kubectl rollout restart deployment ezpresto-dep-web -n ezpresto
   ```

Installation is complete and you can use EzPresto once all pods in the `ezpresto` namespace are running.

Trying to Access a Hive Directory Results in an Access Denied Error

Any schema created with impersonation returns an access denied error if the directory ownership is not set correctly for the impersonating user. To avoid access denied errors, correct the ownership/permissions on the directory before performing any operations:

```
hadoop fs [-chown [-R] [OWNER][:[GROUP]] PATH...]
hadoop fs [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
```

For example, SSH in to the HPE Ezmeral Data Fabric cluster node. If the `mapr` user ticket was used for hive impersonation, then it should be used for following operation:

```
export MAPR_TICKETFILE_LOCATION=/home/bob123/mapruserticket
hadoop fs -chown bob123:ldap maprfs://user01/user/hive/warehouse/foo.db
hadoop fs -chmod 775 maprfs://user01/user/hive/warehouse/foo.db
```

Cannot Add Iceberg as a Data Source when Catalog Type is Hadoop

Recent changes introduced by open source PrestoDB cause Iceberg data connections to fail in Unified Analytics when the Catalog Type is Hadoop.

Workaround for New Installation

To connect Unified Analytics to an Iceberg data source with Catalog Type set as Hadoop, complete the following steps:

1. To update the EzPresto images, run the following `kubectl` commands:

   ```
   kubectl set image statefulset/
   ezpresto-sts-mst
   presto-coordinator=marketplace.us1.
   greenlake-hpe.com/ezua/gcr.io/
   mapr-252711/ezsql-test/
   presto-0.285-fy24-q2:0.0.61 --names
   pace=ezpresto

   kubectl set image statefulset/
   ezpresto-sts-wrk
   presto-worker=marketplace.us1.green
   lake-hpe.com/ezua/gcr.io/
   mapr-252711/ezsql-test/
   presto-0.285-fy24-q2:0.0.61 --names
   pace=ezpresto
   ```

2. Sign in to Unified Analytics and add the Iceberg data source with the Catalog Type set as Hadoop.

Workaround for Upgrade

If you want to upgrade Unified Analytics from version 1.3 to 1.4, and you have an Iceberg data source in place with Catalog Type set as Hadoop, complete the following steps:

1. Sign in to Unified Analytics.
2. Delete the Iceberg connection.
3. Upgrade to Unified Analytics version 1.4.
4. To update the EzPresto images, run the following `kubectl` commands:

   ```
   kubectl set image statefulset/
   ezpresto-sts-mst
   presto-coordinator=marketplace.us1.
   greenlake-hpe.com/ezua/gcr.io/
   mapr-252711/ezsql-test/
   presto-0.285-fy24-q2:0.0.61 --names
   pace=ezpresto

   kubectl set image statefulset/
   ezpresto-sts-wrk
   presto-worker=marketplace.us1.green
   lake-hpe.com/ezua/gcr.io/
   mapr-252711/ezsql-test/
   presto-0.285-fy24-q2:0.0.61 --names
   pace=ezpresto
   ```

5. Sign in to Unified Analytics and add the Iceberg data source with the Catalog Type set as Hadoop.

Insufficient Memory

Currently, the maximum memory available to queries is based on the memory resources of a single worker node instead of total cluster memory (all worker nodes). As a result, queries may fail due to insufficient memory. To address this issue, modify the EzPresto configuration as described in the following steps:

1. In the left navigation bar, go to **Tools & Frameworks > Data Engineering > EzPresto**.
2. Click on the three dots and select **Configure**.

3. In window that appears, remove the entire `cmnConfigMaps` section and replace it with the following:

```
cmnConfigMaps:
  # Configmaps common to both Presto Master and Worker
  logConfig:
    log.properties: |
      # Enable verbose logging from Presto
      #com.facebook.presto=DEBUG

  # Configmaps specific to Presto Master
  prestoMst:
    cmnPrestoCoordinatorConfig:
      config.properties: |

http-server.http.port={{ tpl .Values.ezsqlPresto.locatorService.locatorSv
cPort $ }}

discovery.uri=http://{{ tpl .Values.ezsqlPresto.locatorService.fullname
$ }}:{{ tpl .Values.ezsqlPresto.locatorService.locatorSvcPort $ }}
        coordinator=true
        node-scheduler.include-coordinator=false
        discovery-server.enabled=true
        catalog.config-dir =
{{ .Values.ezsqlPresto.stsDeployment.volumeMount.mountPathCatalog }}

catalog.disabled-connectors-for-dynamic-operation=drill,parquet,csv,sales
force,sharepoint,prestodb,raptor,kudu,redis,accumulo,elasticsearch,redshi
ft,localfile,bigquery,prometheus,mongodb,pinot,druid,cassandra,kafka,atop
,presto-thrift,ampool,hive-cache,memory,blackhole,tpch,tpcds,system,examp
le-http,jmx
        generic-cache-enabled=true
        transparent-cache-enabled=false
        generic-cache-catalog-name=cache
        generic-cache-change-detection-interval=300
        catalog.config-dir.shared=true
        node.environment=production
        plugin.dir=/usr/lib/presto/plugin
        log.output-file=/data/presto/server.log
        log.levels-file=/usr/lib/presto/etc/log.properties
        query.max-history=1000
        query.max-stage-count=1000
        query.max-memory={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        query.max-total-memory={{ mulf 0.7
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        # query.max-memory-per-node={{ mulf 0.5
( tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapSize . ) |
floor }}MB
        # query.max-total-memory-per-node={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapSize . ) |
floor }}MB
        # memory.heap-headroom-per-node={{ mulf 0.3
( tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapSize . ) |
floor }}MB
        experimental.spill-enabled=false
        experimental.spiller-spill-path=/tmp
        orm-database-url=jdbc:sqlite:/data/cache/metadata.db

plugin.disabled-connectors=accumulo,atop,cassandra,example-http,kafka,kud
u,localfile,memory,mongodb,pinot,presto-bigquery,prestodb,presto-druid,pr
esto-elasticsearch,prometheus,raptor,redis,redshift
```

```
        log.max-size=100MB
        log.max-history=10
        discovery.http-client.max-requests-queued-per-destination=10000
        dynamic.http-client.max-requests-queued-per-destination=10000
        event.http-client.max-requests-queued-per-destination=10000
        exchange.http-client.max-requests-queued-per-destination=10000

failure-detector.http-client.max-requests-queued-per-destination=10000

memoryManager.http-client.max-requests-queued-per-destination=10000

node-manager.http-client.max-requests-queued-per-destination=10000
        scheduler.http-client.max-requests-queued-per-destination=10000
        workerInfo.http-client.max-requests-queued-per-destination=10000

  # Configmaps specific to Presto Worker
  prestoWrk:
    prestoWorkerConfig:
      config.properties: |
        coordinator=false

http-server.http.port={{ tpl .Values.ezsqlPresto.locatorService.locatorSv
cPort $ }}

discovery.uri=http://{{ tpl .Values.ezsqlPresto.locatorService.fullname
$ }}:{{ tpl .Values.ezsqlPresto.locatorService.locatorSvcPort $ }}
        catalog.config-dir =
{{ .Values.ezsqlPresto.stsDeployment.volumeMount.mountPathCatalog }}

catalog.disabled-connectors-for-dynamic-operation=drill,parquet,csv,sales
force,sharepoint,prestodb,raptor,kudu,redis,accumulo,elasticsearch,redshi
ft,localfile,bigquery,prometheus,mongodb,pinot,druid,cassandra,kafka,atop
,presto-thrift,ampool,hive-cache,memory,blackhole,tpch,tpcds,system,examp
le-http,jmx
        generic-cache-enabled=true
        transparent-cache-enabled=false
        generic-cache-catalog-name=cache
        catalog.config-dir.shared=true
        node.environment=production
        plugin.dir=/usr/lib/presto/plugin
        log.output-file=/data/presto/server.log
        log.levels-file=/usr/lib/presto/etc/log.properties
        query.max-memory={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        query.max-total-memory={{ mulf 0.7
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        query.max-memory-per-node={{ mulf 0.5
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . ) |
floor }}MB
        query.max-total-memory-per-node={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . ) |
floor }}MB
        memory.heap-headroom-per-node={{ mulf 0.2
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . ) |
floor }}MB
        experimental.spill-enabled=false
        experimental.spiller-spill-path=/tmp
        orm-database-url=jdbc:sqlite:/data/cache/metadata.db

plugin.disabled-connectors=accumulo,atop,cassandra,example-http,kafka,kud
u,localfile,memory,mongodb,pinot,presto-bigquery,prestodb,presto-druid,pr
esto-elasticsearch,prometheus,raptor,redis,redshift
```

```
        log.max-size=100MB
        log.max-history=10
        discovery.http-client.max-requests-queued-per-destination=10000
        event.http-client.max-requests-queued-per-destination=10000
        exchange.http-client.max-requests-queued-per-destination=10000

node-manager.http-client.max-requests-queued-per-destination=10000
        workerInfo.http-client.max-requests-queued-per-destination=10000
### values_cmn_configmap.yaml contents END
```

4. Click **Configure** to update the configuration on each of the presto pods and restart the pods. This operation takes a few minutes.

If this workaound does not resolve the issue, contact HPE Support.

Failed Queries

If queries fail, go to the Presto UI and view the stack trace for the queries. You can also view the EzPresto log files.

You can access the Presto UI from the HPE Ezmeral Unified Analytics Software UI.

1. In the left navigation bar, select **Tools & Frameworks**.
2. Select the **Data Engineering** tab.
3. In the **EzPresto** tile, click on the **Endpoint** URL.
4. In the Presto UI, select the **Failed** state.
5. Locate the query and click on the **Query ID**.
6. Scroll down to the **Error Information** section to view the stack trace.

You can also view the logs in the shared directory.

1. In the left navigation bar, select **Data Engineering > Data Sources**.
2. On the Data Sources screen, click **Browse**.
3. Select the following directories in the order shown:
   a. shared/
   b. logs/
   c. apps/
   d. app-core/
   e. ezpresto/
4. Select the log directory for which you want to view **EzPresto** logs.

Hive Data Source Connection Failure (S3-Based External Data Souce)

The following sections describe some issues that can cause Hive connection failures when using Hive to connect to an external s3-based data source, such as HPE Ezmeral Data Fabric Object Store. A workaround is provided for each issue.

Files have 0 length	The folder that contains the CSV or Parquet files has files with 0 length. For example, the files are empty or they are like the files generated by Spark jobs (_SUCCESS). **Workaround:** Remove the empty files.
CSV file with an empty line	A CSV file has an empty line either in the data or in the last line of the file. **Workaround:** Remove the empty lines in the file.
S3 folder with incorrect MIME type	The S3 folder that contains the CSV and Parquet files was created through the HPE Ezmeral Data Fabric Object Store UI. In pre-1.3 versions of HPE Ezmeral Unified Analytics Software, EzPresto does not recognize the folders created through the HPE Ezmeral Data Fabric Object Store UI because the S3 folder MIME type is different than the type set by AWS `s3cmd`. **Workaround:** Use AWS `s3cmd` to create a folder and upload files to a bucket in HPE Ezmeral Data Fabric Object Store, for example, `s3://<bucket>/<folder1>/<folder2>/data.csv`. **NOTE:** You cannot put files directly in the **Data Dir** path that you specified when you created the Hive connection. You must create a folder within the **Data Dir** path that you specified and put files there. For example, if you entered `s3://mytestbucket/` as the **Data Dir**, you must create a folder within that directory, such as `s3://mytestbucket/data/` and put files there.

Data Source Connection Failure (File-Based)

If a file system-based data connection fails, verify that the storage or file location starts with theappropriate scheme, for example `maprfs://`, `hdfs://`, or `file:/`.

GPU

Describes how to identify and debug issues for GPU.

GPU Not Working as Expected

Upload and run Check_gpu_card.ipynb notebook file in GPU-enabled notebook servers. See Creating GPU-Enabled Notebook Servers on page 438.

If the output does not display the GPU card, follow these steps:

1. To access the NVIDIA CLI in the `hpecp-gpu-operator` namespace, run:

   ```
   kubectl exec -it -n hpecp-gpu-operator daemonset/
   nvidia-device-plugin-daemonset -- bash
   ```

2. To show the Python 3 process, run:

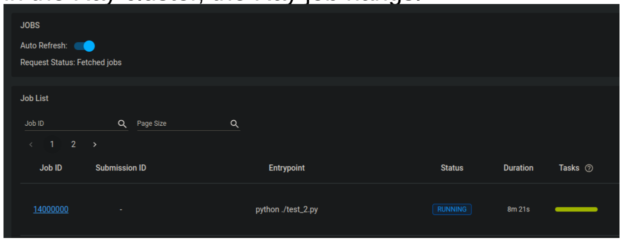
   ```
   nvidia-smi
   ```

   If the output does not show the Python 3 process, contact Hewlett Packard Enterprise support.

Ray

Ray job hangs when you request more than available GPU resource in the Ray cluster.

When you request more than available GPU resource in the Ray cluster, the Ray job hangs.



JOBS

Auto Refresh:

Request Status: Fetched jobs

Job List

Job ID	Submission ID	Entrypoint	Status	Duration	Tasks
14000000	-	python ./test_2.py	RUNNING	8m 21s	

When you go to the logs in Ray Dashboard, you can see the following general log entry. However, this log entry does not specify that the job is hanging as more than available GPU resource is requested.

```
[2023-07-20 08:18:09,674 I 25723
25723] core_worker.cc:651: Waiting
for joining a core worker io thread.
If it hangs here, there might be
deadlock or a high load in the core
worker io service.
```

To confirm that the job hanging has more than the available GPU resource requested, you can perform the following checks:

- Run the following command to get the tasks summary:

  ```
  kubectl -n kuberay exec
  kuberay-head-2dj8n -- ray summary
  tasks
  ```

  Output: When you run the `kubectl` command to check the tasks summary, you can see the job is pending as follows:

  ```
  Defaulted container "ray-head" out
  of: ray-head, autoscaler, init
  (init)
  ======== Tasks Summary: 2023-07-20
  08:15:25.292285 ========
  Stats:
  ------------------------------------

  total_actor_scheduled: 12
  total_actor_tasks: 12
  total_tasks: 192


  Table (group by func_name):
  ------------------------------------

  FUNC_OR_CLASS_NAME

  STATE_COUNTS
  TYPE
  0
  fibonacci_distributed
           FINISHED:
  160                     NORMAL_TASK


  PENDING_NODE_ASSIGNMENT: 32
  1
  RayFraudDetectionExperiment.run_exp
  eriment  FAILED:
  2
  ACTOR_TASK

               FINISHED: 10
  2
  RayFraudDetectionExperiment.__init_
  _        FAILED:
  2
  ACTOR_CREATION_TASK

               FINISHED: 10
  ```

- Run the following command to check the job status:

  ```
  kubectl -n kuberay exec
  kuberay-head-2dj8n -- ray status
  ```

  Output: When you run the `kubectl` command to check the job status, you can see that job hangs until it gets the required resources as follows:

  ```
  Defaulted container "ray-head" out
  of: ray-head, autoscaler, init
  (init)
  ======== Autoscaler status:
  2023-07-20 08:16:04.958109 ========
  Node status
  -----------------------------------------------

  Healthy:
  1 head-group
  1 smallGroup
  1 workerGroup
  Pending:
  no pending nodes)
  Recent failures:
  no failures)

  Resources
  -----------------------------------------------

  Usage:
  0.0/3.0 CPU
  0.0/1.0 GPU
  0B/14.90GiB memory
  0B/4.36GiB object_store_memory

  Demands:
  {'GPU': 2.0}: 32+ pending tasks/
  actors
  ```

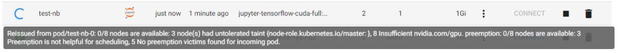
Notebooks

Notebook server creation will be in the pending state when you assign more than one GPU resource.

When you assign more than one GPU resource for notebook servers, the notebook server creation will be in a pending state. If you hover over the spinner, you can see the following message:

Reissued from pod/test-nb-0: 0/8 nodes are available: 3 node(s) had untolerated taint {node-role.kubernetes.io/master: }, 8 Insufficient nvidia.com/gpu. preemption: 0/8 nodes are available: 3 Preemption is not helpful for scheduling, 5 No preemption victims found for incoming pod.

For example:



test-nb just now 1 minute ago jupyter-tensorflow-cuda-full:... 2 1 1Gi

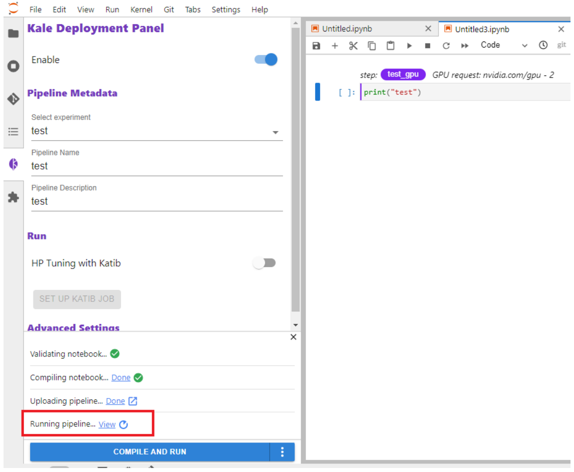
Reissued from pod/test-nb-0: 0/8 nodes are available: 3 node(s) had untolerated taint {node-role.kubernetes.io/master: }, 8 Insufficient nvidia.com/gpu. preemption: 0/8 nodes are available: 3 Preemption is not helpful for scheduling, 5 No preemption victims found for incoming pod.

Kale

The Running Pipeline step will be in the pending state when you assign more than one GPU resource for Kale.

To confirm that the **Running Pipeline** step is in the pending state as more than one GPU resource is assigned for Kale, follow these steps:

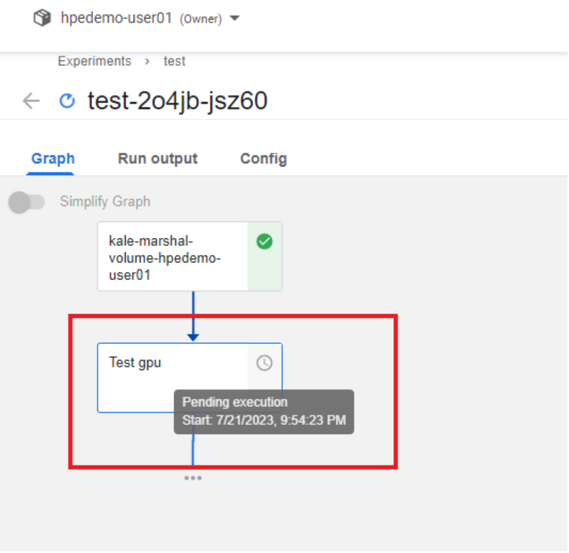
1. Perform the steps to specify the GPU resource in the Kale extension. See Specifying GPU Resources in the Kale Extension on page 444.
2. Run the notebook via Kale.
3. Go to **Running Pipeline** and click **View**. You can see that the pipeline state is in a pending state.



```
[ ]: print("test")
```

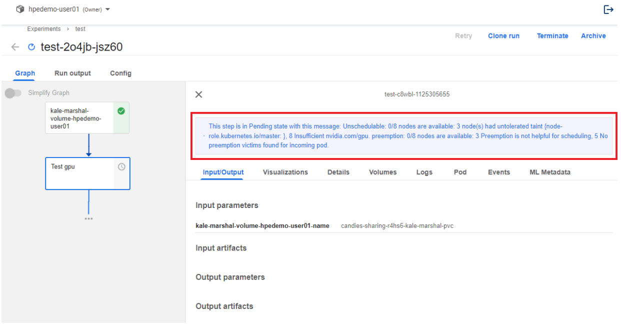
4. Click on the step in the pending state.

   For example: Test gpu is the pending step.



Output: You can see the following message:

This step is in Pending state with this message: Unschedulable: 0/8 nodes are available: 3 node(s) had untolerated taint {node-role.kubernetes.io/master: }, 8 Insufficient nvidia.com/gpu. preemption: 0/8 nodes are available: 3 Preemption is not helpful for scheduling, 5 No preemption victims found for incoming pod.



This step is in Pending state with this message: Unschedulable: 0/8 nodes are available: 3 node(s) had untolerated taint {node-role.kubernetes.io/master: }, 8 Insufficient nvidia.com/gpu. preemption: 0/8 nodes are available: 3 Preemption is not helpful for scheduling, 5 No preemption victims found for incoming pod.

kale-marshal-volume-hpedemo-user01-name candies-sharing-r4hs6-kale-marshal-pvc

Host (Node) Management

Describes how to identify and debug issues for hosts.

Pods Stuck in Terminating State

If you have not updated the SPIFFE CSI driver, as indicated in the Post Installation Steps on page 82, and you encounter pods stuck in the `Terminating` state after restarting, complete the following steps:

Connected Environment

1. Run the following command to update the SPIFFE CSI driver:

   ```
   kubectl -n spire set image
   ds spire-spiffe-csi-driver
   spiffe-csi-driver=ghcr.io/spiffe/
   spiffe-csi-driver:0.2.5
   ```

2. Remove the pods in the `Terminating` state.

Air-Gapped Environment

1. Run the following command to update the SPIFFE CSI driver:

   ```
   kubectl -n spire set image ds
   spire-spiffe-csi-driver
   spiffe-csi-driver=<airgap_registry>
   /ezua/ghcr.io/spiffe/
   spiffe-csi-driver:0.2.5
   ```

2. Remove the pods in the `Terminating` state.

NOTE: If these steps do not resolve the issue, contact HPE Support.

Importing Applications and Managing the Application Lifecycle

Describes how to identify and debug issues while importing applications and managing the application lifecycle.

Downloading the application chart version fails.

If downloading the application chart version fails, verify the chart is present in the chartmuseum repository.

Importing applications results in an error.

If you get errors while importing applications,

- Check the error state in the application tile.
- Check job logs in ezapp-system namespace.

Importing applications after fixing the application charts.	If you need to import the application after fixing the application chart, follow these steps: **1.** Delete the previously imported application. **2.** Update the chart version. **3.** Re-package the application. **4.** Import the re-packaged application.
The Open button within the application tile is not working and the endpoint URL is missing from the tile.	Verify that `values.yaml` file includes the `ezua` section.

Installation

Describes how to identify and debug issues during installation.

HPE Ezmeral Data Fabric as Primary Storage

If any issues related to HPE Ezmeral Data Fabric as primary storage for Unified Analytics occur, inspect the HPE Ezmeral Data Fabric addon and review the bootstrap pod logs to identify the cause of the issue.

- To inspect the HPE Ezmeral Data Fabric addon (EzAddonDeployment), run:

```
kubectl describe -n ezaddon-system ezaddondeployment storage-edf
```

- To review the bootstrap pod logs, run:

```
kubectl logs -n ezaddon-system deploy/hpecp-bootstrap-storage-edf
```

The following table lists log message examples and the reason for each message:

TIP: You can resolve all issues in the following table by following the steps in Preparing HPE Ezmeral Data Fabric to be Primary Storage for HPE Ezmeral Unified Analytics Software on page 78.

Cause	Log Message Example
The ticket provided has an expiration date.	Parsing given MapR ticket... The provided ticket expires at <timestamp>, please provide a ticket that does not expire.
The ticket provided is not a tenant ticket. For example, it may be a service ticket.	Checking if the MapR ticket is a tenant ticket...
The CLDB hosts are not reachable. For example, due to a firewall.	Using maprlogin to authenticate with username/ password... Unable to connect to any of the cluster's CLDBs. CLDBs tried: <list-of-cldbs>. Please check your cluster configuration.
MAPR_EXTERNAL is not configured properly and HPE Ezmeral Data Fabric reports internal IPs that are not accessible.	Checking if mount prefix /ezua exists... 2024-07-18 10:49:44,0671 :5926 Timing out request 2345.234 waiting to xmit binding ips are: <ip-address>:7223 2024-07-18 10:49:44,0673 ERROR Cidcache fc/ cidcache.cc:5333 Thread: 2337 Received error Connection timed out(110) while trying to reach CLDB: <cldb-host>:7223

Cause	Log Message Example
The credentials are invalid.	Using maprlogin to authenticate with username/ password... Authentication failed. Invalid username/password.
The mount prefix does not exist.	Mount prefix /ezua does not exist, attempting to create it.. . mkdir: User ezua(user id 7000) has been denied access to create ezua Mount prefix /ezua does not exist and could not create it

View the Operator Pod and Installation Container Log Files

The operator pod and installation container log files can help you troubleshoot issues with installation.

To access the log files, run the following command:

```
kubectl -n mgmt-cluster-ops-system logs op-installua-ezaf100
```

If you cannot resolve the issue using the information provided in the log files, contact HPE Support.

Metering

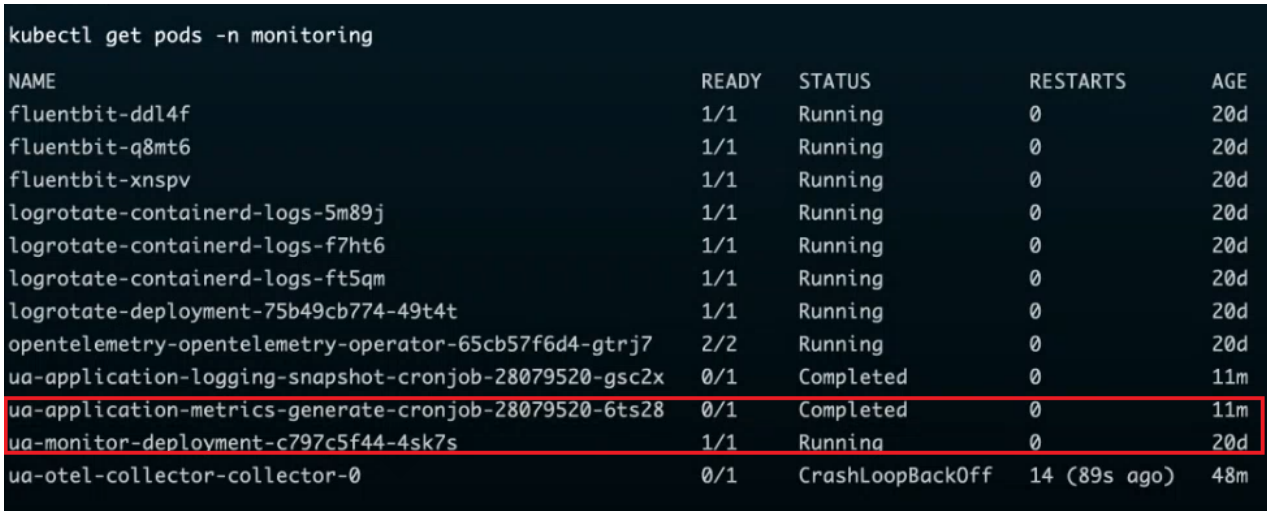
Describes how to identify and debug issues for metering.

Resource Usage or Billing Metrics

Sometimes the UI does not display or update resource usage or billing metrics. The `monitoring` and `prometheus` namespaces are used for observability in HPE Ezmeral Unified Analytics Software. Verify that the pods in these namespaces are running.

To get the list of pods in the `monitoring` namespace, run:

```
kubectl get pods -n monitoring
```



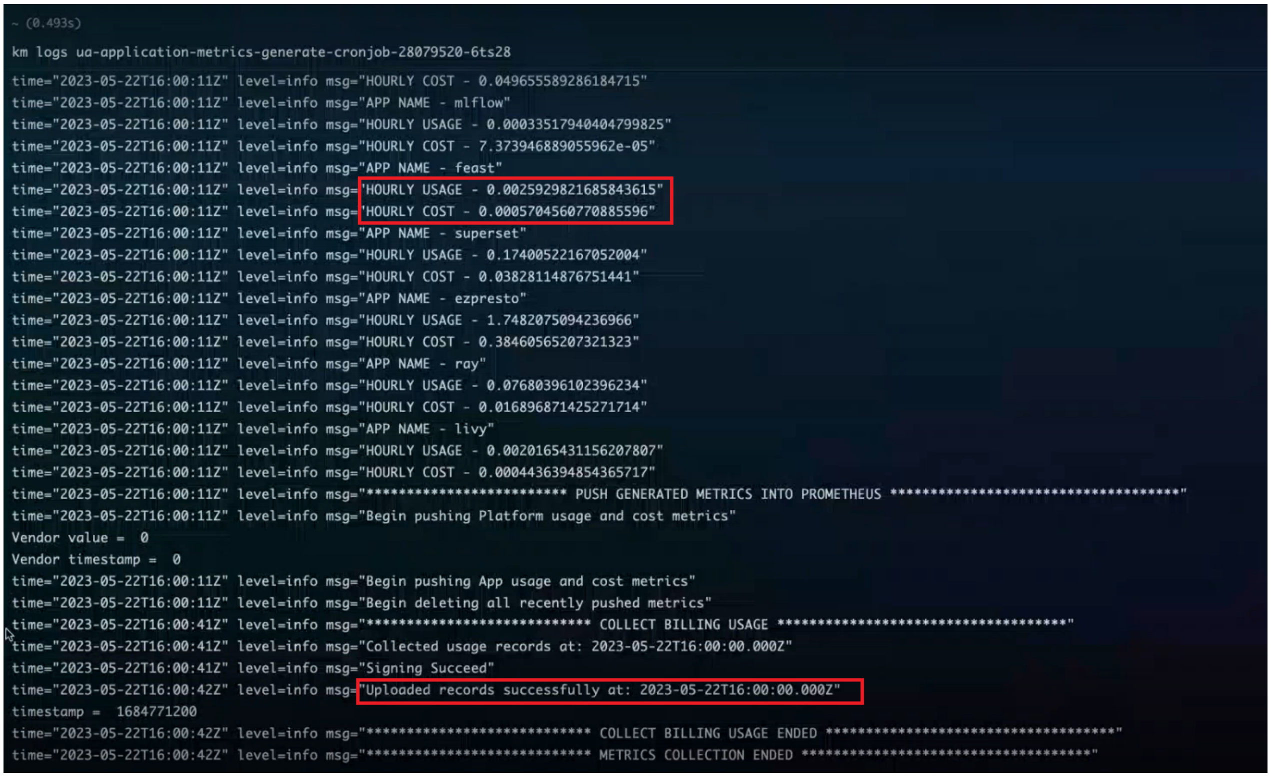
```
kubectl get pods -n monitoring
NAME                                                  READY   STATUS             RESTARTS        AGE
fluentbit-ddl4f                                       1/1     Running            0               20d
fluentbit-q8mt6                                       1/1     Running            0               20d
fluentbit-xnspv                                       1/1     Running            0               20d
logrotate-containerd-logs-5m89j                       1/1     Running            0               20d
logrotate-containerd-logs-f7ht6                       1/1     Running            0               20d
logrotate-containerd-logs-ft5qm                       1/1     Running            0               20d
logrotate-deployment-75b49cb774-49t4t                 1/1     Running            0               20d
opentelemetry-opentelemetry-operator-65cb57f6d4-gtrj7 2/2     Running            0               20d
ua-application-logging-snapshot-cronjob-28079520-gsc2x 0/1    Completed          0               11m
ua-application-metrics-generate-cronjob-28079520-6ts28 0/1    Completed          0               11m
ua-monitor-deployment-c797c5f44-4sk7s                 1/1     Running            0               20d
ua-otel-collector-collector-0                         0/1     CrashLoopBackOff   14 (89s ago)    48m
```

Verify that the `ua-application-metrics-generate-cronjob-28079520-6ts28` pod and `ua-monitor-deployment-c797c5f44` pod are running.

If **AGE** of `ua-application-metrics-generate-cronjob-28079520-6ts28` is less than 60 minutes, the cron job is up to date.

To see logs for cron jobs and to view all the aggregated values and output of values, run:

```
kubectl  logs ua-application-metrics-generate-cronjob-28079520-6ts28 -n
monitoring
```



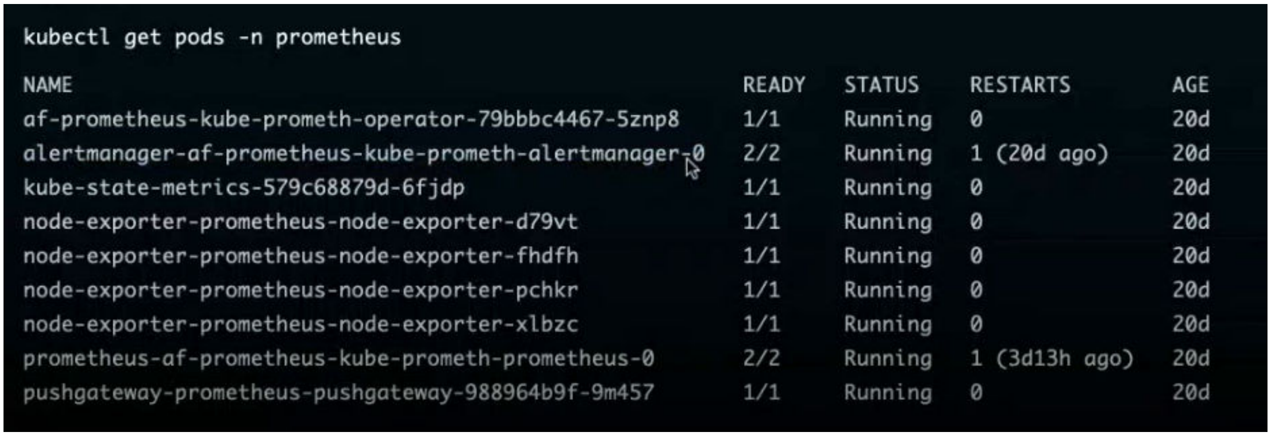
```
~ (0.493s)
km logs ua-application-metrics-generate-cronjob-28079520-6ts28
time="2023-05-22T16:00:11Z" level=info msg="HOURLY COST - 0.049655589286184715"
time="2023-05-22T16:00:11Z" level=info msg="APP NAME - mlflow"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY USAGE - 0.00033517940404799825"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY COST - 7.373946889055962e-05"
time="2023-05-22T16:00:11Z" level=info msg="APP NAME - feast"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY USAGE - 0.0025929821685843615"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY COST - 0.0005704560770885596"
time="2023-05-22T16:00:11Z" level=info msg="APP NAME - superset"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY USAGE - 0.17400522167052004"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY COST - 0.03828114876751441"
time="2023-05-22T16:00:11Z" level=info msg="APP NAME - ezpresto"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY USAGE - 1.7482075094236966"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY COST - 0.38460565207321323"
time="2023-05-22T16:00:11Z" level=info msg="APP NAME - ray"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY USAGE - 0.07680396102396234"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY COST - 0.016896871425271714"
time="2023-05-22T16:00:11Z" level=info msg="APP NAME - livy"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY USAGE - 0.0020165431156207807"
time="2023-05-22T16:00:11Z" level=info msg="HOURLY COST - 0.0004436394854365717"
time="2023-05-22T16:00:11Z" level=info msg="************************ PUSH GENERATED METRICS INTO PROMETHEUS ************************************"
time="2023-05-22T16:00:11Z" level=info msg="Begin pushing Platform usage and cost metrics"
Vendor value =  0
Vendor timestamp =  0
time="2023-05-22T16:00:11Z" level=info msg="Begin pushing App usage and cost metrics"
time="2023-05-22T16:00:11Z" level=info msg="Begin deleting all recently pushed metrics"
time="2023-05-22T16:00:41Z" level=info msg="*************************** COLLECT BILLING USAGE ************************************"
time="2023-05-22T16:00:41Z" level=info msg="Collected usage records at: 2023-05-22T16:00:00.000Z"
time="2023-05-22T16:00:41Z" level=info msg="Signing Succeed"
time="2023-05-22T16:00:42Z" level=info msg="Uploaded records successfully at: 2023-05-22T16:00:00.000Z"
timestamp =  1684771200
time="2023-05-22T16:00:42Z" level=info msg="*************************** COLLECT BILLING USAGE ENDED ************************************"
time="2023-05-22T16:00:42Z" level=info msg="*************************** METRICS COLLECTION ENDED ************************************"
```

The *Uploaded records successfully at: <time>* message suggests that the billing data was uploaded successfully.

If the **HOURLY USAGE** or **HOURLY COST** values are in zeroes, verify that Prometheus is working as expected without any errors or failures. All Prometheus pods are located in the `prometheus` namespace.

To get the list of pods in the `prometheus` namespace, run:

```
kubectl get pods -n prometheus
```



```
kubectl get pods -n prometheus
NAME                                                  READY   STATUS    RESTARTS        AGE
af-prometheus-kube-prometh-operator-79bbbc4467-5znp8  1/1     Running   0               20d
alertmanager-af-prometheus-kube-prometh-alertmanager-0 2/2    Running   1 (20d ago)     20d
kube-state-metrics-579c68879d-6fjdp                   1/1     Running   0               20d
node-exporter-prometheus-node-exporter-d79vt          1/1     Running   0               20d
node-exporter-prometheus-node-exporter-fhdfh          1/1     Running   0               20d
node-exporter-prometheus-node-exporter-pchkr          1/1     Running   0               20d
node-exporter-prometheus-node-exporter-xlbzc          1/1     Running   0               20d
prometheus-af-prometheus-kube-prometh-prometheus-0    2/2     Running   1 (3d13h ago)   20d
pushgateway-prometheus-pushgateway-988964b9f-9m457    1/1     Running   0               20d
```

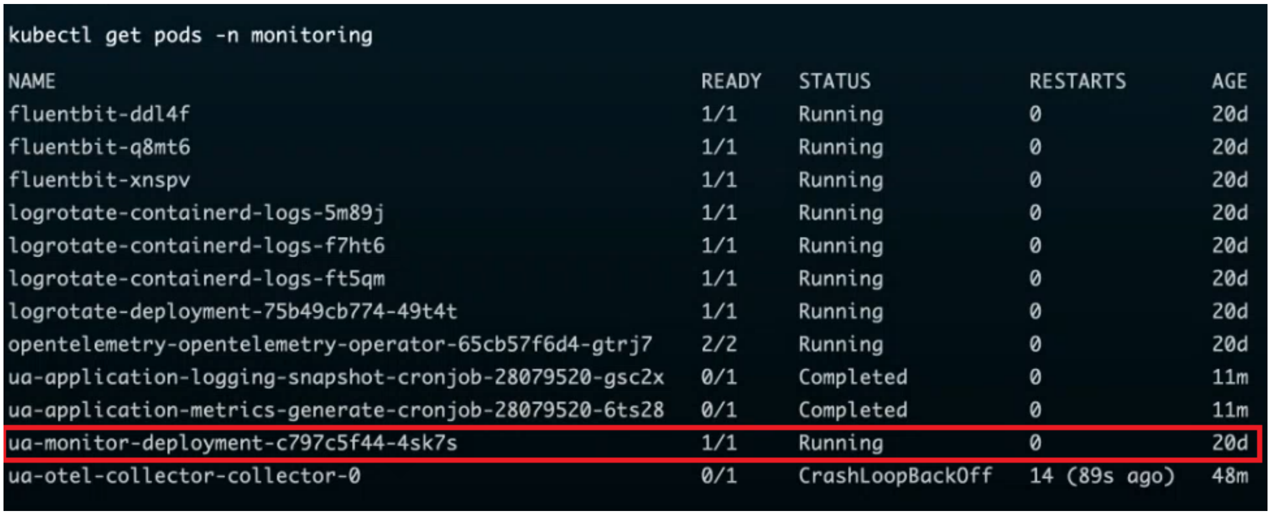
Monitoring

Describes how to identify and debug issues for monitoring.

Failure to display alerts and notifications

Verify that the `ua-monitor-deployment-c797c5f44` pod is up and running. To get the list of pods in the `monitoring` namespace, run:

```
kubectl get pods -n monitoring
```



```
kubectl get pods -n monitoring
NAME                                                 READY   STATUS             RESTARTS        AGE
fluentbit-ddl4f                                      1/1     Running            0               20d
fluentbit-q8mt6                                      1/1     Running            0               20d
fluentbit-xnspv                                      1/1     Running            0               20d
logrotate-containerd-logs-5m89j                      1/1     Running            0               20d
logrotate-containerd-logs-f7ht6                      1/1     Running            0               20d
logrotate-containerd-logs-ft5qm                      1/1     Running            0               20d
logrotate-deployment-75b49cb774-49t4t                1/1     Running            0               20d
opentelemetry-opentelemetry-operator-65cb57f6d4-gtrj7   2/2  Running            0               20d
ua-application-logging-snapshot-cronjob-28079520-gsc2x  0/1  Completed          0               11m
ua-application-metrics-generate-cronjob-28079520-6ts28  0/1  Completed          0               11m
ua-monitor-deployment-c797c5f44-4sk7s                1/1     Running            0               20d
ua-otel-collector-collector-0                        0/1     CrashLoopBackOff   14 (89s ago)    48m
```

Verify that the `alertmanager-kubeprom-alertmanager-0` pod is up and running. To get the list of pods in the `prometheus` namespace, run:

```
kubectl get pods -n prometheus
```

Notebooks

Describes how to identify and debug issues for Notebooks.

The Default User Jupyter Notebook Cannot Connect to Kubeflow

When you try to connect your default user notebook, the Kubeflow UI returns the following message:

`Couldn't find any information for the status of this notebook`

This occurs when a username starts with a number, such as `3user`, because notebooks cannot have names that start with a number.

When a user is added to HPE Ezmeral Unified Analytics Software, the system automatically creates a default notebook for the user and assigns the notebook a name in the following format:

`<username>-notebook`

If the username starts with a number, such as `3user`, the default user notebook name also starts with a number (`3user-notebook`), which is not supported. When this occurs, Kubeflow does not recognize the notebook, due to the name, and cannot connect.

Workaround

Use either of the following options to resolve the issue:

Option 1	Create a new notebook with the same image and configurations. Make sure that the notebook name consists of lowercase alphanumeric characters, with or without dashes (-) and does not start with a number. The name must start with a letter (a-z). For example, you can name a notebook `my-notebook-1`, but you cannot name a notebook `1-my-notebook`.

Option 2

Ask your HPE Ezmeral Unified Analytics Software admin to delete the user account and then create a new one with a username that adheres to the `Username Attribute` naming requirements, as described in AD/LDAP Servers on page 227.

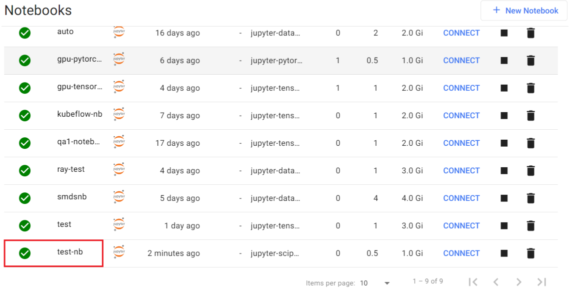
“No healthy upstream” Error in Notebook Server Connection

When connecting to the notebook server, you may get the `"no healthy upstream"` error message due to an unhealthy notebook pod. To identify the issue, you must check pod logs and events either using the Kubeflow UI or manually using the `kubectl` commands.

Using Kubeflow UI

To access pod logs, events, and check the container status from the Kubeflow UI, follow these steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. Click the **Tools & Frameworks** icon on the left navigation bar.
3. Navigate to the **Kubeflow** tile under the **Data Science** tab and click **Open**.
4. In the **Kubeflow Central Dashboard** UI, click **Notebooks** on the left navigation bar.
5. Click **<your-unhealthy-notebook-name>** to view the notebook details.



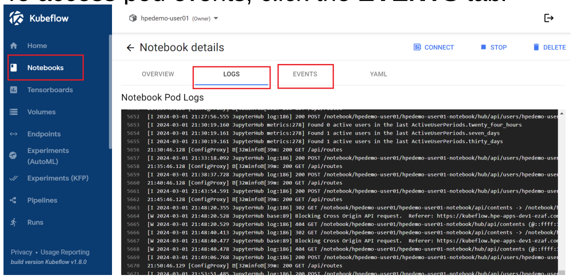
Notebooks

+ New Notebook

✓	auto	jupyter	16 days ago	-	jupyter-data...	0	2	2.0 Gi	CONNECT	■	🗑
✓	gpu-pytorc...	jupyter	6 days ago	-	jupyter-pytor...	1	0.5	1.0 Gi	CONNECT	■	🗑
✓	gpu-tensor...	jupyter	4 days ago	-	jupyter-tens...	1	1	2.0 Gi	CONNECT	■	🗑
✓	kubeflow-nb	jupyter	7 days ago	-	jupyter-tens...	0	1	2.0 Gi	CONNECT	■	🗑
✓	qa1-noteb...	jupyter	17 days ago	-	jupyter-tens...	0	1	2.0 Gi	CONNECT	■	🗑
✓	ray-test	jupyter	4 days ago	-	jupyter-data...	0	1	3.0 Gi	CONNECT	■	🗑
✓	smdsnb	jupyter	5 days ago	-	jupyter-data...	0	4	4.0 Gi	CONNECT	■	🗑
✓	test	jupyter	1 day ago	-	jupyter-tens...	0	1	3.0 Gi	CONNECT	■	🗑
✓	test-nb	jupyter	2 minutes ago	-	jupyter-scip...	0	0.5	1.0 Gi	CONNECT	■	🗑

Items per page: 10 ▾ 1 – 9 of 9

6. To check the current status of the container, click the **OVERVIEW** tab and look for the **Conditions** section. The **Conditions** section shows the current status of the container.
7. To access pod logs, click the **LOGS** tab.
8. To access pod events, click the **EVENTS** tab.



Notebook Pod Logs

```
[I 2024-03-01 21:27:56.555 JupyterHub log:186] 200 POST /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/users/hpedemo-user
[I 2024-03-01 21:30:19.160 JupyterHub metrics:278] Found 0 active users in the last ActiveUserPeriods.twenty_four_hours
[I 2024-03-01 21:30:19.161 JupyterHub metrics:278] Found 1 active users in the last ActiveUserPeriods.seven_days
[I 2024-03-01 21:30:19.161 JupyterHub metrics:278] Found 1 active users in the last ActiveUserPeriods.thirty_days
21:30:46.128 [ConfigProxy] ⊠[32minfo⊠[39m: 200 GET /api/routes
[I 2024-03-01 21:33:18.092 JupyterHub log:186] 200 POST /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/users/hpedemo-user
21:35:46.128 [ConfigProxy] ⊠[32minfo⊠[39m: 200 GET /api/routes
[I 2024-03-01 21:38:37.728 JupyterHub log:186] 200 POST /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/users/hpedemo-user
21:40:46.128 [ConfigProxy] ⊠[32minfo⊠[39m: 200 GET /api/routes
[I 2024-03-01 21:43:54.591 JupyterHub log:186] 200 POST /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/users/hpedemo-user
21:45:46.128 [ConfigProxy] ⊠[32minfo⊠[39m: 200 GET /api/routes
[I 2024-03-01 21:48:20.355 JupyterHub log:186] 302 GET /notebook/hpedemo-user01/hpedemo-user01-notebook/api/contents -> /notebook/h
[W 2024-03-01 21:48:20.528 JupyterHub base:89] Blocking Cross Origin API request.  Referer: https://kubeflow.hpe-apps-dev1-ezaf.com
[W 2024-03-01 21:48:20.529 JupyterHub log:186] 404 GET /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/contents (@::ffff:1
[I 2024-03-01 21:48:40.413 JupyterHub log:186] 302 GET /notebook/hpedemo-user01/hpedemo-user01-notebook/api/contents -> /notebook/h
[W 2024-03-01 21:48:40.477 JupyterHub base:89] Blocking Cross Origin API request.  Referer: https://kubeflow.hpe-apps-dev1-ezaf.com
[W 2024-03-01 21:48:40.478 JupyterHub log:186] 404 GET /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/contents (@::ffff:1
[I 2024-03-01 21:49:06.768 JupyterHub log:186] 200 POST /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/users/hpedemo-user
21:50:46.129 [ConfigProxy] ⊠[32minfo⊠[39m: 200 GET /api/routes
[I 2024-03-01 21:53:51.485 JupyterHub log:186] 200 POST /notebook/hpedemo-user01/hpedemo-user01-notebook/hub/api/users/hpedemo-user
```

Using `kubectl` Commands

To access pod logs, events, and check the container status from the commandline, follow these steps:

- To get pod events and container statuses, run:

  ```
  kubectl describe pod -n <user-ns>
  <notebook-name>-0
  ```

  Output:

  ```
  Name:             temp-0
  Namespace:        hpedemo-user01

  .........

    temp:
      Container ID:
      Image:          gcr.io/
  mapr-252711/kubeflow/notebooks/
  jupyter-tensorflow-full:ezaf-v1.8.0
      Image ID:
      Port:           8888/TCP
      Host Port:      0/TCP
      State:          Waiting
        Reason:       PodInitializing
      Ready:          False
      Restart Count:  0

  .......

  Events:
    Type     Reason
  Age   From
  Message
    ----     ------
   ----  ----
   -------
    Warning  FailedScheduling
  48s   default-scheduler        0/6
  nodes are available: pod has
  unbound immediate
  PersistentVolumeClaims.
  preemption: 0/6 nodes are
  available: 6 Preemption is not
  helpful for scheduling..
    Warning  FailedScheduling
  46s   default-scheduler        0/6
  nodes are available: pod has
  unbound immediate
  PersistentVolumeClaims.
  preemption: 0/6 nodes are
  available: 6 Preemption is not
  helpful for scheduling..
    Normal   Scheduled
  44s   default-scheduler
  Successfully assigned
  hpedemo-user01/temp-0 to
  mip-bd-dev04.mip.storage.hpecorp.ne
  t
    Normal   SuccessfulAttachVolume
  44s   attachdetach-controller
  AttachVolume.Attach succeeded for
  ```

```
volume
"mapr-pv-bd0db07c-4e43-4e78-8503-7f
61649a7bd0"
  Normal   Pulling
35s   kubelet
Pulling image
"marketplace.us1.greenlake-hpe.com/
ezua/istio/proxyv2:1.16.2"
  Normal   Pulled
34s   kubelet
Successfully pulled image
"marketplace.us1.greenlake-hpe.com/
ezua/istio/proxyv2:1.16.2" in
1.127945155s (1.127954107s
including waiting)
  Normal   Created
34s   kubelet
Created container istio-validation
  Normal   Started
34s   kubelet
Started container istio-validation
  Normal   Pulling
33s   kubelet
Pulling image
"marketplace.us1.greenlake-hpe.com/
ezua/istio/proxyv2:1.16.2"
  Normal   Pulled
29s   kubelet
Successfully pulled image
"marketplace.us1.greenlake-hpe.com/
ezua/istio/proxyv2:1.16.2" in
4.611252056s (4.611259156s
including waiting)
  Normal   Created
29s   kubelet
Created container istio-proxy
  Normal   Started
28s   kubelet
Started container istio-proxy
  Normal   Pulling
27s   kubelet
Pulling image "gcr.io/mapr-252711/
kubeflow/notebooks/
jupyter-tensorflow-full:ezaf-v1.8.0
"
```

- To get pod logs, run:

  ```
  kubectl logs -n <user-ns>
  <notebook-name>-0
  ```

Result:

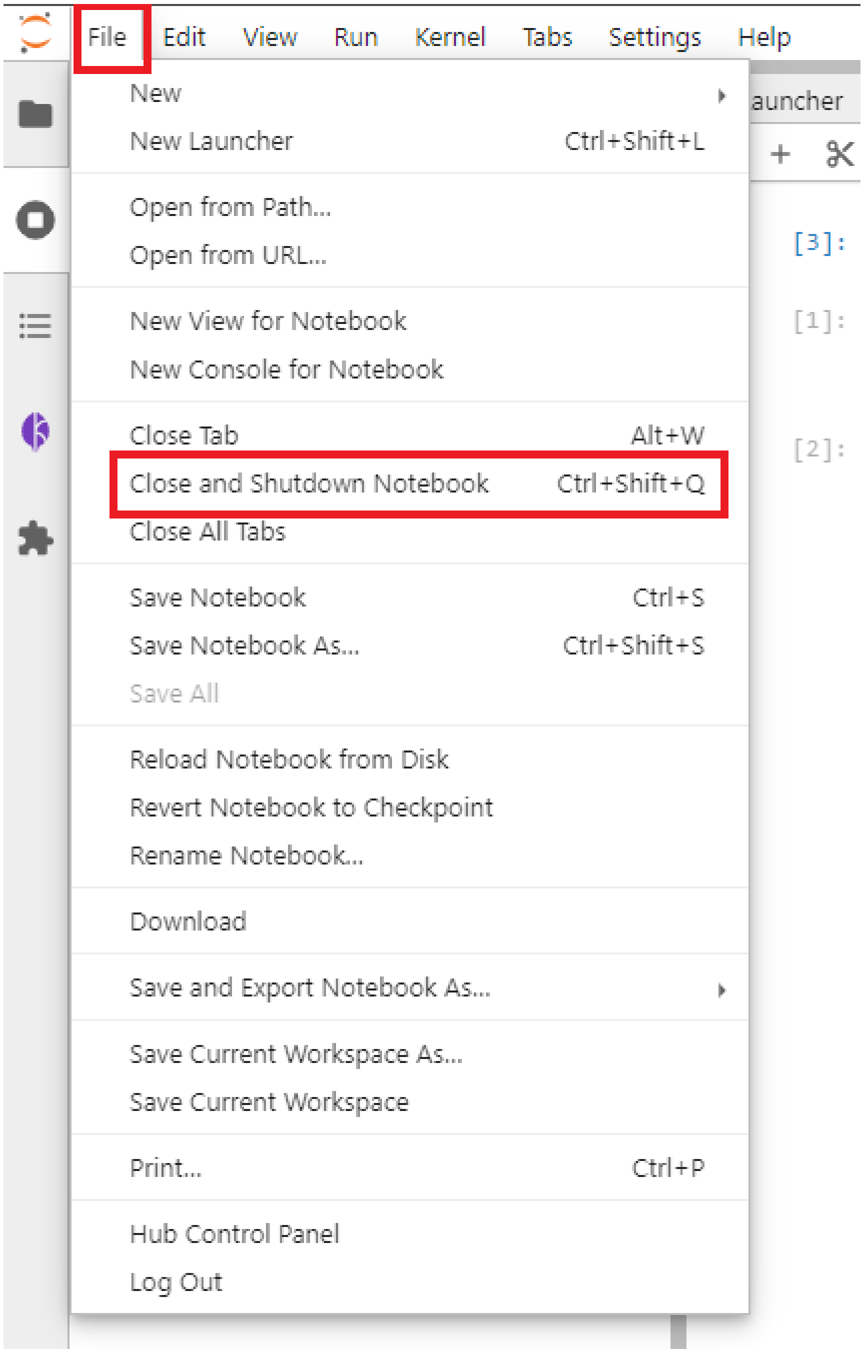
You can now identify the issue by checking pod logs, events, and the current status of the container.

Memory Accumulation and Unreleased Memory in Jupyter Notebooks

Memory consumption keeps increasing as Jupyter Notebooks are run. Even after closing the notebook, memory is not released which leads to a gradual accumulation of objects in memory with each notebook run. Eventually, the notebook server becomes unusable as memory reaches its limits and you are required to launch a new notebook server.

To release the memory, follow these steps to kill the kernels of closed notebooks:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. Click **Notebooks icon** on the left navigation bar of HPE Ezmeral Unified Analytics Software screen.
3. Connect to the notebook server.
4. Open the notebook you want to close.
5. Click **File** in the menu bar.
6. Select **Close and Shutdown Notebook**.



7. Repeat the process for any other notebooks that are no longer in use.

Result:

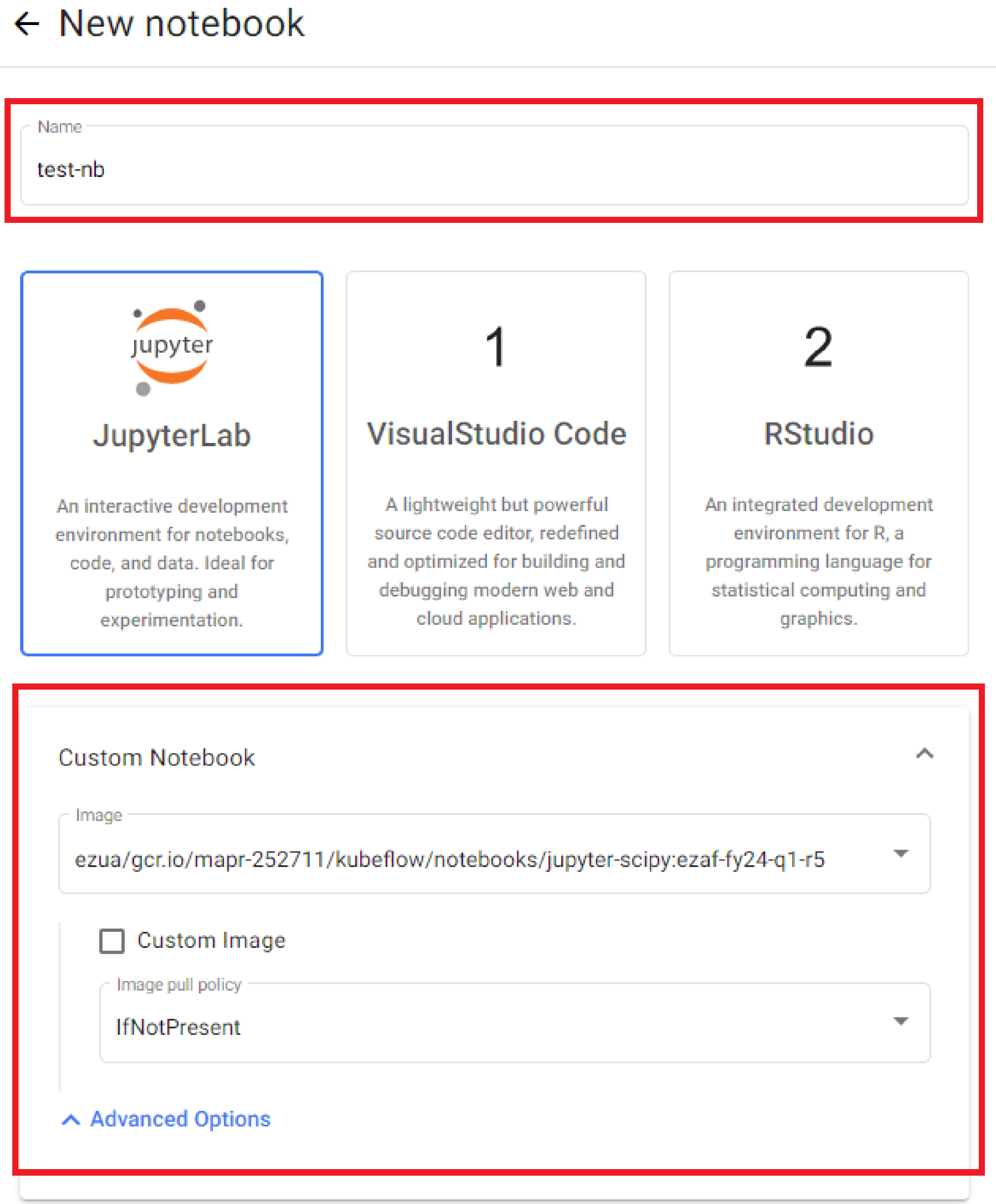
By closing the notebooks using the **Close and Shutdown Notebook** option, you ensure that associated kernel is properly shut down which releases the memory it was using. This prevents the accumulation of objects in memory and keeps the notebook server usable for longer periods.

Specified Image Pull Policy Not Applied to a Pod

When you create a notebook server and set the `imagePullPolicy` to `IfNotPresent` or `Never`, the specified image pull policy is not set to the pod. In both scenarios, the `imagePullPolicy` is set to `Always`.

To verify that the specified image pull policy is not applied to a pod, follow these steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. Click **Notebooks icon** on the left navigation bar of HPE Ezmeral Unified Analytics Software screen.
3. Click **New Notebook Server**. You will be navigated to the Kubeflow Notebooks UI.
4. Enter the name of the notebook server.



← New notebook

Name: test-nb

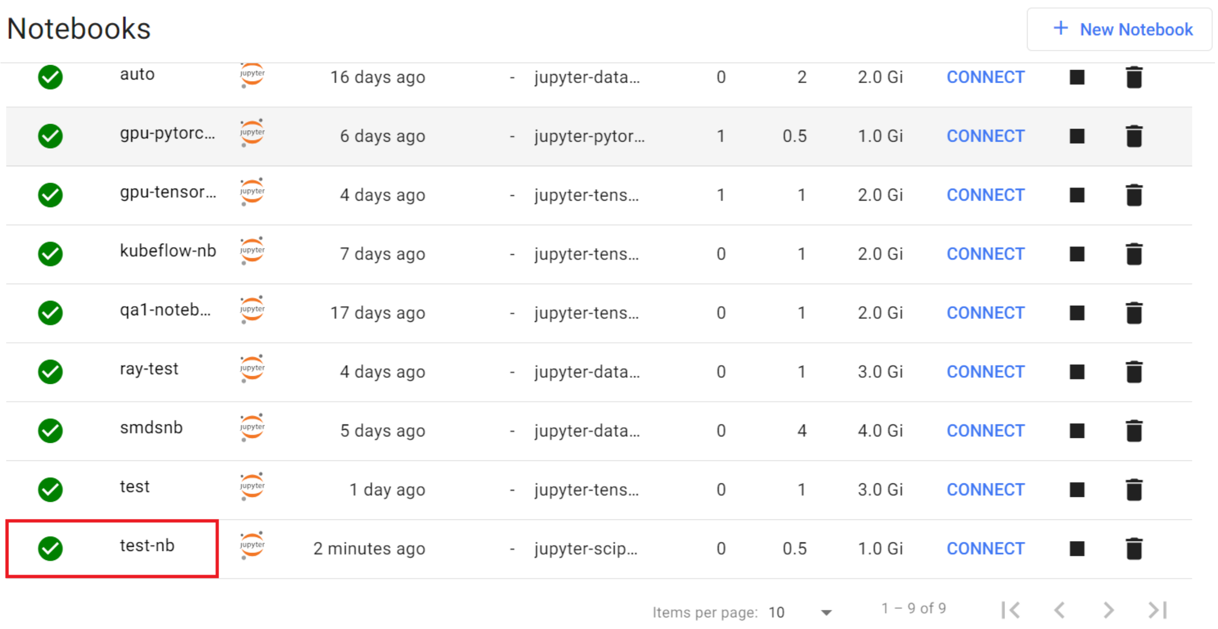
jupyter
JupyterLab
An interactive development environment for notebooks, code, and data. Ideal for prototyping and experimentation.
1
VisualStudio Code
A lightweight but powerful source code editor, redefined and optimized for building and debugging modern web and cloud applications.
2
RStudio
An integrated development environment for R, a programming language for statistical computing and graphics.
Custom Notebook

Image: ezua/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezaf-fy24-q1-r5

□ Custom Image

Image pull policy: IfNotPresent

5. Click **Custom Notebook**.
6. Click **Advanced Options**.
7. Set **Image pull policy** to **IfNotPresent**.
8. To launch the notebook server, click **Launch.**
9. After creating the notebook server, click **<your-notebook-name>** to view the notebook details.



Notebooks

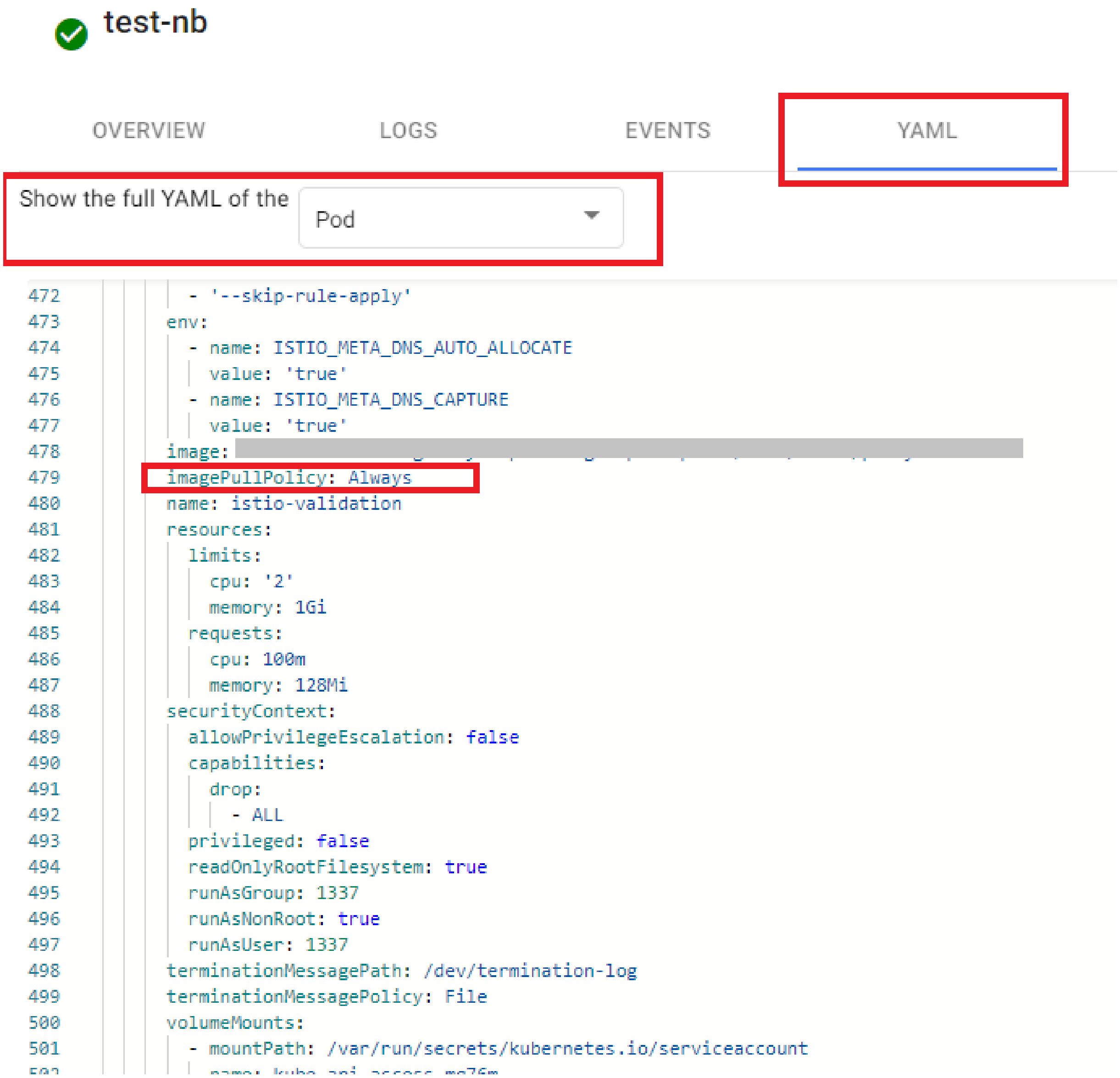
✓	auto	jupyter	16 days ago	-	jupyter-data...	0	2	2.0 Gi	CONNECT	■	🗑
✓	gpu-pytorc...	jupyter	6 days ago	-	jupyter-pytor...	1	0.5	1.0 Gi	CONNECT	■	🗑
✓	gpu-tensor...	jupyter	4 days ago	-	jupyter-tens...	1	1	2.0 Gi	CONNECT	■	🗑
✓	kubeflow-nb	jupyter	7 days ago	-	jupyter-tens...	0	1	2.0 Gi	CONNECT	■	🗑
✓	qa1-noteb...	jupyter	17 days ago	-	jupyter-tens...	0	1	2.0 Gi	CONNECT	■	🗑
✓	ray-test	jupyter	4 days ago	-	jupyter-data...	0	1	3.0 Gi	CONNECT	■	🗑
✓	smdsnb	jupyter	5 days ago	-	jupyter-data...	0	4	4.0 Gi	CONNECT	■	🗑
✓	test	jupyter	1 day ago	-	jupyter-tens...	0	1	3.0 Gi	CONNECT	■	🗑
✓	test-nb	jupyter	2 minutes ago	-	jupyter-scip...	0	0.5	1.0 Gi	CONNECT	■	🗑

Items per page: 10 ▾ 1 – 9 of 9

10. Click the **YAML** tab.

11. Select **Show the full YAML of the Pod**.

12. Locate the `imagePullPolicy` property for the image used in creating the notebook.



```
  - '--skip-rule-apply'
env:
  - name: ISTIO_META_DNS_AUTO_ALLOCATE
    value: 'true'
  - name: ISTIO_META_DNS_CAPTURE
    value: 'true'
image: [redacted]
imagePullPolicy: Always
name: istio-validation
resources:
  limits:
    cpu: '2'
    memory: 1Gi
  requests:
    cpu: 100m
    memory: 128Mi
securityContext:
  allowPrivilegeEscalation: false
  capabilities:
    drop:
      - ALL
  privileged: false
  readOnlyRootFilesystem: true
  runAsGroup: 1337
  runAsNonRoot: true
  runAsUser: 1337
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
  - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
```

Result:

The `imagePullPolicy` is set to `Always`.

Security

Describes how to identify and debug issues related to security.

Internal Network Connection Issue

SPIRE is currently implemented to rotate the certificates that secure mTLS traffic internally. An internal network connection issue can arise if SPIRE elements go down.

To troubleshoot internal network connection issues, run the following commands to get the logs from the spire-controller-manager container in the SPIRE server pods:

```
kubectl -n spire logs spire-server-0 -c spire-controller-manager

kubectl -n spire logs spire-server-1 -c spire-controller-manager
```

Oauth2-Proxy Pod in CrashLoopBackOff Error State

The container in the `oauth2-proxy` pod started, but crashes and repeatedly restarts incorrectly. To resolve this issue, update the DNS A record with the node IP address where the Istio Ingress gateway pod was deployed. You can get the node IPs on the Status page of the HPE Ezmeral Unified Analytics Software installer.

Authentication

If authentication is not working, check to see if pods are running and check pod logs. You may just need to kill a pod and restart it on a healthy node. For more complicated issues, you may need to collect some logs and talk to HPE support.

The following table provides pod information and commands you may want to run if authentication is not working:

Component	Information

Internal OpenLDAP server	The server runs the `ldap-0` pod in the `hpe-ldap` namespace. You can access the `ldap-0` pod with the following internal service DNS name:
	`ldap-svc.hpe-ldap.svc.cluster.local:389`
	If you need to bind it to read some users, use the following DN and password:
	`cn=readonly,dc=example,dc=com`
	`password: mapr`
	Search from the following base:
	`ldapsearch -Y EXTERNAL -Q -H ldapi:/// -b ou=users,dc=example,dc=com`
	You can also `exec` into the `ldap-0` pod and use `ldap *` local utilities like `ldap search` to investigate what the internal LDAP server looks like.

Oauth-2 proxy	HPE Ezmeral Unified Analytics Software uses Oauth2 proxy for authentication. Oauth2 runs in the `oauth2-proxy` namespace. To get the pod logs, run the following command: `kubectl -n oauth2-proxy logs -l app=oauth2-proxy`
Keycloak	HPE Ezmeral Unified Analytics Software uses a local instance of Keycloak as its OIDC provider. Keycloak runs in the `keycloak` namespace in the `keycloak-0` pod. There is also a PostgreSQL pod running in the namespace that stores the configuration and current known user information. Keycloak pod logs show login attempts and any AD/LDAP integration errors. To view the Keycloak logs, run: `kubectl -n keycloak logs keycloak-0`

Keycloak/LDAP User Authentication Test

Use a direct grant endpoint to see if a user can authenticate to Keycloak. Run this test on the command line to verify that the AD/LDAP integration is working for a particular user and that the password is correct. This test hits a client in Keycloak that allows the direct grant on the authentication flow. This test is not doing a web-based redirect flow; it is just saying give me your credentials for a token in return.

Run this test on the command line and then go to https://jwt.io/:

```
USER=<username>
PASS=<password>
DOMAIN=<your-domain>.com
RESULT=$(curl -k --data
"username=$USER&password=$PASS&grant_type=password&client_id=ua-grant"
https://keycloak.$DOMAIN/realms/UA/protocol/openid-connect/token)
ACCESS_TOKEN=$(echo $RESULT | sed 's/.*access_token":"//g' | sed 's/".*//g')
```

You can also use the direct grant endpoint for REST API endpoints that must accept and validate username and password credentials. The direct grant endpoint can validate the incoming username and password and get an access token. The remainder of the flow inside HPE Ezmeral Unified Analytics Software is based on the access token.

Keycloak Admin Web Console

The Keycloak Admin Web Console is useful for finding and resolving issues. To access the Keycloak Admin Web Console, you will need the system master's `kubectl` privileges that you got when you first installed and created the HPE Ezmeral Unified Analytics Software cluster. This is required to get the password for the keycloak administrator.

To access the Keycloak Admin Web Console:

1. Go to `keycloak.<your-UA-domain>`.
2. Enter `admin` as the user.
3. Enter the password. You can get the password with `kubectl` and `kube.config`:

   ```
    kubectl -n keycloak get secret admin-pass -o
   jsonpath="{.data.password}" | base64 -D
   ```

   NOTE: For Linux, use `base64 -d`. For Mac, use `base64 -D`.

4. In the left navigation bar of the Keycloak UI, switch over to the **UA realm**.

To find all users:

1. In the left navigation bar, select **Users**.
2. On the **User list** tab, enter * in the search field to see all known users.

To modify roles for a particular user:

NOTE: This process can be useful if someone deletes the HPE Ezmeral Unified Analytics Software admin users in the AD server. The user whose role you modify may need to sign out of HPE Ezmeral Unified Analytics Software, and sign back in for the change to take effect.

1. On the **Users** page, search for the user and click on the **username**.
2. Select the username in the **User list**.

3. On the user's page, select **Role Mapping**.

4. Select the role that you want to assign to the user.

To change how users sign in (email vs username):

1. In the left navigation bar, go to **Realm Settings > Login tab**.

2. On the **Login tab**, select the **Login with email** switch.

   Duplicate emails turns off automatically when you do this. This is safe to do for external AD/LDAP servers. For internal AD/LDAP servers, you can also turn this on, but there are no guard rails on the internal user management to prevent duplicate emails. If this happens, one of the users will not be able to log in.

To manually grant or remove the `ua-enabled` role:

NOTE: The `ua-enabled` role is the internal role that grants access to HPE Ezmeral Unified Analytics Software. If the user object does not have this role in Keycloak, they cannot authenticate and get access to HPE Ezmeral Unified Analytics Software applications. However, this does not do all the user onboarding and offboarding that the user management operator does; it merely prevents access to HPE Ezmeral Unified Analytics Software. Setting this role in Keycloak should only be used in special or emergency circumstances.

1. In the left navigation bar, select **Users**.

2. Search for the user and select the user.

3. On the user's page, select the **Role Mapping** tab.

4. Select **ua-enabled**.

To see the LDAP settings that identify the source of the HPE Ezmeral Unified Analytics Software user directory:

1. In the left navigation bar, select **User federation**.

2. On the **LDAP** page, select the **Settings tab**.

You can change the following settings:

- How Keycloak batch requests to the remote server.

- How often Keycloak synchs users from the server; the default is hourly.

  NOTE: You should not modify the settings you entered through the HPE Ezmeral Unified Analytics Software installer, such as the server address. Changing those settings here can have negative consequences because this is not the only place that those settings are stored. Keycloak uses the LDAP information, but so do some of the other HPE Ezmeral Unified Analytics Software applications. Currently, it is best not to change LDAP settings because it can break applications that depend on them.

Bad HTTP Request

When a custom framework expects HTTPS traffic, the following error displays when you try to access the service in the browser:

```
400 Bad Request - The plain HTTP request was sent to HTTPS port.
```

To resolve this issue, add a DestinationRule in addition to the VirtualService, as shown in the following examples.

DestinationRule Example

```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: {{ include "test-app.fullname" . }}
  namespace: {{ .Release.Namespace }}
  labels:
    {{- include "test-app.labels" . | nindent 4 }}

 #The URL should point to the corresponding service.
 #Kubernetes provides an internal DNS mapping for services using the format
<ServiceName>.<ServiceNamespace>.svc.cluster.local.
spec:
  host: {{ include "test-app.fullname" . }}.
{{ .Release.Namespace }}.svc.cluster.local
  trafficPolicy:
    tls:
      mode: SIMPLE
```

VirtualService Example

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: {{ include "test-app.fullname" . }}
  namespace: {{ .Release.Namespace }}
  labels:
    {{- include "test-app.labels" . | nindent 4 }}
spec:
  gateways:
    - {{ .Values.ezua.VirtualService.istioGateway }}
  hosts:
    - {{ .Values.ezua.VirtualService.endpoint }}
  #The following VirtualService options are specific and depend on the
application implementation.
  #This example is a simple application with single service and simple
match routes.
  #The URL should point to the corresponding service.
  #Kubernetes provides an internal DNS mapping for services using the
format <ServiceName>.<ServiceNamespace>.svc.cluster.local.
  http:
    - match:
        - uri:
            prefix: /
      rewrite:
        uri: /
      route:
        - destination:
            host: {{ include "test-app.fullname" . }}.
{{ .Release.Namespace }}.svc.cluster.local
            port:
              number: {{ .Values.service.port }}
```

Spark

Describes how to identify and debug issues for Spark.

Spark History Server

Long-running Spark applications exceed disk quotas for Spark History Server.

Repeatedly running long-running Spark applications generates a large volume of logs in the Spark History Server event log directory. This can exceed disk quotas, causing failures in other Spark applications. You must monitor log sizes and manage disk space to mitigate this issue.

Workaround

To prevent the exceeding of disk quota for the Spark History Server event log directory, modify the Spark History Server configuration options as follows:

To periodically clean up event logs from storage, set:

```
spark.history.fs.cleaner.enabled true
```

To delete job log files older than the specified value, set:

```
spark.history.fs.cleaner.maxAge 1d
```

Here, job log files that are older than 1d are deleted by the filesystem history cleaner.

To specify the frequency for the filesystem job history cleaner to check for the files to be deleted, set:

```
spark.history.fs.cleaner.interval 12h
```

Here, the filesystem job history cleaner checks every 12h for files to be deleted.

To enable event log rolling based on size, set:

```
spark.eventLog.rolling.enabled true
```

The default is deactivated.

To specify the maximum size of the event log file before it rolls over, set:

```
spark.eventLog.rolling.maxFileSize 128m
```

The default is 128 MB.

To specify the maximum number of non-compacted event log files to retain, set:

```
spark.history.fs.eventLog.rolling.maxFilesToRetain 2
```

By default, all event log files are retained. To compact older event logs, reduce the value. The minimum value accepted is 1.

NOTE: Compaction tries to exclude events that point to outdated event log files, such as the following events:

- Events for finished jobs and related staged/task events
- Events for the terminated executor
- Events for finished SQL execution and related job/staged/task events

Discarded events do not display in the Spark History Server UI.

NOTE: If the disk quota is full, contact HPE Support for assistance.

Spark Operator

Spark application submission hangs or fails.

If the Spark application submission hangs or fails, check the submission pod state.

- If the pod is in the pending state, wait for more resources to be available.
- If the pod is in the failed state, collect pod logs and contact HPE Support.

Spark application hangs in the Submitted or Running state.

If the Spark application hangs in the **Submitted** or **Running** state, check the state of the driver pod.

- If the driver pod is in the `ContainerCreating` state, check the pod events.
    - If the image is downloading, wait until the image is downloaded.
    - For the `FailedMount` reason, you need to identify what volume is missing.
        - By default, all Spark workloads submitter pods are preconfigured to mount system volumes such as Spark History Server PVC, user PVC, and shared PVC.
        - If the problem is with the system volume, contact HPE Support.
    - If the driver pod is in a `Running` state, check if executor pods are in a Running state as well . Sometimes executor pods are in a pending state due to a lack of resources, in this case, wait for the resources to be available.
- For other reasons, collect driver and executor pod logs and contact HPE Support.

Spark application fails.

If the Spark application fails, collect the driver pod logs.

- If the container fails before running the application code, contact HPE Support as there is a problem with the image.

	• If the container fails while the application is running, check the exception in a driver log: • For the functional exception (e.g. NullPointerException), review the application source code. • For the non-functional exception (e.g. OutOfMemoryError), increase memory allocation for the driver pod and/or review the application source code.

Livy

Livy session creation fails.	If the Livy session fails, create a Livy session with the default configuration and run. • If it runs successfully, check the configuration of your failed Livy session for configuration issues. • If it fails, collect the Livy server pod logs and driver pod logs (if available) and contact HPE Support.
Livy session hangs in the Starting state.	Verify that the driver pod is not in the **Pending or ContainerCreating** state.
Livy statement run hangs or fails.	If the Livy statement run hangs or fails, • Analyze the error message and fix the statement. For a detailed error message information, go to the Livy Server UI. • Verify the executor pods are not in the **Pending** state. For the Livy statements to run, executor pods must be available. • For other reasons, collect driver and executor pod logs and contact HPE Support.
Livy session disappears.	No action required as this is an expected behaviour for the idling sessions.
Executor pod logs are not available for the interactive sessions.	When you create a Spark interactive session by setting `spark.log.level` as key and `INFO` as value and then submit the Livy statements, the executor pod logs are not available after the session completion. **Workaround:** To resolve this issue, follow these steps: **1.** Create a custom log4j properties file named `custom_log4j.properties` in the shared volume (`local:///mounts/shared-volume/custom_log4j.properties`).

2. Copy the following content of the `log4j2.properties` file of the driver pod of the Livy session to the `custom_log4j.properties` file.

```
#
# Licensed to the Apache Software
Foundation (ASF) under one or more
# contributor license agreements.
See the NOTICE file distributed
with
# this work for additional
information regarding copyright
ownership.
# The ASF licenses this file to
You under the Apache License,
Version 2.0
# (the "License"); you may not use
this file except in compliance with
# the License.  You may obtain a
copy of the License at
#
#    http://www.apache.org/
licenses/LICENSE-2.0
#
# Unless required by applicable
law or agreed to in writing,
software
# distributed under the License is
distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS
OF ANY KIND, either express or
implied.
# See the License for the specific
language governing permissions and
# limitations under the License.
#

# Set everything to be logged to
the console
rootLogger.level = WARN
rootLogger.appenderRef.stdout.ref
= console

# In the pattern layout
configuration below, we specify an
explicit `%ex` conversion
# pattern for logging Throwables.
If this was omitted, then (by
default) Log4J would
# implicitly add an `%xEx`
conversion pattern which logs
stacktraces with additional
# class packaging information.
That extra information can
sometimes add a substantial
# performance overhead, so we
disable it in our default logging
config.
# For more information, see
SPARK-39361.
```

```
appender.console.type = Console
appender.console.name = console
appender.console.target =
SYSTEM_ERR
appender.console.layout.type =
PatternLayout
appender.console.layout.pattern =
%d{yy/MM/dd HH:mm:ss} %p %c{1}:
%m%n%ex

# Set the default spark-shell/
spark-sql log level to WARN. When
running the
# spark-shell/spark-sql, the log
level for these classes is used to
overwrite
# the root logger's log level, so
that the user can have different
defaults
# for the shell and regular Spark
apps.
logger.repl.name =
org.apache.spark.repl.Main
logger.repl.level = WARN

logger.thriftserver.name =
org.apache.spark.sql.hive.thriftser
ver.SparkSQLCLIDriver
logger.thriftserver.level = WARN

# Settings to quiet third party
logs that are too verbose
logger.jetty1.name =
org.sparkproject.jetty
logger.jetty1.level = WARN
logger.jetty2.name =
org.sparkproject.jetty.util.compone
nt.AbstractLifeCycle
logger.jetty2.level = WARN
logger.replexprTyper.name =
org.apache.spark.repl.SparkIMain$ex
prTyper
logger.replexprTyper.level = INFO
logger.replSparkILoopInterpreter.na
me =
org.apache.spark.repl.SparkILoop$Sp
arkILoopInterpreter
logger.replSparkILoopInterpreter.le
vel = INFO
logger.parquet1.name =
org.apache.parquet
logger.parquet1.level = WARN
logger.parquet2.name = parquet
logger.parquet2.level = WARN

# SPARK-9183: Settings to avoid
annoying messages when looking up
nonexistent UDFs in SparkSQL with
Hive support
logger.RetryingHMSHandler.name =
org.apache.hadoop.hive.metastore.Re
```

```
tryingHMSHandler
logger.RetryingHMSHandler.level =
ERROR
logger.FunctionRegistry.name =
org.apache.hadoop.hive.ql.exec.Func
tionRegistry
logger.FunctionRegistry.level =
ERROR
logger.HiveConf.name =
org.apache.hadoop.hive.conf.HiveCon
f
logger.HiveConf.level = ERROR

# SPARK-327: Settings to suppress
the unnecessary warning message
from
MultiMechsAuthenticationHandler
logger.MultiMechsAuthenticationHand
ler.name =
org.apache.hadoop.security.authenti
cation.server.MultiMechsAuthenticat
ionHandler
logger.MultiMechsAuthenticationHand
ler.level = ERROR
logger.KerberosAuthHandler.name =
org.apache.hadoop.security.authenti
cation.server.KerberosAuthHandler
logger.KerberosAuthHandler.level =
ERROR

# SPARK-575: Settings to suppress
the unnecessary warning message
from AuthenticationFilter
logger.AuthenticationFilter.name =
org.apache.hadoop.security.authenti
cation.server.AuthenticationFilter
logger.AuthenticationFilter.level
= ERROR

logger.NativeCodeLoader.name =
org.apache.hadoop.util.NativeCodeLo
ader
logger.NativeCodeLoader.level =
ERROR
logger.YarnClient.name =
org.apache.spark.deploy.yarn.Client
logger.YarnClient.level = ERROR
logger.HiveUtils.name =
org.apache.spark.sql.hive.HiveUtils
logger.HiveUtils.level = ERROR
logger.HiveMetastore.name =
org.apache.hadoop.hive.metastore.Hi
veMetastore
logger.HiveMetastore.level = ERROR
logger.ObjectStore.name =
org.apache.hadoop.hive.metastore.Ob
jectStore
logger.ObjectStore.level = ERROR
logger.SQLCompleter.name =
org.apache.hive.beeline.SQLComplete
r
```

```
logger.SQLCompleter.level = ERROR

# SPARK-945: Setting to suppress exception when non-cluster admin can not read ssl-server config
logger.Configuration.name = org.apache.hadoop.conf.Configuration
logger.Configuration.level = ERROR

# Hide Spark netty rpc error when driver is finished
logger.Dispatcher.name = org.apache.spark.rpc.netty.Dispatcher
logger.Dispatcher.level = ERROR

# For deploying Spark ThriftServer
# SPARK-34128: Suppress undesirable TTransportException warnings involved in THRIFT-4805
appender.console.filter.1.type = RegexFilter
appender.console.filter.1.regex = .*Thrift error occurred during processing of message.*
# Hide fips specific properties initialization
appender.console.filter.1.regex = .*org.bouncycastle.jsse.provider.PropertyUtils.*
appender.console.filter.1.onMatch = deny
appender.console.filter.1.onMismatch = neutral
```

3. Set the following configurations on the `custom_log4j.properties` file.

   ```
   logger.SparkLogger.name = org.apache.spark
   logger.SparkLogger.level = INFO
   ```

   and

4. Create an interactive session by setting the Spark configuration with `spark.executor.extraJavaOptions` as key and `-Dlog4j.configuration=file:/local:///mounts/shared-volume/custom_log4j.properties` as value. See Creating Interactive Sessions on page 394.

5. Submit the Livy statements. See Submitting Statements on page 395.

Result:

The logs for the executor pod are now available.

Superset

Describes how to identify and debug issues for Superset.

Superset UI

If more than one user needs to access the same browser, the logged-in user must explicitly log out before another user can access the UI. Failure to explicitly log out results in caching and dashboard permission issues if multiple users try to access the same UI.

User Interface

Provides insights that can help you identify and resolve potential challenges when interacting with the HPE Ezmeral Unified Analytics Software UI.

Cannot Sign-In to HPE Ezmeral Unified Analytics Software After Reboot

If you cannot sign in to HPE Ezmeral Unified Analytics Software after the nodes reboot, you may have encountered a Postgres issue that impacts the `postgres-keycloak` pod. The Postgres issue causes the service pods that interact with the `postgres-keycloak` pod to get stuck in a `CrashLoopBackOff` state, which then prevents you from signing in to HPE Ezmeral Unified Analytics Software. Service pods that interact with the `postgres-keycloak` pod include `keycloak`, `airflow`, `ezpresto`, `superset`, `spire`, `ezuser`, and `token-service`.

Workaround

Before you follow the workaround steps, confirm that this is the issue preventing sign-in.

Run the following command to see the `keycloak-postgres` pod logs:

```
kubectl -n keycloak logs keycloak-postgresql-0
```

Check the `keycloak-postgres` pod logs for the following Postgres error:

`PANIC: replication checkpoint has wrong magic …`

If you see this Postgres error, complete the following workaround steps to recover Postgres from Keycloak and then restart the pods.

1. Delete the `checkpoint` file:
    a. To get the volume name, run:

    ```
    VOLUMENAME=$(kubectl -n keycloak get pvc
    data-keycloak-postgresql-0 -o jsonpath='{.spec.volumeName}')
    ```

    b. To get the volume path, run:

    ```
    VOLUMEPATH=$(kubectl get pv $VOLUMENAME -o
    jsonpath='{.spec.csi.volumeAttributes.volumePath}')
    ```

    c. To remove the `checkpoint` file, run:

    ```
    kubectl -n dataplatform exec admincli-0 -- hadoop fs -rm -f
    "$VOLUMEPATH/data/pg_logical/replorigin_checkpoint"
    ```

2. Restart the `keycloak` and `oauth2-proxy` pods:

a. To restart the `keycloak-postgresql-0` pod, run the following command and then wait for the pod to restart:

```
kubectl -n keycloak delete pod keycloak-postgresql-0
```

b. To restart the `keycloak-0` pod, run the following command and then wait for the pod to restart:

```
kubectl -n keycloak delete pod keycloak-0
```

c. To restart the `oauth2-proxy` pod, run the following command and then wait for the pod to restart:

```
kubectl -n oauth2-proxy delete pod -l app=oauth2-proxy
```

Installer UI Does Not Show Ingress Gateway Node IP Addresses (Required for DNS A Record Configuration)

When you install HPE Ezmeral Unified Analytics Software, you need the ingress gateway node IP addresses to configure your DNS A record for access to the HPE Ezmeral Unified Analytics Software domain. However, due to a port conflict that can occur between the istio ingress gateway and OpenShift ingress, the HPE Ezmeral Unified Analytics Software installer UI may not display the ingress gateway node IP addresses after the installation completes. When this conflict occurs, the istio ingressgateway pods go into a `pending` state, and the function that retrieves the node IPs returns empty.

To resolve this issue, complete the following steps:

1. SSH into the HPE Ezmeral Unified Analytics Software master node or have access to the `kubeconfig` of the HPE Ezmeral Unified Analytics Software cluster.

2. To identify the nodes running the OpenShift ingress pods, run:

```
kubectl get pod -n   openshift-ingress -o wide
```

Example

```
[core@master0 ~]$ kubectl get pod -n   openshift-ingress -o wide
NAME                             READY   STATUS    RESTARTS   AGE
IP               NODE                        NOMINATED NODE   READINESS
GATES
router-default-b6c47bcf6-5nlcm   1/1     Running   0          37h
10.227.209.134   worker2.pooja.ezfab.local   <none>           <none>
router-default-b6c47bcf6-w7vcv   1/1     Running   0          38h
10.227.209.135   worker3.pooja.ezfab.local   <none>           <none>
[core@master0 ~]$
```

3. To identify the nodes with the `"ezkf.hpe.com/ingress-gateway=true"` label, run:

   ```
   kubectl get node --show-labels | grep ingress-gateway
   ```

   NOTE: The conflict occurs on the node(s) running OpenShift ingress pods that also have the `"ezkf.hpe.com/ingress-gateway=true"` label.

   Example

   ```
   [core@master0 ~]$ kubectl get node --show-labels | grep ingress-gateway
   worker2.pooja.ezfab.local   Ready    worker                 46h
   v1.25.16+5c97f5b   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/
   os=linux,ezkf.hpe.com/ingress-gateway=true,hpe.com/compute=true,hpe.com/
   dataplatform=true,hpe.com/exclusivecluster=none,hpe.com/
   pin-dataplatform-cldb=true,hpe.com/pin-dataplatform-zk=true,hpe.com/
   status=available,hpe.com/usenode=true,kubernetes.io/
   arch=amd64,kubernetes.io/
   hostname=worker1.pooja.ezfab.local,kubernetes.io/
   os=linux,node-role.kubernetes.io/worker=,node.openshift.io/
   os_id=rhcos,nvidia.com/gpu.deploy.operands=true
   worker4.pooja.ezfab.local   Ready    worker                 46h
   v1.25.16+5c97f5b   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/
   os=linux,ezkf.hpe.com/ingress-gateway=true,hpe.com/compute=true,hpe.com/
   dataplatform=true,hpe.com/exclusivecluster=none,hpe.com/
   status=available,hpe.com/usenode=true,kubernetes.io/
   arch=amd64,kubernetes.io/
   hostname=worker4.pooja.ezfab.local,kubernetes.io/
   os=linux,node-role.kubernetes.io/worker=,node.openshift.io/
   os_id=rhcos,nvidia.com/gpu.deploy.operands=true
   ```

   In this example output, the conflicting node is `worker2.pooja.ezfab.local` because this node has the OpenShift ingress pod running and also has the label. The `"ezkf.hpe.com/ingress-gateway=true"` label on this node must be deleted and then added to a different worker node.

4. On each node with the conflict, complete the following steps:

   a. To remove the label, run:

      ```
      kubectl label node <nodename> ezkf.hpe.com/ingress-gateway-
      ```

      NOTE: The `-` at the end of `ingress-gateway-` is included to indicate deletion of the label.

b. To list all worker nodes, run:

```
kubectl get node
```

Example

```
[core@master0 ~]$ kubectl get node
NAME                        STATUS   ROLES                  AGE
VERSION
master0.pooja.ezfab.local   Ready    control-plane,master   46h
v1.25.16+5c97f5b
master1.pooja.ezfab.local   Ready    control-plane,master   46h
v1.25.16+5c97f5b
master2.pooja.ezfab.local   Ready    control-plane,master   46h
v1.25.16+5c97f5b
worker0.pooja.ezfab.local   Ready    worker                 46h
v1.25.16+5c97f5b
worker1.pooja.ezfab.local   Ready    worker                 46h
v1.25.16+5c97f5b
worker2.pooja.ezfab.local   Ready    worker                 46h
v1.25.16+5c97f5b
worker3.pooja.ezfab.local   Ready    worker                 46h
v1.25.16+5c97f5b
worker4.pooja.ezfab.local   Ready    worker                 46h
v1.25.16+5c97f5b
worker5.pooja.ezfab.local   Ready    worker                 46h
v1.25.16+5c97f5b
```

c. Identify one worker node that does not have the ingress gateway or the `"ezkf.hpe.com/ingress-gateway=true"` label and then run the following command to add the label to that node:

```
kubectl label --overwrite node <nodename>  ezkf.hpe.com/
ingress-gateway=true
```

In the example, the following nodes qualify as worker nodes that could have the `"ezkf.hpe.com/ingress-gateway=true"` label added:

```
worker0.pooja.ezfab.local
worker1.pooja.ezfab.local
worker5.pooja.ezfab.local
```

5. To verify that there are two running `istio-ingressgateway` pods, run:

```
kubectl get pod -n istio-system
```

NOTE: An OpenShift cluster with three worker nodes has only one `istio-ingressgateway` pod in the `Running` state.

6. To get the `ingressgateway` IPs, run:

```
kubectl -n istio-system get pod -l app=istio-ingressgateway -o
jsonpath='{.items[*].status.hostIP}'
```

Use the IP address to configure the DNS A records.

Sign-Out or Timeout Impact on Open Applications in the User Interface (UI)

If you explicitly sign out of theHPE Ezmeral Unified Analytics SoftwareUI or the system signs you out because the session hit the idle duration limit (30 mins), you may face potential disruptions to open applications within the same session with the following message:

```
Bad Message 431
```

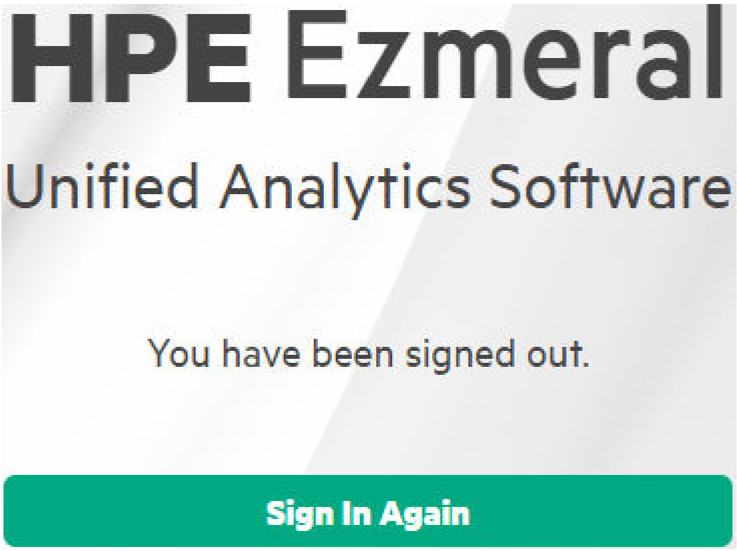
```
Reason: Request Header Fields Too Large
```

For a seamless experience, follow these recommendations:

- If you are not actively using theHPE Ezmeral Unified Analytics SoftwareUI, close the UI tab to prevent an automatic sign out.
- When intentionally signing out, close all application tabs opened through the UI.
- If the `Bad Message 431` error occurs, resolve it by selectively clearing the cookies linked to the HPE Ezmeral Unified Analytics Software cluster. Deleting all cookies is not required.

Sign In Again Returns a 403 Error

When you sign out of HPE Ezmeral Unified Analytics Software, a **Sign In Again** button appears on the page.

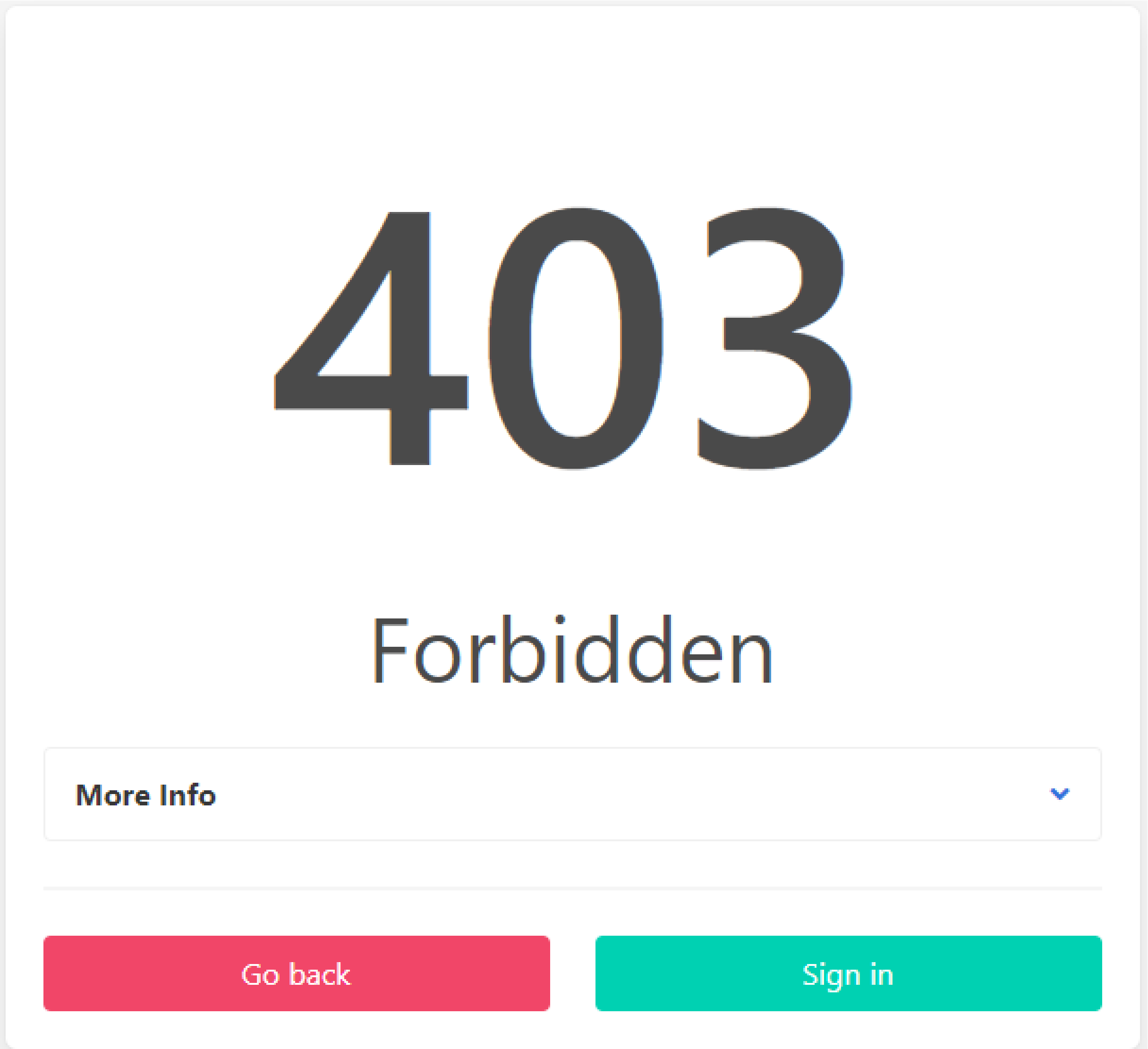


HPE Ezmeral

Unified Analytics Software

You have been signed out.

The **Sign In Again** option is active for fifteen minutes and then times out. After timeout, clicking **Sign In Again** returns a 403 error.



Under the 403 error, click **Sign In** and then click **Go Back** to get back to the HPE Ezmeral Unified Analytics Software Home page. If this does not work, close the browser and sign in to HPE Ezmeral Unified Analytics Software in a new browser window.

Cannot Upload File to Data Fabric

In the Unified Analytics UI, when you go to **Data Engineering > Data Sources** and try to upload a file to Data Fabric on the **Data Volumes** tab, the upload fails and the system returns the following message:

```
Failed to upload file.
```

This issue occurs when the filename includes unsupported special characters, such as &.

To resolve this issue, verify that the filename only contains the following supported special characters:

- Dash (**-**) (Can be used in Unified Analytics version 1.4.1 and later.)
- Dot (**.**)
- Underscore (**_**)
- A single empty space

If the filename contains unsupported special characters and/or more than one empty space, rename the file and upload it again.

Realm Not Enabled Message

If you are signed in to HPE Ezmeral Unified Analytics Software and the license has either expired or has not been uploaded, the URL redirects you to Keycloak instead of redirecting you to the page where you can upload a license, and Keycloak displays the following message:

```
Realm not enabled
```

To resolve this issue, go to the base URL, for example `https://home.company-ua.com` and upload a license. Do not include any remaining path after the base URL, for example `https://home.company-ua.com/home`.

Licensing

Provides insights that can help you identify and resolve potential challenges with HPE Ezmeral Unified Analytics Software licensing.

GPU Workloads Not Running

HPE Ezmeral Unified Analytics Software will not run GPU workloads unless there is a valid GPU license applied. Currently, there is no specific alert for this issue in the the UI; however, you can use `kubectl` commands to troubleshoot licensing issues.

Viewing GPU licensing labels

Run the following command to get the GPU-related licensing labels:

```
(reverse-i-search)'jq': kubectl get
node <node-ip-address> -o json | jq
'.metadata.labels'
```

The following table lists the GPU-related licensing labels to identify in the output:

Label	Description
`nvidia.com/gpu.count`	This label provides the GPU count exposed by NVIDIA. This is the number of virtual GPU devices. For example, if you have your A100 split into two MIG devices, they are reflected in the GPU count. The number reflects the MIG partitions.
`hpe.com@.nvidia.com/gpucount`	HPE Ezmeral Unified Analytics Software introduced this label. The label provides the actual physical GPU count on that node.

Viewing the number of GPUs in use

Run the following command to view the number of GPUs currently in use. You can also see which applications and notebooks are running when you run this command:

```
kubectl describe <node-ip-address>
```

The output shows the number of GPUs currently in use. You can also see which applications and notebooks are running on the node.

Product Version and Lifecycle Support

Provides information about software and component support.

Software Support and Lifecycle Status

Shows the support and lifecycle status for all versions of HPE Ezmeral Unified Analytics Software.

As soon as possible, upgrade to the latest version of HPE Ezmeral Unified Analytics Software for new features, usability enhancements, bug fixes, and active support. If your installed version of the product is in the **In Maintenance** status, you no longer have active support and you have limited time to plan and upgrade to the latest version.

Software Lifecycle and Maintenance Dates

The following table shows release date, lifecycle status, in-maintenance, and end-of-life dates for HPE Ezmeral Unified Analytics Software.

Version	Release Date	Lifecycle Status	In Maintenance	End of Life
1.5.x	August 1, 2024	Active	January 31, 2025	April 30, 2025
1.4.x	May 1, 2024	Active	October 31, 2024	January 31, 2025
1.3.0	Feb 1, 2024	In Maintenance	N/A	October 31, 2024
1.2.0	Nov 1, 2023	End of Life	N/A	July 31, 2024
1.1.0	August 1, 2023	End of Life	N/A	April 30, 2024

More information

Support Matrix on page 201
The tables on this page show the tools and frameworks, HPE Ezmeral Data Fabric, operating system versions, and GPU models that are supported for HPE Ezmeral Unified Analytics Software releases.

Support Matrix

The tables on this page show the tools and frameworks, HPE Ezmeral Data Fabric, operating system versions, and GPU models that are supported for HPE Ezmeral Unified Analytics Software releases.

Primary Storage Support

HPE Ezmeral Unified Analytics Software supports the following storage platforms for primary storage:

HPE Ezmeral Unified Analytics Software	HPE Ezmeral Data Fabric
1.5.0, 1.5.2	7.6.1[1], 7.7.0, 7.8.0

[1] HPE Ezmeral Data Fabric 7.6.0 is deprecated and HPE Ezmeral Data Fabric 7.6.1 is a patch release that replaces release 7.6.0, see Deprecation of Release 7.6.0.

To learn more about the primary storage support in HPE Ezmeral Unified Analytics Software, see Preparing HPE Ezmeral Data Fabric to be Primary Storage for HPE Ezmeral Unified Analytics Software on page 78.

Tools and Frameworks

HPE Ezmeral Unified Analytics Software supports the following Tools & Frameworks in the versions listed:

HPE Ezmeral Unified Analytics Software	Airflow	EzPresto	Feast	Kubeflow	Livy	MLflow	HPE MLDE	Ray	Spark Applications	Spark History Server	Spark Operator	Superset
1.5.0, 1.5.2	2.9.2	0.287	0.39.0	1.8.0	0.8.0.5	2.13.2	0.28.1	2.24.0	3.5.1	3.5.1	1.3.8.7-hpe	4.0.1
1.4.1	2.8.4	0.285	0.34.1	1.8.0	0.8.0	2.10.2	0.28.1	2.9.3	3.5.0	3.5.0	1.3.8.7-hpe	3.1.0
1.4.0	2.7.3	0.285	0.34.1	1.8.0	0.8.0	2.10.2	0.28.1	2.9.3	3.5.0	3.5.0	1.3.8.7-hpe	3.1.0
1.3.0	2.7.3	0.281	0.34.1	1.8.0	0.8.0	2.8.1	0.26.7	2.7.0	3.5.0	3.5.0	1.3.8.4-hpe	3.0.1
1.2.0	2.7.0	0.281	0.34.1	1.7.0	0.7.0	2.7.1	N/A	2.6.1	3.4.1 3.3.1	3.4.1	1.3.8.3-hpe	2.1.1
1.1.0	2.6.1	0.269	0.31.0	1.7.0	0.7.0.302	2.4.0	N/A	2.4.0	3.4.0 3.3.1	3.4.0	1.3.8.2-hpe	2.1.0
1.0.0	2.5.1	0.269	0.29.0	1.6.0	0.7.0	2.1.1	N/A	2.2.0	3.3.1	3.3.1	1.3.8.2-hpe	2.0.1

Notebook Images

The following table lists the default notebook images and their packages in HPE Ezmeral Unified Analytics Software 1.5.2:

Notebook Images	Libraries	General Packages
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezua-1.5.2-175c1e34`	• SciPy 1.11.3 •	• Conda Python 23.3.1-1 • JupyterLab 3.6.6 • kfp 2.5.0 • kfp-pipeline-spec 0.2.2 • kfp-server-api 2.0.5 • feast 0.33.1 • mlflow 2.13.2 • kserve 0.11.2 • kubeflow-katib 0.16.0 • kubeflow-kale 0.7.0 • presto-python-client 0.8.3 • ray 2.24.0 The following packages are installed in a separate Ray kernel, • ray[tune] 2.24.0 • ray[default] 2.24.0 • ray[client] 2.24.0 • ray[serve] 2.24.0
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-full:ezua-1.5.2-175c1e34`	• PyTorch (CPU) • torch 2.1.0 • torchvision 0.16.0 • torchaudio 2.1.0 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-cuda-full:ezua-1.5.2-175c1e34`	• PyTorch (CUDA) • cuda 12.1 • torch 2.1.0 • torchvision 0.16.0 • torchaudio 2.1.0 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezua-1.5.2-175c1e34`	• Tensorflow (CPU) • tensorflow 2.13.0 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezua-1.5.2-175c1e34`	• TensorFlow (CUDA) • tensorflow 2.13.0 • cuda 11.8 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-data-science:ezua-1.5.2-175c1e34`	•	
`gcr.io/mapr-252711/kubeflow/notebooks/codeserver:ezua-1.5.2-175c1e34`	• code-server 4.17.1 (Visual Studio Code) • Python extension 2023.18.0 **NOTE:** This is not the Python language version. •	

The following table lists the default notebook images and their packages in HPE Ezmeral Unified Analytics Software 1.5.0:

Notebook Images	Libraries	General Packages
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezua-1.5.0-d2d18b79`	• SciPy 1.11.3 •	• Conda Python 23.3.1-1 • JupyterLab 3.6.6 • kfp 2.5.0 • kfp-pipeline-spec 0.2.2 • kfp-server-api 2.0.5 • feast 0.33.1 • mlflow 2.13.2 • kserve 0.11.2 • kubeflow-katib 0.16.0 • kubeflow-kale 0.7.0 • presto-python-client 0.8.3 • ray 2.24.0 The following packages are installed in a separate Ray kernel, • ray[tune] 2.24.0 • ray[default] 2.24.0 • ray[client] 2.24.0 • ray[serve] 2.24.0
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-full:ezua-1.5.0-d2d18b79`	• PyTorch (CPU) • torch 2.1.0 • torchvision 0.16.0 • torchaudio 2.1.0 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-cuda-full:ezua-1.5.0-d2d18b79`	• PyTorch (CUDA) • cuda 12.1 • torch 2.1.0 • torchvision 0.16.0 • torchaudio 2.1.0 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezua-1.5.0-d2d18b79`	• Tensorflow (CPU) • tensorflow 2.13.0 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezua-1.5.0-d2d18b79`	• TensorFlow (CUDA) • tensorflow 2.13.0 • cuda 11.8 •	
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-data-science:ezua-1.5.0-d2d18b79`	•	
`gcr.io/mapr-252711/kubeflow/notebooks/codeserver:ezua-1.5.0-d2d18b79`	• code-server 4.17.1 (Visual Studio Code) • Python extension 2023.18.0 **NOTE:** This is not the Python language version. •	

To learn more about descriptions and uses of Notebook images, see Notebook Images Overview on page 435.

HPE Ezmeral Data Fabric

The following table lists the versions of HPE Ezmeral Data Fabric that you can connect HPE Ezmeral Unified Analytics Software to externally:

HPE Ezmeral Unified Analytics Software	HPE Ezmeral Data Fabric
1.5.0, 1.5.2	• 7.6.1[1], 7.7.0, 7.8.0
1.4.0, 1.4.1	• 7.4.0, 7.5.0, 7.6.1[1], 7.7.0
1.3.0	• 7.4.0, 7.5.0, 7.6.1[1]
1.2.0	• 6.2.0, 7.0.0, 7.1.0, 7.2.0, 7.3.0, 7.4.0
1.1.0	• 6.2.0, 7.0.0, 7.2.0
1.0.0	• 7.0.0, 7.2.0

[1] HPE Ezmeral Data Fabric 7.6.0 is deprecated and HPE Ezmeral Data Fabric 7.6.1 is a patch release that replaces release 7.6.0, see Deprecation of Release 7.6.0.

Operating System

HPE Ezmeral Unified Analytics Software supports the following operating systems in the versions listed:

HPE Ezmeral Unified Analytics Software	RHEL Version	Rocky Version
1.5.0, 1.5.2	• 8.9[1] • 8.8[1]	• 8.7[2]
1.4.0, 1.4.1	• 8.9[1] • 8.8[1]	• 8.7[2]
1.3.0	• 8.8[1]	• 8.7[2]
1.2.0	• 8.x[1]	• 8.x[2]
1.1.0	• 8.x[1]	• 8.x[2]

[1] Only RHEL 8.x is supported on GPU hosts.

[2] There is no GPU support for Rocky 8.x, as NVIDIA does not support the GPU operator running on Rocky 8.x.

GPU Models

HPE Ezmeral Unified Analytics Software supports the following GPU models:

HPE Ezmeral Unified Analytics Software	GPU Model
1.5.0, 1.5.2	• NVIDIA A100 • NVIDIA A30 • NVIDIA H100-NVL, H100-PCI • NVIDIA A10G[1] • NVIDIA L40S[1] • NVIDIA V100[1]
1.4.0, 1.4.1	• NVIDIA A100 • NVIDIA A30 • NVIDIA H100-NVL, H100-PCI • NVIDIA A10G[1] • NVIDIA L40S[1] • NVIDIA V100[1]
1.3.0	• NVIDIA A100 • NVIDIA A30
1.2.0	• NVIDIA A100
1.1.0	• NVIDIA A100

[1] The A10G, L40S, and V100 GPU models do not support MIG. During the installation of HPE Ezmeral Unified Analytics Software, you must specify GPU partition size as *Whole* and request the number of GPU instances required for the workload. These GPU models are not split into any partitions and you will get the entire physical GPU. In this configuration, applications can use only one virtual GPU at a time.

To learn more about the GPU support in HPE Ezmeral Unified Analytics Software, see GPU Support on page 137.

Upgrade Bundle Images

The following table lists the upgrade bundle images necessary for upgrading HPE Ezmeral Unified Analytics Software to the following versions:

HPE Ezmeral Unified Analytics Software	Images
1.5.2	`ezua/ezua-system-bundle:v1.5.2-0b8e1c03e-86d69a`
1.4.1	`ezua/ezua-system-bundle:v1.4.1-bffae884e`
1.4.0	`ezua/ezua-system-bundle:v1.4.0-e8061214f`

Release Notes (1.5.0)

This document provides a comprehensive overview of the latest updates and enhancements in HPE Ezmeral Unified Analytics Software (version 1.5.0), including new features, improvements, bug fixes, and known issues.

HPE Ezmeral Unified Analytics Softwareprovides software foundations for enterprises to develop and deploy end-to-end data and advanced analytics solutions from data engineering to data science and machine learning across hybrid cloud infrastructures – delivered as a software-as-a-service model.

New Features

This release includes the following new features:

Support for External Storage Platforms	HPE Ezmeral Unified Analytics Software now integrates with external storage platforms, eliminating the internal data fabric as primary storage. This integration leverages existing storage solutions for a seamless and scalable data management experience while reducing the amount of resources required to deploy an HPE Ezmeral Unified Analytics Software cluster. It also enhances high availability (HA) efficiency to ensure a fully operational cluster after recovery from a power outage or reboot. HPE Ezmeral Unified Analytics Software currently supports HPE Ezmeral Data Fabric as primary storage with support for additional storage solutions coming in subsequent releases. For details, see Primary Storage, Preparing HPE Ezmeral Data Fabric to be Primary Storage for HPE Ezmeral Unified Analytics Software on page 78, and Installing on User-Provided Hosts (Connected and Air-gapped Environments) on page 63.
MAPRSASL Authentication for Hive Metastore	You can now configure a Hive data source in HPE Ezmeral Unified Analytics Software to use MAPRSASL for authentication with the Hive Metastore on HPE Ezmeral Data Fabric. This enhancement ensures secure access and integration, providing an added layer of security for data management. For additional details, see Using MAPRSASL to Authenticate to Hive Metastore on HPE Ezmeral Data Fabric on page 338.

Enhancements

This release includes the following enhancements:

Flexibility in Tools and Frameworks Installation	You now have the option to deploy a subset of tools and frameworks during installation, and the flexibility to install the other tools and frameworks later. You can exclude the following tools and frameworks from the initial installation of HPE Ezmeral Unified Analytics Software: • Superset • EzPresto • Livy • MLDE • Feast

	The tools and frameworks that you choose not to install initially can be installed any time. For additional details, see Installing Included Frameworks Post Unified Analytics Installation on page 130.
UI for Adding Volumes	A new user interface is now available for connecting to external storage platforms, allowing you to use them as data sources for applications and frameworks in your HPE Ezmeral Unified Analytics Software cluster. The UI supports integration with HPE Ezmeral Data Fabric and GreenLake for File Storage, providing a seamless and user-friendly way to access diverse data sources. *Note that with this change, the Data Fabrics option previously under Administration in the left navigation panel has been moved to the Data Volumes tab.* For additional details, see Connecting to HPE Ezmeral Data Fabric on page 133 and Connecting to HPE GreenLake for File Storage on page 135.
Revoke User Access on Data Sources	Administrators can revoke user access to data sources within the **Data Engineering** section of the UI. This functionality allows for easy management of user privileges, ensuring secure access to both structured and object store data. For additional details, see Revoking Member Access to Data.
Run CTAS Queries with Hive Discovery Metastore	The Hive Discovery Metastore now supports running CTAS (CREATE TABLE AS SELECT) queries on CSV and parquet files stored in the HPE Ezmeral Data Fabric file system or S3 object storage, including HPE Ezmeral Data Fabric S3, MinIO S3, and AWS S3. You can also insert data into the created tables. To utilize this feature, set up a Hive data source connection with the specified parameters, as described in Hive Discovery Metastore Connection Parameters on page 314. Use schema discovery for CSV files, delta discovery for delta files, and include the format in the query for parquet files.
Installation Configuration Review	Before finalizing the installation of HPE Ezmeral Unified Analytics Software on your cluster, you can review and adjust the installation configuration details on the **Review** screen. This feature ensures accuracy and customization of the setup process.
Seamless Deletion of Imported Tools and Frameworks	You can now automatically delete a chart from the Charmuseum when an `ezappconfig` custom resource (CR) is deleted. This feature simplifies the management of imported tools and frameworks by ensuring that associated configurations and resources are removed seamlessly.

Resolved Issues

This release includes numerous fixes that enhance system security, stability, and performance, including the following resolutions:

Permission denied error when submitting the Kubeflow pipeline while using the Kubeflow notebook images	Submitting a Kubeflow pipeline using the KFP SDK V2 Kubeflow notebook images no longer returns a permission denied error.

The driver pod of the cloned Spark job remains in the container creating state

When you use the **Clone** option to create a new Spark application with a similar configuration as an existing Spark application, the driver pod of the cloned Spark job no longer remains in the container creating state.

Permission denied error when installing packages while using the Kubeflow notebook images

Installing the Kubeflow notebook images (with KFP SDK V2) provided by HPE Ezmeral Unified Analytics no longer returns a permission denied error.

Replace Fluent Bit with OTEL for log collection and parsing

Log collection and parsing now uses Open Telemetry (OTEL) instead of Fluent Bit, which reduces resource consumption (memory).

Unable to download infrastructure and application services logs

You can download the infrastructure and application services logs without issue.

Unable to delete Data Fabric connection due to "Secret not found" error

You can delete Data Fabric connections by deleting the Data Volume source.

Uploading a term license

Uploading a term license no longer results in an ezlicense controller pod crashloopbackoff error.

Activation code change no longer results in a crashloopbackoff error

The activation code change that caused a crashloopbackoff error when a capacity license was applied before upgrading is resolved.

Known Issues

The following sections describe known issues with workarounds where applicable:

EzPresto installation fails due to mysql pod entering CrashLoopBackOff state

During EzPresto deployment, the HPE Ezmeral Unified Analytics Software installation fails due to slow disk I/O, which leads to the mysql pod in EzPresto entering a CrashLoopBackOff state.

Workaround: To resolve this issue, see EzPresto installation fails due to mysql pod entering CrashLoopBackOff state on page 158.

Installation pre-check fails if the SSH key does not have a passphrase

If you use an SSH key file, the SSH key must have a passphrase; otherwise, the installation pre-check fails and installation cannot occur. You can set the passphrase to any value, even a dummy value.

Running CTAS against a Hive data source fails with ORC file error

Running a CTAS query against a Hive data source that is configured to use MAPRSASL authentication fails with the following error:

```
Error creating ORC file. Error
getting user info for current user,
presto.
```

This issue occurs if the HPE Ezmeral Data Fabric ticket was generated with impersonation enabled uids and impersonation was not enabled when the Hive data source connection was configured in HPE

Ezmeral Unified Analytics Software. For example, the ticket was created as shown:

```
maprlogin
generateticket -user pa -type
servicewithimpersonationandticket \
-impersonateduids 112374829 -out
pa.out
```

Workaround: To resolve this issue, delete the Hive data source connection and create a new Hive data source connection, making sure to include the following options in addition to the other required options:

- Select the **Hive HDFS Impersonation Enabled** option.
- Enter the principal/username that Presto will use when connecting to HPE Ezmeral Data Fabric in the **Hive Hdfs Presto Principal** field. If this field is not visible, perform a search for it in the **Hive Advanced Settings** search field.

For additional information, see Using MAPRSASL to Authenticate to Hive Metastore on HPE Ezmeral Data Fabric on page 338.

CTAS query on Hive Metastore in HPE Ezmeral Data Fabric fails

For Hive connections that authenticate to HPE Ezmeral Data Fabric via MAPRSASL, running a CTAS query against HPE Ezmeral Data Fabric returns the following error:

```
Database 'pa' location does not
exist:<file_path>
```

Workaround: To resolve this issue, create and upload a configuration file that points to the HPE Ezmeral Data Fabric cluster, as described in Using MAPRSASL to Authenticate to Hive Metastore on HPE Ezmeral Data Fabric on page 338.

The Hive connection to HPE Ezmeral Data Fabric exists after deleting files

Deleting the cluster details and tickets from the `mapr-clusters.conf` and `maprtickets` files does not terminate the Hive connection to HPE Ezmeral Data Fabric. Users can still create new Hive connections to HPE Ezmeral Data Fabric and run queries against HPE Ezmeral Data Fabric. This issue occurs because HPE Ezmeral Unified Analytics Software caches the HPE Ezmeral Data Fabric files.

Workaround: After you delete the cluster details and tickets from the `mapr-clusters.conf` and `maprtickets` files, restart the EzPresto pods. To restart the pods, run:

```
kubectl rollout restart
statefulset -n ezpresto
ezpresto-sts-mst

kubectl rollout restart
```

```
statefulset -n ezpresto
ezpresto-sts-wrk
```

Optional Fields display by default when connecting an Iceberg data source

When adding Iceberg as a data source, the UI lists all possible connection fields (mandatory and optional) instead of listing the mandatory connection fields only.

EzPresto does not release memory when a query completes

EzPrestoretains allocated memory after query completion for subsequent queries because of an open-source issue (https://github.com/prestodb/presto/issues/15637). For example, if a query uses 10GB of memory, EzPresto does not release the memory when the query completes and then uses it for the next query. If the next query requires additional memory, for instance, 12GB, EzPresto accumulates an extra 2GB and does not release it after query completion. For assistance, contact HPE support.

Configuration changes to long-running pods are not applied in Ray

Configuration changes or upgrades to long-running pods in Ray, such as adjusting resource capacities or expanding persistent volume (PV) storage are not applied in Ray.

Workaround

To ensure successful configuration changes or upgrades, manually delete relevant pods after the reconfiguration or upgrade process. For details, see https://github.com/ray-project/kuberay/issues/527.

Worker nodes do not automatically spawn with `JobSubmissionClient` in the Ray cluster

When submitting jobs to the Ray cluster using `JobSubmissionClient`, worker nodes do not spawn automatically.

Workaround

To ensure proper functionality when submitting Ray jobs using `JobSubmissionClient`, you must manually specify entry point resources as follows:

- For CPU, set `entrypoint_num_cpus` to 1
- For GPU, set `entrypoint_num_gpus` to 1

For details, see Using JobSubmissionClient to Submit Ray Jobs on page 423.

HPE is actively engaging with the community to address this open-source issue (https://github.com/ray-project/ray/issues/42436).

NVIDIA GPU cannot enforce SELinux

Due to a known NVIDIA GPU issue (https://github.com/NVIDIA/gpu-operator/issues/553), SELinux cannot be enforced for GPU deployments.

Workaround

Set GPU hosts to either `disabled` or `permissive` mode until this issue is resolved.

Ray dashboard UI

A known Ray issue prevents the Ray Dashboard UI from displaying the GPU worker group details correctly. To see updates regarding resolution and to learn more, seehttps://github.com/ray-project/ray/issues/14664.

Upgrade on OpenShift cluster

If you want to perform an in-place upgrade of HPE Ezmeral Unified Analytics Software on an Openshift cluster, contact HPE support for assistance to ensure a smooth transition and to address any potential complexities that can arise during the upgrade process.

Installation

Before you install or upgrade, HPE recommends that you back up your data.

- To installHPE Ezmeral Unified Analytics Software(version 1.5.0), seeInstalling on User-Provided Hosts (Connected and Air-gapped Environments) on page 63.
- To upgrade HPE Ezmeral Unified Analytics Software to version 1.5.0, contact HPE Support.

If you encounter any issues during or after the installation or upgrade process, please contact HPE Support. We appreciate your feedback and strive to continually enhance your product experience.

Additional Resources

- Documentation
- Release note archives:
  - 1.4.1 Release Notes
  - 1.4.0 Release Notes
  - 1.3.0 Release Notes
  - 1.2.0 Release Notes

Thank you for choosingHPE Ezmeral Unified Analytics Software. Enjoy the new features and improvements introduced in thisrelease.

Release Notes (1.5.2)

This document provides a comprehensive overview of the latest updates and enhancements in HPE Ezmeral Unified Analytics Software (version 1.5.2), including improvements, bug fixes, and known issues.

HPE Ezmeral Unified Analytics Softwareprovides software foundations for enterprises to develop and deploy end-to-end data and advanced analytics solutions from data engineering to data science and machine learning across hybrid cloud infrastructures – delivered as a software-as-a-service model.

Enhancements

This release includes several security and stability fixes.

Resolved Issues

This release includes the following resolutions:

Slow disk I/O no longr causes the EzPresto installation to fail due to the mysql pod entering a CrashLoopBackOff state

During EzPresto deployment, the HPE Ezmeral Unified Analytics Software installation no longer fails due to slow disk I/O, which led to the mysql pod in EzPresto entering a CrashLoopBackOff state.

Using a Capacity License for KFP SDK V2 no longer causes Katib jobs to fail

Katib jobs can successfully launch in environments that use a capacity license for KFP SDK V2.

Installation pre-check no longer fails if the SSH key does not have a passphrase

If you use an SSH key file, you do not have to provide a dummy passphrase to pass the installation pre-check.

Configuration changes to long-running pods are now applied in Ray

Configuration changes or upgrades to long-running pods in Ray, such as adjusting resource capacities or expanding persistent volume (PV) storage are now applied in Ray.

Known Issues

The following sections describe known issues with workarounds where applicable:

Delayed metrics data when the HPE Ezmeral Coordinator node reboots

When the HPE Ezmeral Coordinator node reboots, it can take up to 20 minutes for the system to reestablish the on-premises to cloud connection. Once the connection is established, all metrics data is sent.

The system allows you to create object storage connections with a bucket name in the endpoint URL

Users cannot access object storage when the data source connection is created with a bucket name in the endpoint URL, for example `https://s3.us-test-2.amazonaws.com/bucket1`. To resolve this issue, delete the data source connection and create a new connection with an endpoint URL that does not include a bucket name, for example `https://s3.us-test-2.amazonaws.com`.

Katib jobs fail when launched through Kale

If you launch a Katib job through Kale from a notebook, the Katib job fails because resource limits are not provided. Pods get stuck in a pending state and the system returns a warning message stating that resource limits must be defined.

Workaround: To work around this issue:

1. Download the following file and put it in the `/mnt/user` directory:

   ```
   kale-katib.patch
   ```

2. Open a notebook terminal and run the following command:

   ```
   cd /opt/conda/lib/python3.11/
   site-packages
   ```

3. From the notebook terminal, run the following command:

   ```
   git apply /mnt/user/
   kale-katib.patch
   ```

4. Close all the open notebook tabs and shut down all the kernels running in notebooks.

5. In the top menu bar, select **File > Log Out**.
6. Log in again.

Packages created with %createKernel are not available on the new kernel

When you run the `%createKernel` magic function, installed packages may not display on the kernel; however, you can see the installed packages by running `conda list` on the terminal. Some default packages, for example pandas, installed while creating a new kernel may not be available.

Spark application does not run without vCPU driver and executor values set

If you do not set the vCPU driver and executor values when you create a Spark application, the application cannot run and remains in a pending state. These values specify the amount of capacity that the Spark application can consume from the license.

Application status does not change after configuration update

When you modify an application (through the **Tools & Frameworks** page) and click **Configure**, the application status remains in the **Ready** state when it should change to **Updating**.

Workaround: To resolve this issue, refresh the **Tools & Frameworks** page.

Installation fails during OpenTelemetry phase

Slow disk speed can cause an intermittent timing issue that prevents the certificate from being propagated to a webhook before the timeout duration, which then causes the installation to fail during the OpenTelemetry phase.

Submitting an MLflow job from a notebook intermittently returns a ValueError

Submitting an MLflow job from a notebook can intermittently return the following ValueError:

```
ValueError: numpy.dtype size changed,
may indicate binary incompatibility.
Expected 96 from C header, got 88
from PyObject command terminated with
exit code 1
```

Workaround: To resolve this issue, restart the notebook and submit the MLflow job again.

Cannot access HPE GreenLake for File Storage S3 buckets from Livy

HPE Ezmeral Unified Analytics Software users with the member role cannot access buckets in HPE GreenLake for File Storage S3 object storage from Livy when read and write access permissions are granted on the buckets.

Tiles do not appear for imported tools and frameworks

Tiles for imported tools and frameworks do not immediately appear on the **Tools & Frameworks** page after you import a tool or framework. You must refresh the page to see the tiles for imported tools and frameworks.

Cannot create Iceberg connections with hadoop catalog type from the UI

You must create Iceberg connections with hadoop catalog type from the command line using a `curl` command that posts the configuration in JSON format. For details, see EzPresto on page 157.

SQL Client and Query Editor return incorrect results for bigint data type

The SQL Client and the Query Editor return incorrect results for the bigint data type by rounding up the last few digits of large numbers. For example, if you run the following query:

```
SELECT 714341252076979033 LIMIT 1
```

The SQL Client and the Query Editor return `714341252076979100` when they should return `714341252076979033`.

To work around this issue, use the CAST() function to cast the number, column, or expression to VARCHAR, for example:

```
SELECT CAST('714341252076979033' AS
VARCHAR) LIMIT 1
```

Running CTAS against a Hive data source fails with ORC file error

Running a CTAS query against a Hive data source that is configured to use MAPRSASL authentication fails with the following error:

```
Error creating ORC file. Error
getting user info for current user,
presto.
```

This issue occurs if the HPE Ezmeral Data Fabric ticket was generated with impersonation enabled uids and impersonation was not enabled when the Hive data source connection was configured in HPE Ezmeral Unified Analytics Software. For example, the ticket was created as shown:

```
maprlogin
generateticket -user pa -type
servicewithimpersonationandticket \
-impersonateduids 112374829 -out
pa.out
```

Workaround: To resolve this issue, delete the Hive data source connection and create a new Hive data source connection, making sure to include the following options in addition to the other required options:

- Select the **Hive HDFS Impersonation Enabled** option.
- Enter the principal/username that Presto will use when connecting to HPE Ezmeral Data Fabric in the **Hive Hdfs Presto Principal** field. If this field is not visible, perform a search for it in the **Hive Advanced Settings** search field.

For additional information, see Using MAPRSASL to Authenticate to Hive Metastore on HPE Ezmeral Data Fabric on page 338.

CTAS query on Hive Metastore in HPE Ezmeral Data Fabric fails

For Hive connections that authenticate to HPE Ezmeral Data Fabric via MAPRSASL, running a CTAS query against HPE Ezmeral Data Fabric returns the following error:

```
Database 'pa' location does not
exist:<file_path>
```

Workaround: To resolve this issue, create and upload a configuration file that points to the HPE Ezmeral Data Fabric cluster, as described in Using MAPRSASL to Authenticate to Hive Metastore on HPE Ezmeral Data Fabric on page 338.

The Hive connection to HPE Ezmeral Data Fabric exists after deleting files

Deleting the cluster details and tickets from the `mapr-clusters.conf` and `maprtickets` files does not terminate the Hive connection to HPE Ezmeral Data Fabric. Users can still create new Hive connections to HPE Ezmeral Data Fabric and run queries against HPE Ezmeral Data Fabric. This issue occurs because HPE Ezmeral Unified Analytics Software caches the HPE Ezmeral Data Fabric files.

Workaround: After you delete the cluster details and tickets from the `mapr-clusters.conf` and `maprtickets` files, restart the EzPresto pods. To restart the pods, run:

```
kubectl rollout restart
statefulset -n ezpresto
ezpresto-sts-mst

kubectl rollout restart
statefulset -n ezpresto
ezpresto-sts-wrk
```

Optional Fields display by default when connecting an Iceberg data source

When adding Iceberg as a data source, the UI lists all possible connection fields (mandatory and optional) instead of listing the mandatory connection fields only.

EzPresto does not release memory when a query completes

EzPrestoretains allocated memory after query completion for subsequent queries because of an open-source issue (https://github.com/prestodb/presto/issues/15637). For example, if a query uses 10GB of memory, EzPresto does not release the memory when the query completes and then uses it for the next query. If the next query requires additional memory, for instance, 12GB, EzPresto accumulates an extra 2GB and does not release it after query completion. For assistance, contact HPE support.

Worker nodes do not automatically spawn with `JobSubmissionClient` in the Ray cluster

When submitting jobs to the Ray cluster using `JobSubmissionClient`, worker nodes do not spawn automatically.

Workaround

To ensure proper functionality when submitting Ray jobs using `JobSubmissionClient`, you must manually specify entry point resources as follows:

- For CPU, set `entrypoint_num_cpus` to 1

	• For GPU, set `entrypoint_num_gpus` to 1 For details, see Using JobSubmissionClient to Submit Ray Jobs on page 423. HPE is actively engaging with the community to address this open-source issue (https://github.com/ray-project/ray/issues/42436).
NVIDIA GPU cannot enforce SELinux	Due to a known NVIDIA GPU issue (https://github.com/NVIDIA/gpu-operator/issues/553), SELinux cannot be enforced for GPU deployments. **Workaround** Set GPU hosts to either `disabled` or `permissive` mode until this issue is resolved.
Ray dashboard UI	A known Ray issue prevents the Ray Dashboard UI from displaying the GPU worker group details correctly. To see updates regarding resolution and to learn more, seehttps://github.com/ray-project/ray/issues/14664.
Upgrade on OpenShift cluster	If you want to perform an in-place upgrade of HPE Ezmeral Unified Analytics Software on an Openshift cluster, contact HPE support for assistance to ensure a smooth transition and to address any potential complexities that can arise during the upgrade process.

Installation

Before you install or upgrade, HPE recommends that you back up your data.

- To installHPE Ezmeral Unified Analytics Software version 1.5.2, seeInstalling on User-Provided Hosts (Connected and Air-gapped Environments) on page 63.
- To upgrade HPE Ezmeral Unified Analytics Software to version 1.5.2, see Upgrading HPE Ezmeral Unified Analytics Software on page 101.

If you encounter any issues during or after the installation or upgrade process, please contact HPE Support. We appreciate your feedback and strive to continually enhance your product experience.

Additional Resources

- Documentation
- Release note archives:
  - 1.5.0 Release Notes
  - 1.4.1 Release Notes
  - 1.4.0 Release Notes
  - 1.3.0 Release Notes
  - 1.2.0 Release Notes

Thank you for choosingHPE Ezmeral Unified Analytics Software. Enjoy the new features and improvements introduced in thisrelease.

Term Licensing

Describes term licensing for HPE Ezmeral Unified Analytics Software.

Term licensing is a capacity-based licensing model that can be purchased for one, three, or five years. Term licenses are based on vCPU and GPU capacity, with the option to procure vCPU and GPU licenses separately.

A term license provides the applications and frameworks that run in an HPE Ezmeral Unified Analytics Software cluster with a specific amount of vCPU and GPU resources. The system manages and accounts for services separately, ensuring that license capacity is only consumed by applications and frameworks.

Because a term license is strictly for capacity, it does not impact which applications and frameworks run. Users can run the applications and frameworks packaged with HPE Ezmeral Unified Analytics Software or third-party applications and frameworks.

An administrator can install HPE Ezmeral Unified Analytics Software on any size cluster and apply a term license after activation. Every application and framework that runs consumes resources (vCPU or GPU) which are deducted from the license. If usage exceeds the license capacity, the system audits the event and raises alerts to notify users. Any applications or frameworks submitted remain in a pending state until capacity is made available, either by purchasing additional licensing capacity or freeing up unused resources held by other applications and frameworks. Any in progress applications and frameworks continue to run if they secured sufficient resources to complete; otherwise, they stop running and sit in a pending state until capacity is made available.

Adding Capacity

You can add capacity either by purchasing additional licensing capacity or freeing up unused resources held by other applications and frameworks.

You can purchase additional licensing capacity through the MY HPE SOFTWARE CENTER customer portal and then complete the following steps to upload the license to Unified Analytics:

1. Sign in to the Unified Analytics UI.
2. In the left navigation panel, select **Administration > Settings**.
3. On the **Settings** page, select the **Activation Key** tab and then click **Upload Activation Key**.

Comparing vCPU and GPU Licenses

The following table describes vCPU and GPU licensing:

License	Description
vCPU	• A unit of CPU (physical, virtual, or hyperthreaded) as enumerated by HPE Ezmeral Unified Analytics Software. • The vCPU licenses are counted toward cores on HPE Ezmeral Unified Analytics Software worker nodes. • Cores on the HPE Ezmeral Coordinator and HPE Ezmeral Unified Analytics Software master nodes are not counted toward the license capacity.

License	Description
GPU	• A term license counts against physical GPU devices. • A GPU refers to the whole GPU device. MIG partitions do not count toward GPU licenses. A license applies to the whole GPU device. • The number of GPU licenses required is proportional to the number of GPU devices on a GPU card. • GPU licenses are required to activate GPU-based applications and frameworks; however, GPU licenses also require HPE Ezmeral Unified Analytics Software with vCPU as a prerequisite.

Restrictions and Limitations

Term licenses have the following restrictions and limitations:

- A term license applies to a single HPE Ezmeral Unified Analytics Software cluster. You can add additional vCPU and GPU term licenses to expand cluster capacity, but you cannot apply the license across multiple HPE Ezmeral Unified Analytics Software clusters.
- Each term license is tied to an HPE Ezmeral Coordinator. Mixed consumption and term licenses are not allowed on HPE Ezmeral Unified Analytics Software clusters that are managed by the same HPE Ezmeral Coordinator.
- If you are using a Not-For-Resale (NFR) license, you must delete the NFR license before you apply a production license.
- You must have a valid vCPU license before you can apply a GPU license.
- You must apply an HPE Machine Learning Development Environment (MLDE) license before you run workloads on your cluster. The MLDE term license is distinct and must be procured separately from the HPE Ezmeral Unified Analytics Software term license. For more MLDE licensing information, visit HPE Machine Learning Development Environment Software or contact your HPE account team.

License Capacity Aggregation and Reinforcement

Term licenses adhere to the following capacity aggregation and reinforcement rules:

- Total licensed capacity (vCPU, GPU) is aggregated across all valid vCPU and GPU licenses applied to the cluster. For example, if you apply two vCPU licenses, each with 100 vCPU capacity, the total license capacity is 200 vCPUs. If one license expires, the total capacity becomes 100 vCPUs.
- GPU is counted by physical GPU devices in a cluster. From the command line, you can see the GPU count exposed by NVIDIA and actual physical GPU count on a node. See Licensing on page 200.
- The vCPU capacity on GPU worker nodes counts toward the licensed vCPU capacity.
- GPU applications and frameworks only run when there is a valid GPU license; otherwise, the GPU applications and frameworks sit idle in a pending state.
- To comply with contract terms, you must apply additional licenses when license capacity has been exceeded. The HPE Ezmeral Unified Analytics Software UI displays an alert if total worker-node capacity exceeds the license capacity.

Related concepts

Audit Logging on page 288
Describes auditing in HPE Ezmeral Unified Analytics Software and how to access audit logs.

More information

Metering and Billing on page 258
Describes metering and billing in HPE Ezmeral Unified Analytics Software.

Alerting on page 263
Describes alerting in HPE Ezmeral Unified Analytics Software.

Viewing Alerts and Notifications on page 276
Describes how to view alerts and notifications in HPE Ezmeral Unified Analytics Software.

Importing Frameworks and Managing the Application Lifecycle on page 117
Describes how to import, manage, and secure tools and frameworks in HPE Ezmeral Unified Analytics Software.

Security

Describes security in HPE Ezmeral Unified Analytics Software.

Identity and Access Management

Describes identity and access management in HPE Ezmeral Unified Analytics Software.

HPE Ezmeral Unified Analytics Software uses Keycloak as its OIDC provider for identity and access management. Keycloak secures access to HPE Ezmeral Unified Analytics Software and applications through authorization, authentication, and SSO protocols. Users authenticate to Keycloak instead of authenticating to multiple application services.

The following steps describe the basic access flow for a user signing in to HPE Ezmeral Unified Analytics Software application services:

1. A user goes to the application URL with their web browser.
2. If the user has not yet signed in to an application in this cluster, the user's browser is redirected to a sign-in page that is managed by the cluster's Keycloak instance.
3. The user enters their credentials (username and password) at the sign-in page.
4. Keycloak verifies the user's credentials against those in the organization's AD/LDAP server.
5. If the provided credentials are valid, the user's browser is redirected to the originally requested application URL. The browser receives one or more cookies. The cookies represent active sessions with Keycloak and the application.
6. The application is (through a secure back channel) provided with an access token that encapsulates the user's authentication and their authorized roles within the cluster.
7. The application internally uses the access token to determine the user's identity and authorization. Some applications may also use this token to communicate with other services within the cluster.

Once a user signs in to HPE Ezmeral Unified Analytics Software, SSO enables the user to seamlessly switch between different authentication-requiring application services while the session is valid. For example, the user can open the Feast application without reentering their credentials. However, if the user signs out of HPE Ezmeral Unified Analytics Software and then tries to access the Feast endpoint URL, the OIDC provider (Keycloak) prompts the user to reenter their credentials.

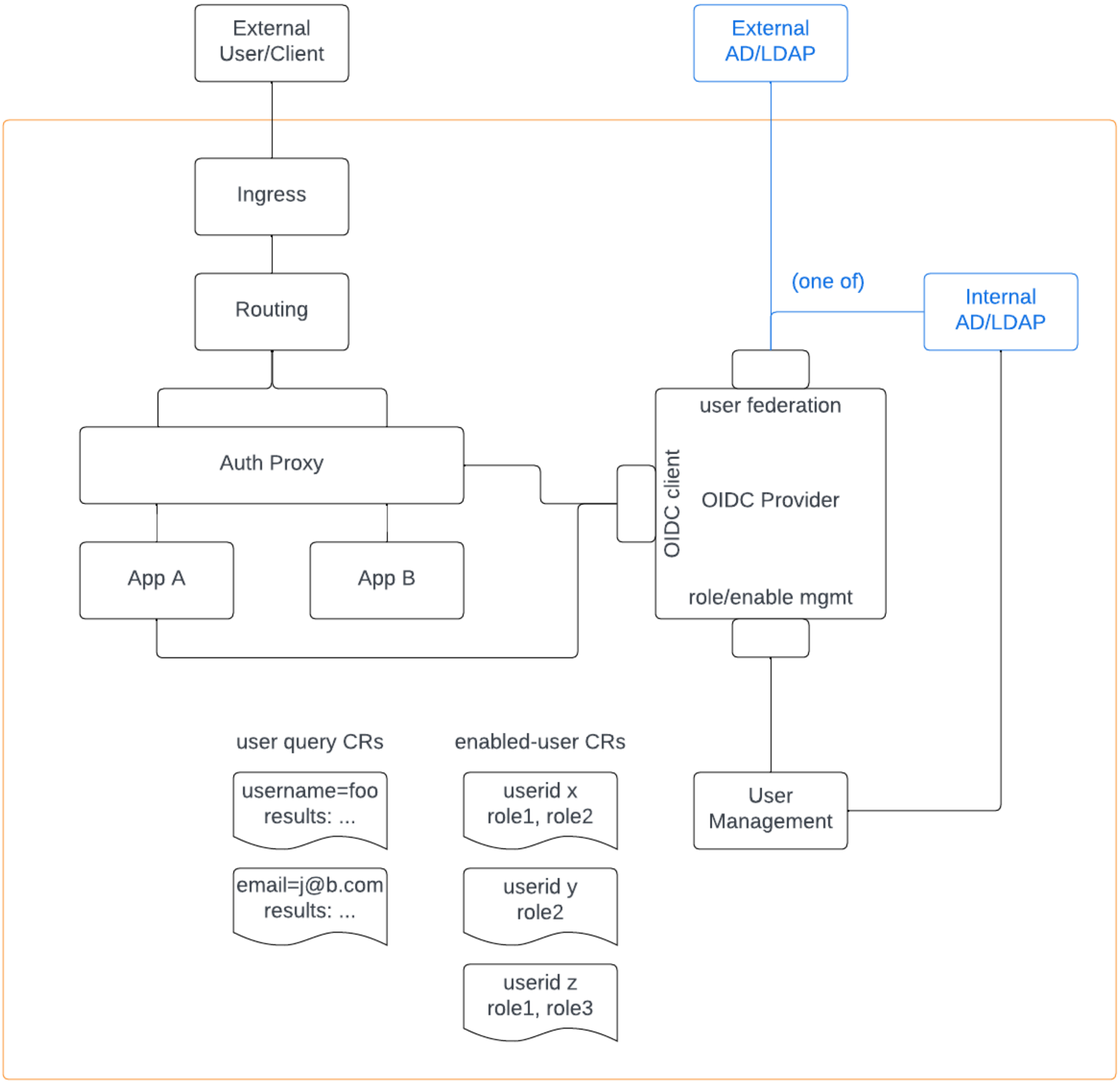
If the browser is left idle in the main interface for more than one hour, the user is automatically signed out. If more than one week has passed since the user has authenticated, the user must re-enter credentials.

Architecture

The following diagram shows two access flows - one for application A and one for application B. Application A is an OIDC-native application that understands how to integrate with a provider such as Keycloak for user authentication and authorization. Application B is not an OIDC-native application.

The auth proxy interacts with Keycloak to ensure that access to application B is only available to authenticated users. The proxy also provides information about user identity and roles to application B through HTTP headers.

Note that although application A is OIDC-native, it also sits behind the auth proxy. This ensures that, regardless of how the application itself manages sessions and access tokens, a user will be immediately blocked from accessing the application if an admin has revoked the user's cluster access.



The following sections describe the components in the access flow diagram:

Ingress	Istio provides the service mesh, request routing, policy enforcement, and the proxies used to intervene in service requests.

The Istio Ingress gateway performs TLS termination for all incoming traffic and validates JSON Web Tokens (JWTs) issued by Keycloak. External client access to application services is TLS-terminated at the Istio Ingress gateway, then routed to internal service endpoints with mutual TLS encryption. Internal service communications also use TLS.

Communication to internal services (from the gateway or from applications) is policy-restricted to a set of allowed clients. The clients are identified by SPIFFE credentials. Istio and SPIRE manage the SPIFFE credentials.

Routing

Istio routes traffic from the Ingress gateway to the appropriate application service based on the DNS name destination of the traffic. During HPE Ezmeral Unified Analytics Software installation, the administrator can set up a DNS domain that includes the entire sub-domain DNS (sub-domain wild card A record) to route all domain traffic to the Ingress of the application environment.

Auth Proxy (Oauth2 Proxy)

Oauth2 Proxy gates access to applications that are not OIDC aware. It gives those applications information about the user's token and claims in the token by inserting header values (individual claim values as well as the entire token). The primary header values populated by the proxy are:

- Authorization, from "Bearer" prefixed to the entire token in JSON Web Token (JWT) format
- X-Auth-Request-Preferred-Username, from the preferred_username claim
- X-Auth-Request-Email, from the email claim
- X-Auth-Request-Groups, from the groups claim

(Some additional headers are populated with the same username and groups values for backwards compatibility reasons.)

Oauth2 Proxy is also used with OIDC-native apps in order to promptly and universally enforce administrative revocation of user access.

Oauth2 Proxy hooks into application traffic through Istio authorization policies. The Istio authorization policy forces traffic to go through the proxy before accessing services in HPE Ezmeral Unified Analytics Software.

OIDC Client

An OIDC client provides a set of API endpoints used for interactions with the OIDC provider, such as authenticating users.

The OIDC client instance used by browser-accessed applications in an HPE Ezmeral Unified Analytics Software cluster is represented by the ID `ua` and a unique generated secret. This secret is passed to application installation scripts during initial setup, then stored in a Kubernetes secret for later use in deploying applications that you import into HPE Ezmeral Unified Analytics Software.

For any OIDC-native application that integrates with this OIDC client, Keycloak must be configured to be aware of an application-specific "callback URL" that will be used as part of the OIDC flow. For applications imported after initial setup, you must modify Keycloak's list of allowed callback URLs using the Keycloak web interface or REST API.

A separate OIDC client with the ID `ua-grant` (no client secret) is available, which can be used from a CLI or program to directly exchange user credentials for tokens. This client implements the *resource owner password credentials grant* flow, or what Keycloak documentation calls *Direct Access Grant*.

The `ua-grant` OIDC client is used for two main purposes, both of which apply to REST APIs (or other non-browser service endpoints) exposed to out-of-cluster users:

- If the service requires token-based authentication, the out-of-cluster caller can use the `ua-grant` client to obtain a token which is then provided to the service. Note that it is the caller's responsibility to securely store and otherwise manage the token.
- If the service requires username/password authentication, perhaps because of constraints from existing service clients, the service can use the `ua-grant` client internally to validate the user and also obtain a token that can be used to communicate with other cluster services.

OIDC Provider (Keycloak)

Keycloak sources user information from the internal or external AD/LDAP directory. Keycloak imports user data from the AD/LDAP server on an hourly basis. The following user attributes are mapped from the AD/LDAP server to Keycloak:

- username
- email
- full name

NOTE: The specific attribute names representing these three items are provided in the AD/LDAP configuration details when the HPE Ezmeral Unified Analytics Software is installed.

Users authenticate with Keycloak instead of authenticating with individual applications. Keycloak assigns a special Keycloak ID to each user and supplies applications with tokens in JWT format. Each token contains *claims* that describe the user's authenticated identity and other attributes.

The claims mapped from the AD/LDAP user attributes, respectively, are:

- preferred_username
- email

	• name, given_name, and family_name (The latter two formed by splitting "name" at the first space.) The token also contains a *groups* claim. This claim contains a list of the user's group memberships that are important to Keycloak or to applications. Currently, the only application-significant group is *admin*, which is present in the groups list if the user has been designated as an Administrator of the cluster. For additional information about Keycloak, including how to access the Keycloak Admin Web Console, refer to the *Keycloak Admin Web Console* section in Security on page 182.
Internal/External AD/LDAP	See AD/LDAP Servers on page 227 and Working with Certs and the Truststore on page 230.
User Management (Management Operator)	An administrator manages users through the HPE Ezmeral Unified Analytics Software UI; for example, creating users and assigning roles. These operations result in the creation of custom Kubernetes resources (representing queries and user configuration) that are processed by the backend user management service. This service has credentials for the Keycloak administrative REST API, the Kubernetes API, and (if applicable) the internal LDAP server. Tasks performed by this service include: • Accessing the internal LDAP server to create and delete users. • Marking a user in Keycloak to enable or disable their ability to authenticate into the cluster. • Assigning roles to users in Keycloak.

User Isolation

Describes user isolation in HPE Ezmeral Unified Analytics Software.

When an HPE Ezmeral Unified Analytics Software administrator adds a new user to HPE Ezmeral Unified Analytics Software, the system automatically assigns each user a user-designated workspace. User-designated workspaces isolate each user's applications and objects from other users in the cluster. If a user wants to share their work, they can do so by setting access controls directly on the objects they create or by changing the namespace in which their applications run.

HPE Ezmeral Unified Analytics Software bundles applications with different isolation mechanisms and assurances. For example, HPE Ezmeral Unified Analytics Software bundles cloud-native applications and open-source web applications. Cloud-native applications such as Kubeflow use namespaces to isolate users, whereas web applications such as open-source Airflow and Superset require customized changes to the open-source code to support user isolation and roles in HPE Ezmeral Unified Analytics Software. Customization entails mapping the HPE Ezmeral Unified Analytics Software user role (member or admin) to permissions in the open-source applications.

The following table summarizes user isolation in HPE Ezmeral Unified Analytics Software with regard to HPE Ezmeral Unified Analytics Software user roles (admin and member) and application permission mappings, as well as the result of changing user roles and deleting users on applications and objects:

	MLflow	**Airflow**	**Superset**	**Spark**
Admin	• Assumes admin role • View/Edit access on all experiments • Does not have personal models or experiments	• Assumes admin role • View/Edit access on all DAGs • Does not have personal DAGs	• Assumes admin role • View/Edit access on all dashboards, datasets, and charts • Does not have personal dashboards	• N/A (no role hierarchy in Spark) • Can only view/ access personal Spark jobs
Member	• Assumes member role • Can only view/ access personal experiments • No access to other users' experiments and models	• Assumes custom role (segregated) • Must explicitly define own role when creating DAGs to keep private; otherwise, DAGs are shared	• Assumes customized Alpha role with added permissions to create database connections • Must explicitly define own role when creating DAGs to keep private; otherwise, DAGs are shared • Can view all dashboards and create charts based on all dashboards. • Cannot edit the dashboards	• N/A (no role hierarchy in Spark; similar to Kubeflow) • Can only view/ access personal Spark jobs
Running in user namespace	N/A	Yes	N/A	Yes
User role propagation	Yes	Yes	Yes	N/A (no role hierarchy in Spark)
User deletion	Objects remain untouched; only admins have access	DAGs remain untouched; only admins have access	Objects remain untouched; only admins have access	Jobs are removed with the user namespace

IMPORTANT: Do not modify user roles or permissions in the applications that users access through HPE Ezmeral Unified Analytics Software. Modifying roles or permissions directly in an application can break the mapping between the HPE Ezmeral Unified Analytics Software user role and application permission setting. For example, do not assign an HPE Ezmeral Unified Analytics Software member the Admin role in the Superset application. If you want a user to have admin-level permissions in Superset, add the admin role to the user directly in HPE Ezmeral Unified Analytics Software. Changing a user’s role to admin in HPE Ezmeral Unified Analytics Software grants the user access to the Administration settings in HPE Ezmeral Unified Analytics Software. To edit a user role, see Adding and Removing Users on page 233 .

The following topics describe user isolation in more detail for each of the applications that curenntly support user isolation:

- Defining RBACs on MLflow Experiments on page 417
- Defining RBACs on DAGs on page 360
- Defining RBACs in Superset on page 365
- Running Spark Applications in Namespaces on page 405

User Roles

Describes roles that you can assign to users in HPE Ezmeral Unified Analytics Software.

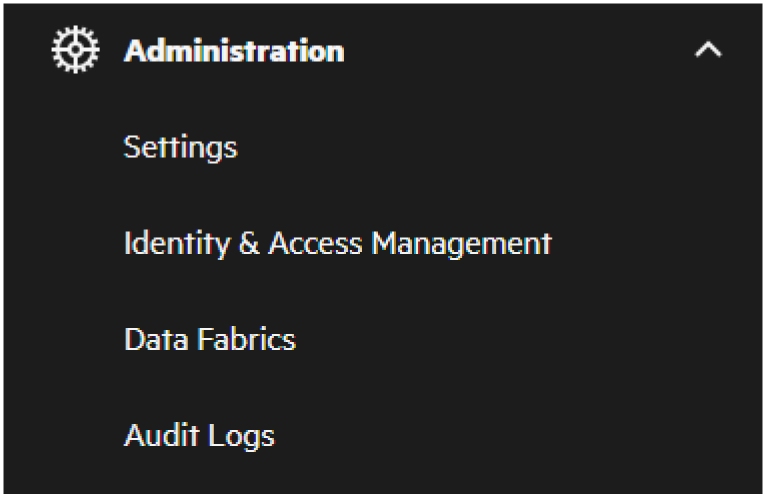
In HPE Ezmeral Unified Analytics Software, a user is either a member or an administrator. The user that installs HPE Ezmeral Unified Analytics Software and applies the license is the platform administrator. After applying the license, the administrator is prompted to sign in using the credentials entered during installation. Once signed in, the administrator can add users. See Adding and Removing Users on page 233.

Any user added and assigned the *admin* role can also add and remove users. Users that are not assigned the *admin* role are *members*. Members have access to all areas of HPE Ezmeral Unified Analytics Software except for the Administration area.

NOTE: Admins can only add users that are in the AD/LDAP server. The platform administrator configures AD/LDAP settings for HPE Ezmeral Unified Analytics Software during installation. For additional information, see Installation on page 63 and AD/LDAP Servers on page 227.

Administrators

Only users assigned the *admin* role can see and use the **Administration** area in the left navigation bar.



The following list describes the tasks that an admin can perform through the **Administration** options and provides links to additional information:

Settings	- Upload the activation key and activation code to activate services. See HPE Ezmeral Unified Analytics Software Service Activation and Billing Processes on page 92. - Update application container images. See Upgrading Included Frameworks on page 125. - Register an Otel endpoint or view the JDBC endpoint. See Configuring Endpoints on page 135 and Connecting External Applications to EzPresto via JDBC on page 345.
Identity & Access Management	- Add and remove users. See Adding and Removing Users on page 233.
Data Fabric	- Connect to HPE Ezmeral Data Fabric clusters. See Connecting to HPE Ezmeral Data Fabric on page 133.

Audit Logs	• View a chronological set of records that document the events that occur in an HPE Ezmeral Unified Analytics Software cluster. See Audit Logging on page 288.

Related concepts

Adding and Removing Users on page 233
Describes how administrators can add, remove users, and edit the role and password for users in HPE Ezmeral Unified Analytics Software.

Adding and Removing Users Programmatically on page 235
Describes how to add and remove users through the Kubernetes API using the EzUserQuery and EzUserConfig custom resources.

AD/LDAP Servers on page 227
Describes the differences between the internal OpenLDAP server in HPE Ezmeral Unified Analytics Software and external AD/LDAP servers. Also describes some of the server-related configuration options that you set during installation.

AD/LDAP Servers

Describes the differences between the internal OpenLDAP server in HPE Ezmeral Unified Analytics Software and external AD/LDAP servers. Also describes some of the server-related configuration options that you set during installation.

When you install HPE Ezmeral Unified Analytics Software, the configuration options vary depending on whether you use an external AD/LDAP (default and recommended) included with HPE Ezmeral Unified Analytics Software or the internal OpenLDAP server.

After installation, the designated administrator can sign in and grant users permission to access HPE Ezmeral Unified Analytics Software and assign roles. A user management operator running in HPE Ezmeral Unified Analytics Software sets up local resources for users, such as their user profile and workspace, and also enables access.

NOTE:

- SSO does not support applications that use AD/LDAP integration to validate credentials presented to an external service.
- The AD/LDAP server supports access by PLAIN (unsecured) LDAP, LDAPS, or StartTLS. Do not use PLAIN LDAP in production. If using LDAPS or StartTLS, one or more custom certificates may be needed to validate the server certificate. See Working with Certs and the Truststore on page 230.

The following sections describe the differences between internal and external AD/LDAP servers:

External AD/LDAP Server

When you configure an external directory during installation, HPE Ezmeral Unified Analytics Software references the external AD/LDAP server and gets users from it.

When you sign in to HPE Ezmeral Unified Analytics Software, you can search for users, grant access, and assign roles.

HPE Ezmeral Unified Analytics Software has the following external AD/LDAP server requirements:

- The AD/LDAP server must already exist.
- The AD/LDAP server must be network-accessible to the deployed HPE Ezmeral Unified Analytics Software instance.

- For AWS deployments, the AD/LDAP server must be accessible to the VPC where the HPE Ezmeral Unified Analytics Software instance runs.
- The AD/LDAP server must contain user objects with the required attributes. Any addition, removal, or modification of users and their attributes must be done at the AD/LDAP server.
  - The user objects on the external AD/LDAP server must have the following attributes:
    - Username
    - Fullname
    - UID
    - GID

    These attributes are required to federate users from the AD/LDAP server to Keycloak. User objects obtained from the direct AD/LDAP integration does not contain any role assignments and does not know which users are enabled to use HPE Ezmeral Unified Analytics Software.
- The AD/LDAP server contains the Group GID attribute on the group objects.

  NOTE: If the server is not Active Directory, you must specify the Group Name attribute for group objects.

When you configure the external AD/LDAP server during installation, you specify the following information:

- How to contact the LDAP server.
- How to bind to the server to find account information.
- Truststore for validating the server certificate.
- Information about how user objects are configured.

The following table lists and describes some of the AD/LDAP fields that you configure during installation:

Field	Description
Active Directory	If you do not select the Active Directory (AD) option, the possible schemas are more varied. You must enter additional information to properly describe the user and group objects.
Validation	The validation check boxes are for sanity checks before the installation starts and during the installation process. The validation can detect issues with the AD integration server before the installation is well underway. Only disable these options when running the installation container in an environment that cannot access AD.
Search Base DN	Must cover both user and group objects.
Security Protocol	If the security protocol is LDAPS or StartTLS, the server certificate will be validated. If the server certificate was signed by something other than a known public CA, a truststore must be provided. A truststore is a JKS file such as those created by the Java keytool utility. If a provided truststore is password protected, the truststore password must be supplied.

Field	Description
Username Attribute	Must contain the name of a user object attribute on the server that contains a username following some content rules: • Syntax is like POSIX except that a username cannot begin or end with a dot or underscore. • Can have capital letters, alphanumeric beginnings and endings, dots, dashes, and underscores, which are all valid for use within a 63-character limit. • Usernames are case-insensitive. If a username is bob, this user can sign in as BOB, Bob, or bob. Regardless of how bob signs in, the username always displays in lowercase (bob).
Fullname Attribute	Must contain the name of a user object attribute on the server that contains the user's full name. This is typically the `name` attribute on AD servers or `cn` on OpenLDAP servers.
Email Attribute	If the admin performing the installation selects the **Allow Login By Email Address** option, users can sign in using their email address or username; otherwise, users can only sign in with their usernames. Even if you do not select the option **Allow Login By Email Address**, you can still specify an email attribute for users, and their email addresses (if available) will be discovered for display purposes. However, in this scenario, users will not be able to sign in to Unified Analytics using their email address. Each user must have a unique email address. This is typically the `mail` attribute on AD or OpenLDAP servers.
UID Attribute	The user object attribute that is expected to contain an integer user ID value.
GID Attribute	The user object attribute that is expected to contain the integer value for this user's primary group ID.
Group GID	The group object attribute that is expected to contain an integer group ID value.
Group Name	The group object attribute that is expected to contain the group name.
Default Admin User	Must identify a user that already exists on the server. The value specified here should be the value of the **Username Attribute** on that user object.

Internal OpenLDAP Server

You can select **Use Internal LDAP Server** during installation and configure an internal directory. In HPE Ezmeral Unified Analytics Software, the internal directory setup is an OpenLDAP server. Only use the internal directory for POCs and demos; do not use the internal directory in production.

If you opt to use the internal directory, during installation you specify the following information to create the administrator in the system:

- username
- full name
- email
- password

The administrator is the initial user that signs in to HPE Ezmeral Unified Analytics Software to add other users and perform administrative tasks. Adding users creates the internal user directory.

When you remove a user, the user can no longer access the HPE Ezmeral Unified Analytics Software cluster, and the system clears the local resources. See Adding and Removing Users on page 233.

Related concepts

Adding and Removing Users on page 233
Describes how administrators can add, remove users, and edit the role and password for users in HPE Ezmeral Unified Analytics Software.

Working with Certs and the Truststore

Describes how to provide a truststore with a valid server certificate, including how to view and locate certs, as well as how to create and validate a truststore for certs.

When you use an external AD/LDAP server, Keycloak verifies the server certificate. If the certificate is not signed by a commonly known certificate authority (CA), you must provide a truststore with the information required to verify that the server certificate is valid.

Note the following guidelines and conditions related to certs and the truststore:

- A truststore is needed for StartTLS or LDAPS connections. Different applications with potentially different default trusted-certificate stores may need to verify the connection, so the means to verify the LDAP server's certificate must be explicitly provided.
- A truststore contains the certs required to finish the signing chain – from the issuing cert mentioned in the cert that the server presents to the trusted self-signed cert. In the case of a self-signed server cert, the chain is the server cert itself.
- Any truststore that you provide must contain all of the necessary certs. The truststore must be a chain of certs signed by certs that terminate in a self-signed cert.

Locating the Certs to put in the Truststore

If you do not know which certs need to go into the truststore (possibly due to IT protocols), use `openssl` to probe the server to see which certs are being presented by the AD/LDAP server.

The following commands create the files `myserver-cert1.pem, myserver-cert2.pem`, and so on. These are the certs presented by the server. The first one is the server cert, followed by any intermediate certs.

LDAPS server

The following command probes an LDAPS server running at `myserver.com` on `port 636`:

```
openssl s_client -showcerts -verify
10 -connect myserver.com:636 < \
    /dev/null | \
    awk '/BEGIN/,/END/{if(/BEGIN/) {a+
+}; out="myserver-cert"a".pem"; print
>out}'
```

StartTLS server

The following command probes an StartTLS server running at `myserver.com` on `port 389`:

```
openssl s_client -showcerts -verify
10 -connect
myserver.com:389 -starttls ldap < \
    /dev/null | \
    awk '/BEGIN/,/END/{if(/BEGIN/) {a+
+}; out="myserver-cert"a".pem"; print
>out}'
```

Getting the Issuer and Subject from a cert file

To get the Issuer and Subject from a cert file, run the following command:

```
openssl x509 -in
myserver-cert1.pem -text | grep '\
(Issuer\|Subject\)'
```

TIP:

- If there is only one cert and it refers to itself as Issuer, that means it is a self-signed server cert, and that server cert needs to go into the truststore.
- If there is a list of Issuers certs, there is typically one Issuer that does not have a match among the Subjects. That missing Issuer cert is the next link in the trust chain. You will need to get that cert either by way of the CA, your IT department, or whoever configured and runs the server. Often the missing Issuer is a custom root cert, in which case you only have one cert to put in your truststore.
- If the missing Issuer is not a root cert and is actually an intermediate cert, you will need to get the intermediate cert and also get the cert that the intermediate cert is signed by and continue this process until you get to the root (self-signed) cert.

Creating a Truststore

Build the truststore with the Java `keytool` utility by performing a series of cert imports.

Note the following points from the Keystore docs about accepted input cert formats:

- Keytool can import X.509 v1, v2, and v3 certificates, and PKCS#7 formatted certificate chains consisting of certificates of that type.
- The data to be imported must be provided either in binary encoding format or in printable encoding format (also known as Base64 encoding) as defined by the Internet RFC 1421 standard. In the latter case, the encoding must be bounded at the beginning by a string that starts with "-----BEGIN" and bounded at the end by a string that starts with "-----END"."

Importing a truststore and setting the password

Alias values in the commands are used for readability when dumping the truststore. You can use any alias you choose.

The first import creates the truststore. During the first import, you set the password for the truststore. Subsequent imports will ask for this password.

To import a self-signed server cert from the `servercert.pem` file, run the following command:

NOTE: This is the only command you have to run for a self-signed certificate.

```
keytool -importcert -alias selfsigned
\
    -file servercert.pem -keystore
truststore.jks
```

When asked if you want to trust it, respond with `yes`.

Importing a custom root cert and intermediate certs

ATTENTION: If you follow the instructions to import down the trust chain, you should *not* be asked whether any of the intermediate certs should be trusted because keytool should be aware of what cert they were signed by.
If you get that question when importing an intermediate cert, you may have missed a link in the chain or you are importing in the wrong order.

To import a custom root cert and intermediate certs, start by running the following command to import the custom root cert (`root.pem` in this example):

```
keytool -importcert -alias root \
    -file root.pem -keystore
truststore.jks
```

When asked if you want to trust it, respond with `yes`.

Importing intermediate certs

If you have intermediate certs to import, start with the one closest to the root, and work down the signing chain toward the server cert. If your first intermediate cert was signed by a default trusted cert, run the following command to import it (example filename `intermediate.pem`):

```
keytool -importcert -trustcacerts -ali
as intermediate \
    -file intermediate.pem -keystore
truststore.jks
```

For any intermediate cert signed by something previously imported into your truststore, run the following command to import it *without* the trustcacerts argument:

```
keytool -importcert -alias
intermediate \
    -file intermediate.pem -keystore
truststore.jks
```

Validating a Truststore

Run the command appropriate for your server type and then press **enter** to kill the connection. If the validation is successful, the system returns the following message:

```
Verify return code: 0 (ok)
```

If the truststore is not correct, the system returns the following message:

```
Verify return code: 20 (unable to get local issuer certificate)
```

LDAPS

To validate a truststore, run the following command:

NOTE: The following example validates that a truststore named `truststore.jks` with password `mypass` works for an LDAPS server running at `myserver.com` port 636 as follows:

```
openssl s_client -verify 10 -connect
myserver.com:636 \
    -CAfile
<(keytool -list -rfc -keystore
truststore.jks -storepass mypass)
```

StartTLS

To validate a truststore, run the following command:

NOTE: The following example validates that a truststore named `truststore.jks` with password `mypass` works for an LDAPS server running at `myserver.com` port 389 as follows:

```
openssl s_client -verify 10 -connect
myserver.com:389 -starttls ldap \
    -CAfile
<(keytool -list -rfc -keystore
truststore.jks -storepass mypass)
```

Adding and Removing Users

Describes how administrators can add, remove users, and edit the role and password for users in HPE Ezmeral Unified Analytics Software.

The user search field is only enabled for HPE Ezmeral Unified Analytics Software installations configured to use an external AD/LDAP directory. The search field does not work for installations using the internal OpenLDAP configuration; however, an admin can still add new users.

Adding Users for an External AD/LDAP Directory

For an *external* AD/LDAP directory, complete the following steps to add a user:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Administration > Identity & Access Management**.
3. In the search field, enter a substring search on the user's username or email ID and then enable HPE Ezmeral Unified Analytics access. The search for the user's username is case insensitive. You can also assign the admin role if you want the user to have administrative access in HPE Ezmeral Unified Analytics Software.

IMPORTANT:

- When you do not specify the **Email Attribute** for external AD/LDAP directory during HPE Ezmeral Unified Analytics Software installation, you will not see the **Email** column on the **Identity and Access Management** screen.
- When you specify the value for **Email Attribute**, the value is syntax-checked with compliance to RFC 5322 and RFC 6532. Keycloak only allows the valid email address. If Keycloak rejects the value for **Email Attribute**, the value is changed to `invalid-<unified-analytics-username>@<unified-analytics-domain>`. You can see the value for the updated email address in the **Identity and Access Management** screen.

  When signing in to Unified Analytics using your email address, you must use the updated email address visible on the **Identity and Access Management** screen. You cannot sign in with the email address specified during installation, which Keycloak identified as having bad syntax.

Adding Users for the Internal OpenLDAP Directory

For the *internal* OpenLDAP directory, complete the following steps to add a user:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Administration > Identity & Access Management**.
3. Click **Add User**.
4. In the drawer that opens, enter the following information:
   - **Username** - Enter the username.

     You cannot add a user with a username that is a case-insensitive match of an existing username. For example, if you have a user named "bob," you cannot add a user named "Bob," and so on.
   - **First Name** - Enter the first name of the user.
   - **Last Name** - Enter the last name of the user.
   - **Email ID -** Enter the email ID associated with the user.
   - **Password** - Enter the password for the user.
   - **Role -** Selecting Administrator assigns the user the administrator role, which gives the user permission to act as an administrator in the HPE Ezmeral Unified Analytics Software UI. If you do not select Administrator, the user is assigned the member role.

Removing Users

To remove a user:

1. In the list of users, select the user you want to remove.
2. Click into the **Actions** column, and click the **Delete** option. Alternatively, click the **Delete** button on the screen. The system prompts you to confirm the action. Once you confirm, the user is removed.

Editing Roles and Password

To edit the role and password for a user:

1. In the list of users, select the user you want to edit.
2. Click into the **Actions** column, and click the **Edit** option.

3. In the drawer that opens, change the password and role for the user.

4. Click **Update**.

Adding and Removing Users Programmatically

Describes how to add and remove users through the Kubernetes API using the EzUserQuery and EzUserConfig custom resources.

The user management operator in HPE Ezmeral Unified Analytics Software responds to the EzUserQuery and EzUserConfig custom resources when they are created by a client with the required Kubernetes API permissions.

Use the administrative `kubectl config` that you get when you create the HPE Ezmeral Unified Analytics Software cluster to onboard and manage users programmatically through the Kubernetes API.

To onboard a user, complete the following steps:

1. Use the EzUserQuery custom resource to search for the user in the internal or external AD/LDAP directory. The EzUserQuery returns a list of attributes for a user, including the Keycloak ID. The Keycloak ID is required to onboard a user.

2. Use the EzUserConfig custom resource to onboard the user.

The following sections describe the custom resources:

EzUserQuery

Use the EzUserQuery custom resource to query the user AD/LDAP directory.

The EzUserQuery properties map directly to the query types of the Keycloak user API. Providing values for the *email*, *firstName*, *lastName*, and/or *username* properties sets criteria that must match the returned users. The *search* property value is typically the most useful and can match against the *email* or *username*.

Keycloak returns the query response and the status updates. The query results are bounded and a query only returns up to five results. Narrow your search criteria to reduce the number of results returned. Results show you attributes for the user, for example if they are enabled (true/false), id (keycloak user ID), and role (admin or not).

EzUserQuery resources self-delete after they expire.

Using the EzUserQuery Custom Resource

1. In a YAML file, add the following properties, specifying your own values:

```
apiVersion: ezconfig.hpe.ezaf.com/
v1alpha1
kind: EzUserQuery
metadata:
  name: my-query-1
spec:
  search: joel
```

2. To create and get the query, run the following commands, specifying your YAML file name:

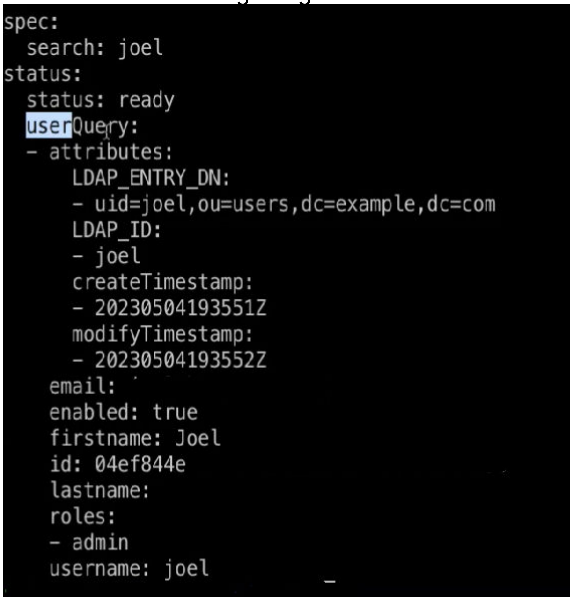
   ```
   kubectl create -f query.yaml

   kubectl get ezuserquery
   ```

   A ready status indicates that there are query results. This is the `status.status` property.

3. To query the AD/LDAP directory, run the following command, specifying your query name:

   ```
   kubectl get ezuserquery
   my-query-1 -o yaml
   ```

   The command returns results similar to those shown in the following image:



```
spec:
  search: joel
status:
  status: ready
  userQuery:
  - attributes:
      LDAP_ENTRY_DN:
      - uid=joel,ou=users,dc=example,dc=com
      LDAP_ID:
      - joel
      createTimestamp:
      - 20230504193551Z
      modifyTimestamp:
      - 20230504193552Z
    email: [redacted]
    enabled: true
    firstname: Joel
    id: 04ef844e
    lastname:
    roles:
    - admin
    username: joel
```

   The `userQuery` property displays the user attributes.

EzUserConfig

Use the EzUserConfig custom resource to enable/disable users and manage user roles.

EzUserConfig identifies the user (via keycloak ID) and indicates the roles that a user should have when onboarded.

The following table describes the differences between internal and external AD/LDAP servers when using EzUserConfig:

AD/LDAP Server Type	Description
Internal	EzUserConfig *creates* and *enables* a user. Deleting an EzUserConfig disables and deletes the user.
External	EzUserConfig *enables* a user. EzUserConfig identifies the user (via Keycloak ID) and sets the user role. Deleting an EzUserConfig disables the user.

The EzUserConfig status stanza shows user attributes, whether the user is successfully enabled, the roles that have successfully been assigned, and any error messages.

To retrieve and display information about the `ezuserconfig` resource, run:

```
kubectl get ezuserconfig
```

You can view the NAMESPACE column when running the `kubectl get ezuserconfig` command. Note that the namespace name does not match the user's username. The namespace name is derived by combining the prefix from the username and adding the suffix. For example, if the AD username is `Arun1.Gowda` and Unified Analytics username is `arun1.gowda`, the Unified Analytics namespace name will be `arun1-gowda-2ccc059b`.

TIP:

- The user management operator actually onboards the user.
- *Enabled* is not a role that you can assign to a user.
- The following resources include the label `hpe-ezua/username=<ua-username>`:
  - ezuserconfig
  - user namespace
  - profile
- To find the namespace name for your username, run:

  ```
  kubectl get ns -l hpe-ezua/username=<ua-username>
  ```

Using the EzUserConfig Custom Resource

1. In a YAML file, add the following properties, specifying your own values:

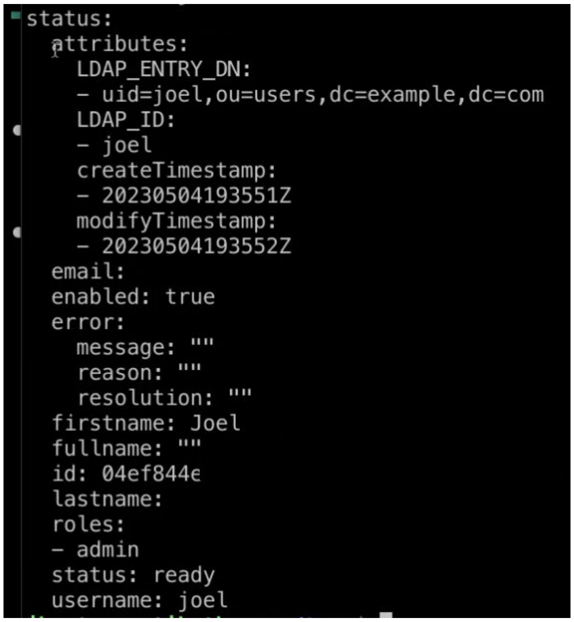
   ```
   apiVersion: ezconfig.hpe.ezaf.com/
   v1alpha1
   kind: EzUserConfig
   metadata:
     name: my-admin-user-1
   spec:
     id: 04ef844e
     roles:
       - admin
   ```

   Note that the ID is the Keycloak ID that you can get using the EzUserQuery custom resource.

2. To see a list of all the attributes for a user, run the following command, specifying the name of the user you want to see attributes for:

   ```
   get ezuserconfig joel -o yaml
   ```

   The command returns results similar to those shown in the following image:



```
status:
  attributes:
    LDAP_ENTRY_DN:
    - uid=joel,ou=users,dc=example,dc=com
    LDAP_ID:
    - joel
    createTimestamp:
    - 20230504193551Z
    modifyTimestamp:
    - 20230504193552Z
  email:
  enabled: true
  error:
    message: ""
    reason: ""
    resolution: ""
  firstname: Joel
  fullname: ""
  id: 04ef844e
  lastname:
  roles:
  - admin
  status: ready
  username: joel
```

Auth Tokens

Describes auth tokens and how auth tokens work in HPE Ezmeral Unified Analytics Software.

An auth token encapsulates a user's authentication and authorized roles within the Unified Analytics cluster. Unified Analytics uses auth tokens to determine a user's identity and authorization. Some applications also use the auth tokens to communicate with other services in the Unified Analytics cluster.

All auth tokens in Unified Analytics are short-lived for security and functionality purposes. Short-lived tokens:

- Work well for long-running jobs
- Promptly reflect changes in user enablement and roles
- Have a short lifespan, which limits the time a malicious person or program can use them if accessed

In Unified Analytics, auth tokens are in the:

- Request header (of incoming web requests)
- Token secret (Unified Analytics-specific access token in a user's namespace)

Request Header Token

Applications can look at the request header and use the token in request processing. When a user signs in to HPE Ezmeral Unified Analytics Software, the token in the request header is valid for one hour. This one hour lifetime limits the amount of time that an external client can store and reuse the access token.

When the token expires, it becomes invalid and the external client program that was using the token must obtain a new access token. HPE does not recommend storing this token in applications or reusing the token.

Token Secret

When a Unified Analytics administrator creates a new user or adds a user to the Unified Analytics platform, the system automatically creates a unique namespace for that user. All jobs that the user launches, such as a Spark job, run in the user's unique namespace, isolated from other users. Included in the user's namespace is a secret, created by the Unified Analytics token service. The secret in the user's namespace contains an access token, aptly named `access-token`, in the value of AUTH_TOKEN.

The access token in the secret has the following attributes:

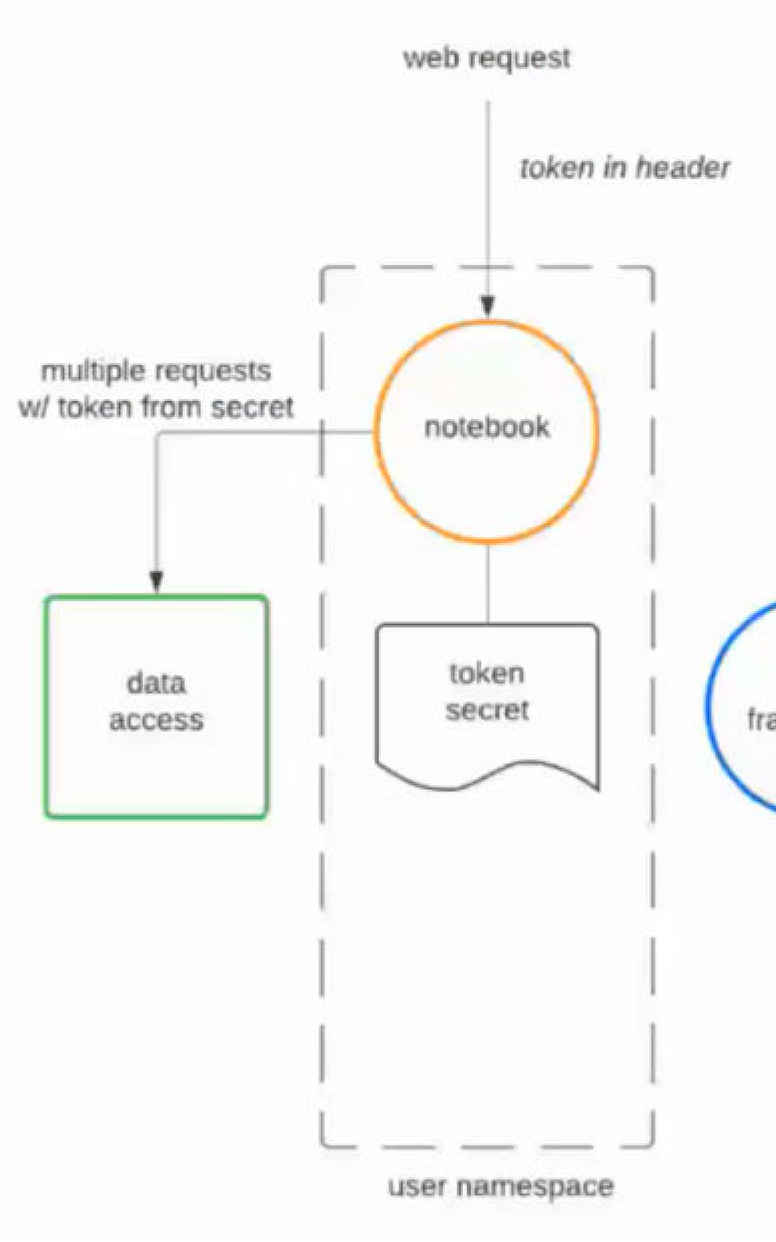
- Valid for up to 30 minutes, but always has a minimum of 10 minutes left to live. When the token is read from the secret, the token always has 10 minutes left to live.
- The token is started when a user signs in to Unified Analytics.
- The Unified Analytics token service regularly refreshes the token unless the user login is blocked for some reason.
- Token claims:
  - Contains the `namespace` claim. You can locate the user's namespace using the `namespace` claim.
  - When the sign in via the email address is not allowed, the `email` claim remains empty incase the email address is not provided.

Notebook Access to the Token Secret

A web request to the user's notebook contains a token in the request header; however, if the notebook needs to make requests to another service that requires a token for authentication, the notebook can get the access token out of the secret.

The secret mounts the notebook's filesystem in the local pod at `/etc/secrets/ezua/.auth_token`. The notebook reads the token from the file and puts the token in an outgoing request. The notebook can do this repeatedly because the token service in Unified Analytics regularly refreshes the access token.

The following diagram shows the basic access flow between the notebook and the secret:



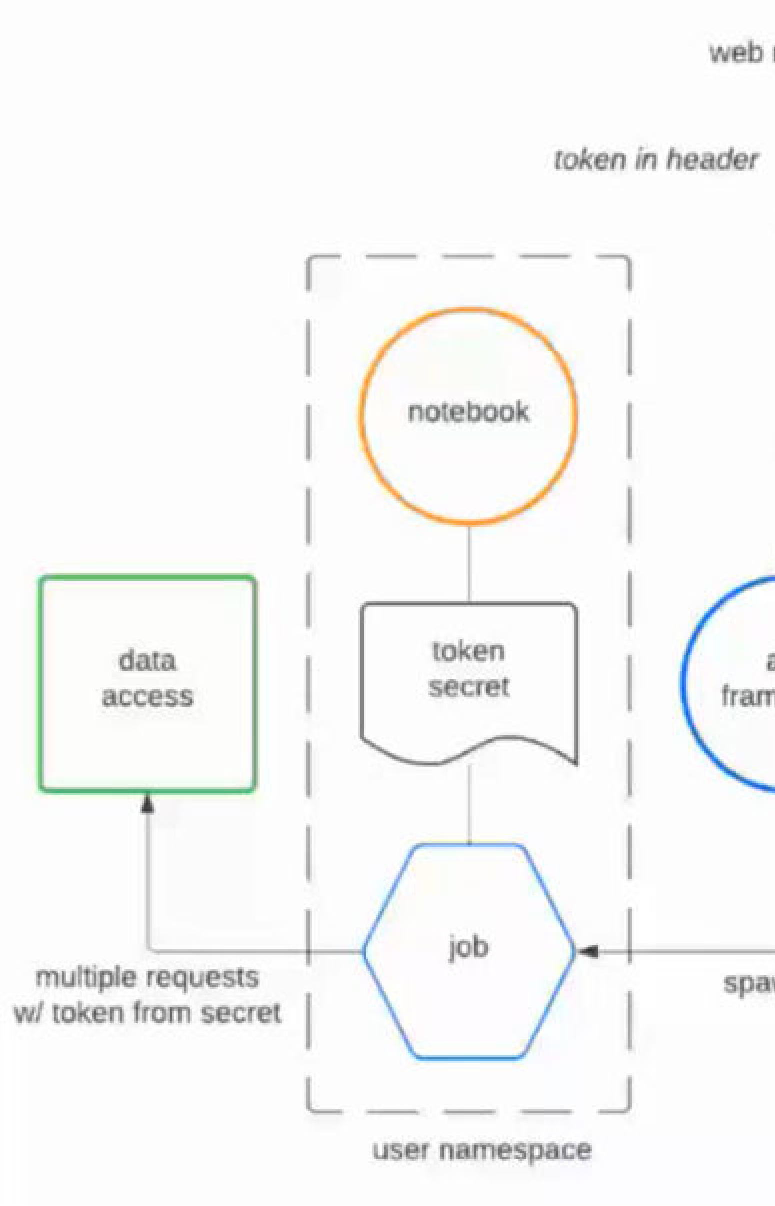
You can mount the access token to any pod running jobs in your namespace. To mount the access token to a pod, add the `hpe-ezua/add-auth-token` annotation to the Pod configuration, as shown in the following example:

```
apiVersion: v1
kind: Pod
metadata:
   name: nginx
   namespace: hpedemo-usr
   annotations:
     hpe-ezua/add-auth-token: "true"
spec:
  containers:
  - name: nginx
    image: nginx: 1.14.2
    ports:
    - containerPort: 80
```

Application Access to the Token Secret

A web request to an application in Unified Analytics contains a token in the request header. When a user runs an application in Unified Analytics, the application runs in the user's namespace. If the application needs to access data or another service, the application can get the access token from the secret in the user's namespace and then use the token to make multiple requests for data access.

The following diagram shows the basic access flow between an application, secret, and data source:



Auth Tokens for External Client Application Access to External APIs

Some application APIs are exposed for use outside of the Unified Analytics cluster. These APIs are also authenticated by tokens. External client applications can obtain a token to talk to these external APIs.

If the external client application understands OIDC OAuth2 protocols, you can point the application to a particular OIDC client that Keycloak exposes. However, if you need to get a token and put the token in a particular header request that you are sending to an API, you can do this through the API endpoint in Unified Analytics that is exposed for Keycloak, by providing the user credentials (username and password) in a cURL POST request to obtain a token and then use the token to authenticate, as described in Obtaining Access Tokens with User Credentials on page 243.

The token is a short-lived token in JWT format. If the external client application will be running multiple API requests where authentication is required, the external client should get a refresh token from the API and then repeatedly request new access tokens. See Obtaining Refresh Tokens on page 244.

Auth Tokens for External Clients that Require Kubernetes API Credentials

If an external client can read secrets, the external client can obtain the access token from the secret in a user's namespace; however, this is not recommended as the default method. See Obtaining Access Tokens with a Kubernetes Secret on page 246.

Related concepts

Obtaining Access Tokens with User Credentials on page 243
Describes how to obtain access and refresh tokens with a user's credentials and the Keycloak service address.

Obtaining Refresh Tokens on page 244
Describes how to refresh access tokens.

Obtaining Access Tokens with a Kubernetes Secret on page 246
Describes how to obtain access tokens with a Kubernetes secret.

Changing External Auth Token Settings in Keycloak on page 247
Describes how to change access and refresh token settings through the Keycloak Admin Console.

More information

Identity and Access Management on page 220
Describes identity and access management in HPE Ezmeral Unified Analytics Software.

Obtaining External Auth Tokens for REST API Endpoint Access

Describes an example process for an external client application to obtain access tokens for authenitcated access to a REST API endpoint and provides links to topics with instructions for obtaining auth tokens.

The following steps describe an example process for an external client application that needs to repeatedly authenticate to a REST API endpoint over a period of time:

1. The extermal client application presents a user's credentials to obtain a refresh token. The client application stores the refresh token.
2. Before any authenticated API access, the client application uses the stored refresh token to get an access token and a new refresh token.
3. The freshly-obtained access token is used to access the REST API endpoint, and the new refresh token is stored (replacing the old one).
4. Repeat steps 2 and 3 for the duration of the client applications’s activity.

The following sections provide information and instructions related to external client programs obtaining access tokens to access REST API endpoints exposed by the applications in an HPE Ezmeral Unified Analytics Software cluster.

Related concepts

Changing External Auth Token Settings in Keycloak on page 247
Describes how to change access and refresh token settings through the Keycloak Admin Console.

Obtaining Access Tokens with User Credentials

Describes how to obtain access and refresh tokens with a user's credentials and the Keycloak service address.

An external client program can obtain tokens for a user through a cURL POST request to a token-granting URL path under the Keycloak service address. Keycloak has an endpoint on the ua-grant OIDC client in the HPE Ezmeral Unified Analytics Software realm for the resource owner's password credentials. The OIDC client is the API endpoint that the external client program interacts with for token operations. For additional information, see Identity and Access Management on page 220.

Use the Keycloak service address (`keycloak.<cluster-DNS-domain-name>.com`) and user credentials (username and password) in a cURL POST request to obtain an access token (in JWT format) from the response body. You can then use the access token in requests to application API endpoints by specifying the token as a bearer token in the `Authorization` header.

cURL POST Request

Use the following cURL POST request to obtain access and refresh tokens:

```
KC_ADDR=keycloak.<cluster-DNS-domain-name>.com
USERNAME=<username>
PASSWORD=<user-password>

response_json=$(curl --data
"username=$USERNAME&password=$PASSWORD&grant_type=password&client_id=ua-gran
t" "https://$KC_ADDR/realms/UA/protocol/openid-connect/token")
```

> **TIP:** For testing purposes, you can use `curl -k` to skip peer certificate validation if the local CA certificate store cannot validate the Unified Analytics gateway certificate.

Offline Access

If you do not want the token to expire, include the `offline_access`scope in the request, as shown:

```
response_json=$(curl --data
"username=$USERNAME&password=$PASSWORD
&grant_type=password&client_id=ua-gran
t&scope=offline_access" "https://
$KC_ADDR/realms/UA/protocol/
openid-connect/token")
```

An `offline_access` token can be used repeatedly; however, if an `offline_access` refresh token is not used for thirty days, the token becomes invalid.

Reconfigured ua-grant OIDC Client as a Confidential Client

If the ua-grant OIDC client is reconfigured to be a confidential client, you must specify the `client_secret` as one of the data parameters in the cURL request. For example, if ua-grant is a confidential client with the a secret value of 3EMVFnKnOU3B5Yh9B8MchwcFHvOVTcdh, then the cURL request must include that value for the `client_secret` parameter, as shown:

```
response_json=$(curl --data
"username=$USERNAME&password=$PASSWORD
&grant_type=password&client_id=ua-gran
t&client_secret=3EMVFnKnOU3B5Yh9B8Mchw
cFHvOVTcdh" "https://$KC_ADDR/
realms/UA/protocol/openid-connect/
token")
```

For additional information, see Making the ua-grant OIDC Client a Confidential Client on page 248.

Getting the Access and Refresh Tokens from the Response Body

To get the access and refresh tokens, extract the `access_token` and `refresh_token` attributes from the JSON object in the response body. For example, you can use the jq command-line JSON processor, as shown:

```
ACCESS_TOKEN=$(echo "$response_json" | jq -r '.access_token')
REFRESH_TOKEN=$(echo "$response_json" | jq -r '.refresh_token')
```

The tokens are in JWT format.

Example

The DNS domain name for a Unified Analytics cluster is `my-ua.com`, which makes the Keycloak address `keycloak.my-ua.com`. An external client program can obtain tokens for a user (bob) through a cURL POST request to the token-granting URL path under the `keycloak.my-ua.com` service address, as shown:

```
KC_ADDR=keycloak.my-ua.com
USERNAME=bob
PASSWORD=bobspassword
response_json=$(curl --data
"username=$USERNAME&password=$PASSWORD&grant_type=password&client_id=ua-gran
t" "https://$KC_ADDR/realms/UA/protocol/openid-connect/token")
```

From the response body, extract the `access_token` and `refresh_token` attributes from the JSON object, as shown:

```
ACCESS_TOKEN=$(echo "$response_json" | jq -r '.access_token')
REFRESH_TOKEN=$(echo "$response_json" | jq -r '.refresh_token')
```

To use the access token in requests to the application API endpoints, specify the token as a bearer token in the `Authorization` header.

Related concepts

Changing External Auth Token Settings in Keycloak on page 247
Describes how to change access and refresh token settings through the Keycloak Admin Console.

Obtaining Refresh Tokens

Describes how to refresh access tokens.

You can use a cURL POST request to refresh an access token. When you refresh an access token, the new token reflects the user's current roles and attributes. Each refresh token is typically valid for a week. Including the `offline_access` scope in the cURL POST obtains a refresh token that does not expire unless the token is not used for thirty days.

cURL POST Request

Use the following cURL POST request to obtain new access and refresh tokens:

```
response_json=$(curl --data
"grant_type=refresh_token&client_id=ua-grant&refresh_token=$REFRESH_TOKEN"
"https://$KC_ADDR/realms/UA/protocol/openid-connect/token")
```

> **TIP:** For testing purposes, you can use `curl -k` to skip peer certificate validation if the local CA certificate store cannot validate the Unified Analytics gateway certificate.

Offline Access

If you do not want the token to expire, include the `offline_access`scope in the request, as shown:

```
response_json=$(curl --data
"grant_type=refresh_token&client_id=u
a-grant&refresh_token=$REFRESH_TOKEN&s
cope=offline_access" "https://
$KC_ADDR/realms/UA/protocol/
openid-connect/token")
```

An `offline_access` token can be used repeatedly; however, if an `offline_access` refresh token is not used for thirty days, the token becomes invalid.

Reconfigured ua-grant OIDC Client as a Confidential Client

If the ua-grant OIDC client is reconfigured to be a confidential client, you must specify the `client_secret` as one of the data parameters in the cURL request. For example, if ua-grant is a confidential client with the a secret value of 3EMVFnKnOU3B5Yh9B8MchwcFHvOVTcdh, then the cURL request must include that value for the `client_secret` parameter, as shown:

```
response_json=$(curl --data
"grant_type=refresh_token&client_id=u
a-grant&refresh_token=$REFRESH_TOKEN&c
lient_secret=3EMVFnKnOU3B5Yh9B8MchwcFH
vOVTcdh" "https://$KC_ADDR/realms/UA/
protocol/openid-connect/token")
```

For additional information, see Making the ua-grant OIDC Client a Confidential Client on page 248.

Getting the Access and Refresh Tokens from the Response Body

To get the access and refresh tokens, extract the `access_token` and `refresh_token` attributes from the JSON object in the response body. For example, you can use the jq command-line JSON processor, as shown:

```
ACCESS_TOKEN=$(echo "$response_json" | jq -r '.access_token')
REFRESH_TOKEN=$(echo "$response_json" | jq -r '.refresh_token')
```

The tokens are in JWT format.

To use the access token in requests to the application API endpoints, specify the token as a bearer token in the `Authorization` header.

Refreshing Tokens in Notebook

If you encounter a JWT token expiration error while running cells in the notebbok, you can resolve it by running the `%update_token` magic function. To learn more, see %update_token on page 451

Related concepts

Changing External Auth Token Settings in Keycloak on page 247
Describes how to change access and refresh token settings through the Keycloak Admin Console.

Obtaining Access Tokens with a Kubernetes Secret

Describes how to obtain access tokens with a Kubernetes secret.

Each user that signs in to the HPE Ezmeral Unified Analytics Software UI is assigned a user-specific namespace. The user-specific namespace contains a Kubernetes secret with an access token, aptly named `access-token`. The access token is created specifically for the user in the value of AUTH_TOKEN. A token read from this resource (AUTH_TOKEN) has between 30 and 10 minutes to live.

NOTE: Keycloak settings do not affect the lifetime of this access token.

Obtaining an Access Token for a User

Any external client program with the appropriate Kubernetes API credentials can obtain a valid access token for a user.

Run the following `kubectl` command to obtain the access token for a specified user:

```
kubectl -n $USER_NAMESPACE get secret access-token -o
jsonpath='{.data.AUTH_TOKEN}' | base64 -d
```

IMPORTANT: To run this command, `kubectl` must be set up with either admin access or a configuration that has credentials that allow access to a user's secret.

Storing Kubernetes Credentials Externally

Before you store any Kubernetes API credentials outside of the HPE Ezmeral Unified Analytics Software cluster, consider the security implications. As with any externally stored credentials, the external client is responsible for securing the credentials.

However, there may be cases where external storage is appropriate. For example, if a client already requires Kubernetes API credentials for other reasons, then the client can use this method to get valid user access tokens without having to use and secure refresh tokens.

Usage Notes

The following list describes scenarios where a user's access token becomes invalid or does not exist:

- If a user exists in HPE Ezmeral Unified Analytics Software but has not signed in to the HPE Ezmeral Unified Analytics Software UI, the secret does not contain an access token for the user. The secret only contains an access token after the user signs in.

- If the user is removed (offboarded) from the HPE Ezmeral Unified Analytics Software cluster, the user's namespace and secret are also removed and no longer exist.

- An expired HPE Ezmeral Unified Analytics Software license disables the cluster. When a cluster is disabled, the token expires and becomes invalid until the cluster is enabled and the user successfully signs in to HPE Ezmeral Unified Analytics Software through the UI.

- If a user is disabled in the AD/LDAP server, the token expires and becomes invalid until the user is enabled in the AD/LDAP server and signs in to HPE Ezmeral Unified Analytics Software through the UI.

Changing External Auth Token Settings in Keycloak

Describes how to change access and refresh token settings through the Keycloak Admin Console.

For access and refresh tokens obtained through the user credential or refresh methods, the site administrator can set token lifetimes to any value appropriate for the external client applications at the site. The site administrator can also change the ua-grant OIDC client to a confidential client configuration.

A site administrator can sign in to the Keycloak Admin Console to make the following changes:

- Change the Externally Granted Access Token Lifetime
- Change the 7-Day Refresh Token Lifetime
- Change the 30-Day Idle Timer for Offline Refresh Tokens
- Make the ua-grant OIDC Client a Confidential Client

NOTE: You can look at the value of the `exp` claim in the token itself to determine the access token and refresh token expiration. The `exp` claim is the UNIX-epoch representation of the token's expiration date and time. If a client needs to make token-handling decisions based on times, using the `exp` value is best.

When changing the token lifetime, consider both security and usability. For example, long-lived tokens are reusable, which can be convenient but can also cause security issues if the token is accessed by an unauthorized user or application. Any person or entity holding a user's access token can act as that user to access the application endpoints until the token expires.

NOTICE: HPE recommends a one hour lifetime; however, the site administrator must make the token lifetime determination for their environment and adjust for the security tradeoffs.

To sign in and make changes, the site administrator must have the Keycloak admin password. Accessing the Keycloak Admin Console on page 247 provides the steps to get the Keycloak admin password.

Accessing the Keycloak Admin Console

To access the Keycloak Admin Console, complete the following steps:

1. To get the Keycloak admin password, use `kubectl` and the administrative (full access) `kubeconf`, as shown:

```
kubectl -n ezaddon-system get secret hpecp-bootstrap-authcreds -o
jsonpath='{.data.SUPER_ADMIN_PASSWORD}' | base64 -d
```

2. In a web browser, go to `https://$KC_ADDR/admin/master/console/` and use the admin password to log in as the admin user.

   TIP: KC_ADDR is the Keycloak service address:

   `keycloak.<cluster-DNS-domain-name>.com`

Changing the Lifetime of the Externally Granted Access Token

To change the lifetime of the externally granted access token for an HPE Ezmeral Unified Analytics Software site, complete the following steps:

1. Sign in to the Keycloak Admin Console, as described in Accessing the Keycloak Admin Console on page 247.

2. In the upper left pulldown, switch to the **UA realm**.

3. In the left navigation bar, select **Clients**.
4. Select the **ua-grant** client.
5. Select the **Advanced** tab.
6. On the right, click **Advanced Settings**.
7. Change the **Access Token Lifespan** value.
8. Click **Save** at the bottom of the **Advanced Settings** box.

Changing the 7-Day Refresh Token Lifetime

To change the lifetime of the 7-day refresh token, complete the following steps:

1. Sign in to the Keycloak Admin Console, as described in Accessing the Keycloak Admin Console on page 247.
2. In the upper left pulldown, switch to the **UA realm**.
3. In the left navigation bar, select **Realm Settings**.
4. Select the **Sessions** tab.
5. Change the values of **SSO Session Idle** and/or **SSO Session Max**.

   NOTE: Changing these values affects the behavior of the refresh tokens and the Keycloak session cookies. The session cookies set the upper limit for how long a user can stay logged in through the web browser.
6. Click **Save** at the bottom of the page.

Changing the 30-Day Idle Timer for Offline Refresh Tokens

To change the 30-day idle timer for offline refresh tokens, complete the following steps:

1. Sign in to the Keycloak Admin Console, as described in Accessing the Keycloak Admin Console on page 247.
2. In the upper left pulldown, switch to the **UA realm**.
3. In the left navigation bar, select **Realm Settings**.
4. Change the value of **Offline Session Idle**.
5. Click **Save** at the bottom of the page.

Making the ua-grant OIDC Client a Confidential Client

A confidential client configuration ensures that a refresh token is used with a secret for additional security. For example, if a refresh token is leaked, the refresh token is useless unless the token is accompanied by the secret.

For added security, the external client program should use different storage methods for the secret and refresh token.

The same secret value should be provided to all client programs that use the ua-grant OIDC client in an HPE Ezmeral Unified Analytics Software cluster.

To make the ua-grant OIDC client a confidential client, complete the following steps:

1. Sign in to the Keycloak Admin Console, as described in Accessing the Keycloak Admin Console on page 247.

2. In the upper left pulldown, switch to the **UA realm**.

3. In the left navigation bar, select **Clients**.

4. Select the **ua-grant client**.

5. Scroll down to **Client authentication** and toggle on.

6. Click **Save** at the bottom of the page.

7. Select the **Credentials** tab that appears.

8. In the **Client secret** box, click on the eye icon to reveal the client secret value, and/or click the copy icon to copy the value to the clipboard.

9. Use this `client_secret` value in POST requests to this OIDC client.

Managing Data Access

Describes data access management and how to grant members access to data.

HPE Ezmeral Unified Analytics Software administrators have unrestricted access to all data sources and underlying schemas, tables, views, and buckets. Admins can grant *public access* to a data source or they can grant members access to specific schemas, tables, views, or buckets in a data source.

Public access grants all members *read* and *write* access to all data in a data source. Alternatively, admins can grant members *read*, *write*, or *read & write* access to specific schemas, tables, views, or buckets in a data source.

TIP:

- Members should contact their HPE Ezmeral Unified Analytics Software admin to request access.

- Any access granted can also be revoked by an HPE Ezmeral Unified Analytics Software admin.

- If an admin deletes a member in HPE Ezmeral Unified Analytics Software, the member's access to data is also deleted.

- The system transparently enforces data access policies across all applications and clients.

The following sections provide the steps for granting and revoking data access.

Granting Public Access to a Data Source

HPE Ezmeral Unified Analytics Software administrators can make a data source publicly accessible. When an admin makes a data source publicly accessible, all members have full access (read and write) permissions on the data source and the data within it.

To make a data source publicly accessible, complete the following steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.

2. In the left navigation bar, select **Data Engineering > Data Sources**.

3. Select the **Structured Data** or **Object Store Data** tab.

4. In the data source tile, click the three-dots.

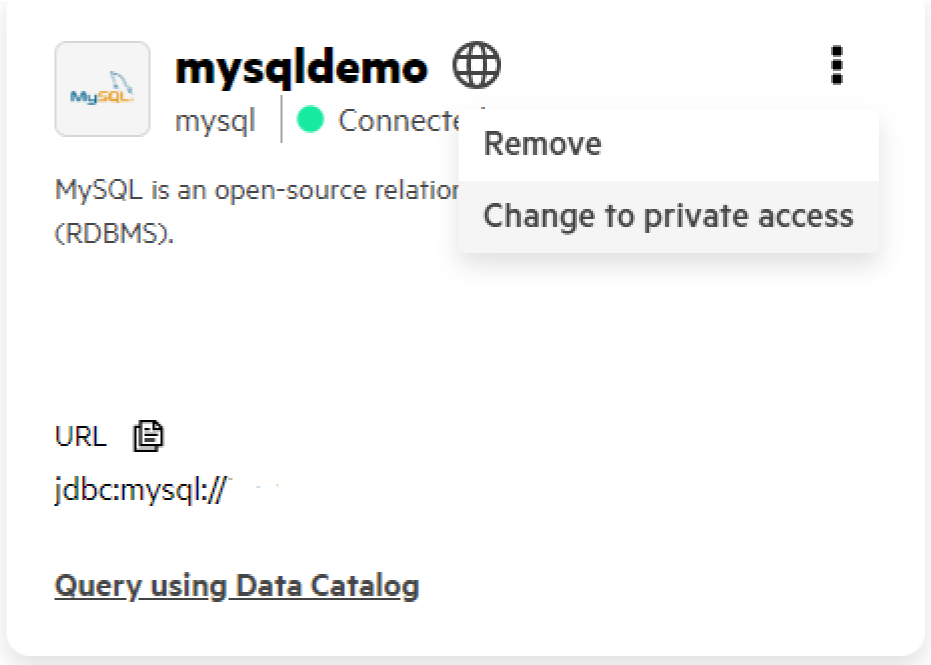
5. Select **Change to public access**.
6. In the **Data Access** dialog, click **Proceed** or **Cancel**. If you choose to proceed, the system displays the message:

   ```
   Access changed for the data source: <data-source-name>
   ```

Revoking Public Access to a Data Source

HPE Ezmeral Unified Analytics Software administrators can revoke public access to a data source. Revoking public access to a data source makes the data in the data source totally inaccessible to all members. Only admins can access the data in the data source.

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Data Engineering > Data Sources**.
3. Select the **Structured Data** or **Object Store Data** tab.
4. In the data source tile, click the three-dots.



5. Select **Change to private access**.
6. In the **Data Access** dialog, click **Proceed** or **Cancel**. If you choose to proceed, the system displays the message:

   ```
   Access changed for the data source: <data-source-name>
   ```

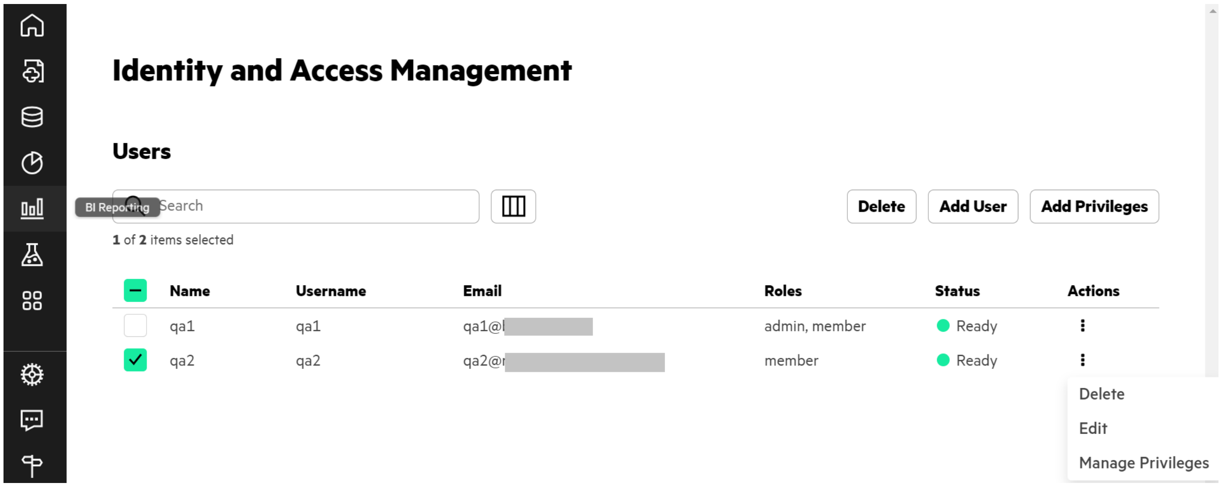
Granting a Member Access to Data

HPE Ezmeral Unified Analytics Software administrators can grant a member access to one or more tables, views, or buckets in a schema.

To grant a member access to data, complete the following steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Administration > Identity & Access Management**.

3. On the **Identity and Access Management** screen, locate the user.
4. In the **Actions** column of the user row, click the three-dots and select **Manage Privileges**.



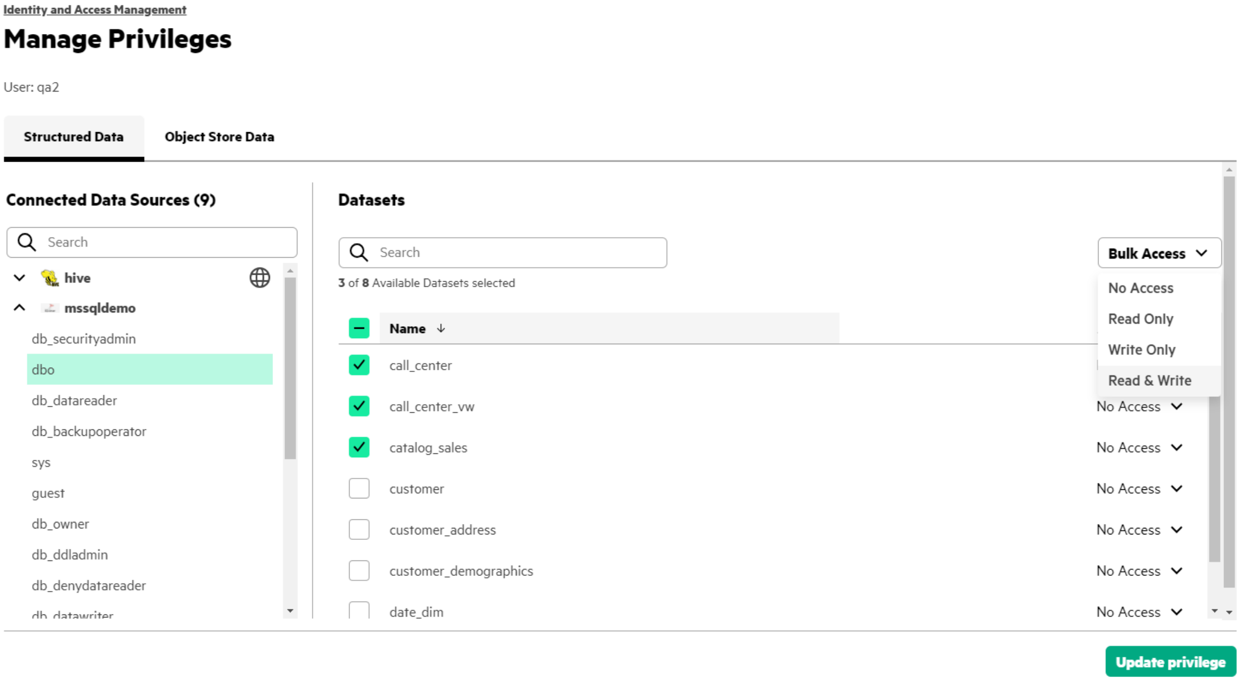
Identity and Access Management

Users

1 of 2 items selected

	Name	Username	Email	Roles	Status	Actions
□	qa1	qa1	qa1@l[redacted]	admin, member	● Ready	⋮
☑	qa2	qa2	qa2@r[redacted]	member	● Ready	⋮

5. On the **Manage Privileges** screen, select the **Structured Data** or **Object Store Data** tab, depending on the type of data that you want to grant the user access to.
6. Expand a data source and select a schema.



Manage Privileges

User: qa2

3 of 8 Available Datasets selected

	Name ↓	
☑	call_center	
☑	call_center_vw	No Access
☑	catalog_sales	No Access
□	customer	No Access
□	customer_address	No Access
□	customer_demographics	No Access
□	date_dim	No Access

7. In the **Datasets** area, select the tables, views, or buckets that you want to grant the user access to. You can grant **Read**, **Write** or **Read & Write** access.
   - To grant a user access to a single table, view, or bucket, use the **Access Type** column dropdown in the row of the table, view, or bucket.
   - To grant a user access to multiple tables, views, or buckets, use the **Bulk Access** dropdown and select the access that you want to grant the user on the selected tables, views, or buckets.

8. Click **Update Privilege**. The system displays the message:

   ```
   Updated privileges for the user: <user-name>
   ```

Granting Group Access to Data

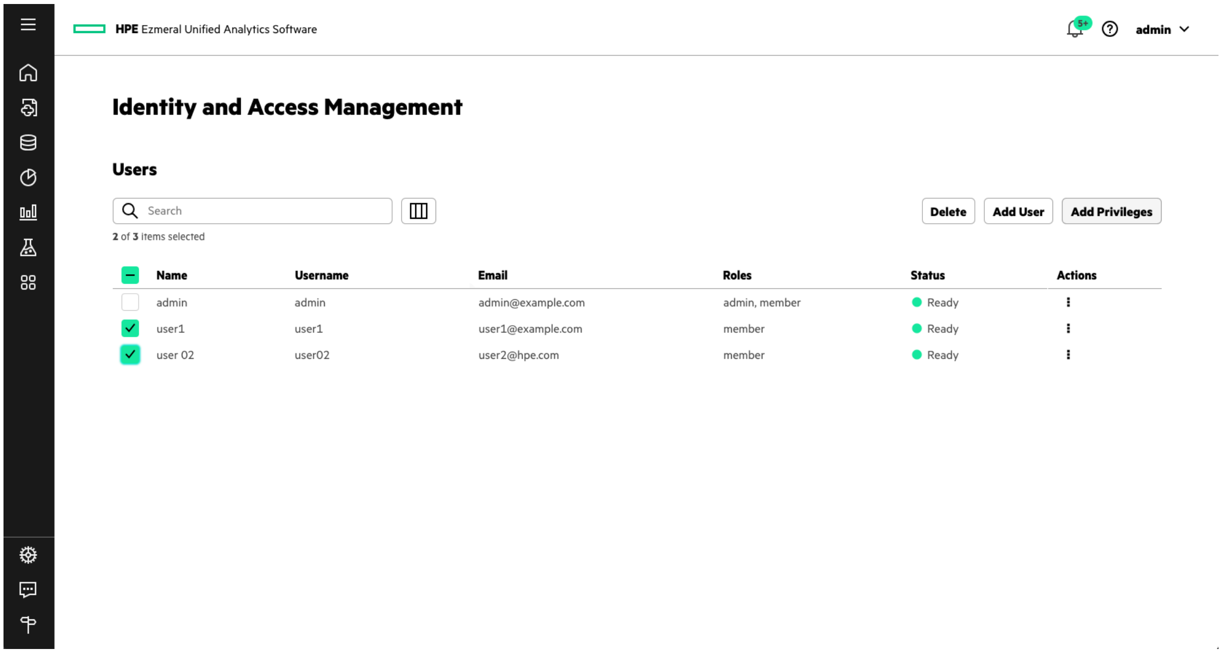
HPE Ezmeral Unified Analytics Software administrators can simultaneously grant a group of users access to one or more tables, views, or buckets in a schema.

To grant group access to data, complete the following steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Administration > Identity & Access Management**.
3. On the **Identity and Access Management** screen, choose one or more member users by selecting their checkboxes.

   NOTE: You cannot select the current session user. For example, if you are signed in as **admin**, you cannot select **admin**.

4. Click **Add Privileges**.



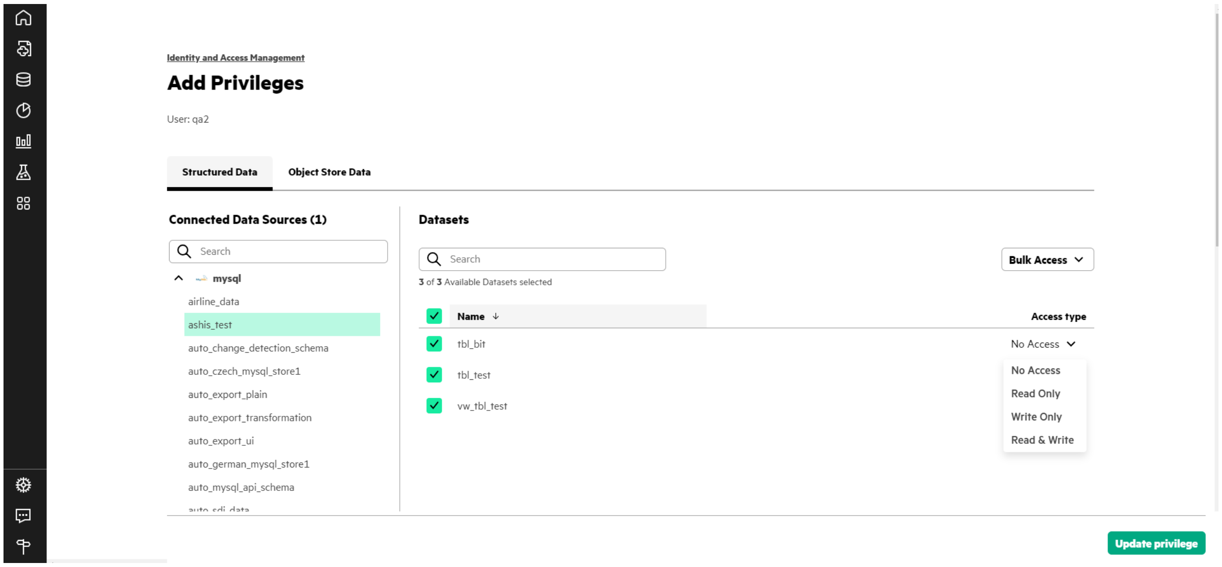
Identity and Access Management

Users

2 of 3 items selected

	Name	Username	Email	Roles	Status	Actions
□	admin	admin	admin@example.com	admin, member	● Ready	⋮
☑	user1	user1	user1@example.com	member	● Ready	⋮
☑	user 02	user02	user2@hpe.com	member	● Ready	⋮

5. On the **Add Privileges** screen, select the **Structured Data** or **Object Store Data** tab, depending on the type of data that you want to grant users access to.
6. Expand a data source and select a schema.


Identity and Access Management

Add Privileges

User: qa2

Structured Data | Object Store Data

Connected Data Sources (1)

Search

mysql
- airline_data
- ashis_test
- auto_change_detection_schema
- auto_czech_mysql_store1
- auto_export_plain
- auto_export_transformation
- auto_export_ui
- auto_german_mysql_store1
- auto_mysql_api_schema
- auto_sdi_data

Datasets

Search

Bulk Access

3 of 3 Available Datasets selected

	Name ↓	Access type
☑	tbl_bit	No Access
☑	tbl_test	
☑	vw_tbl_test	

No Access
Read Only
Write Only
Read & Write

Update privilege

7. In the **Datasets** area, select the tables, views, or buckets that you want to grant the user access to. You can grant **Read**, **Write** or **Read & Write** access.

   NOTE:

   - New access privileges are added to the privileges a user already has; they do not replace the previous access privileges. For example, if **user1** previously had **Read** access and you grant **user1** **Write** access, **user1** now has **Read & Write** access to the data.
   - To grant users access to a single table, view, or bucket, use the **Access Type** column dropdown in the row of the table, view, or bucket.
   - To grant users access to multiple tables, views, or buckets, use the **Bulk Access** dropdown and select the access that you want to grant the user on the selected tables, views, or buckets.

8. Click **Update privilege**.

Revoking Member Access to Data

HPE Ezmeral Unified Analytics Software administrators can revoke a member's access to schemas, tables, views, and buckets. Revoking access makes the data inaccessible to the member.

Administrators can use the **Manage Privileges** screen to revoke access to one or more data sources and their schemas, tables, views, and buckets. To revoke all access to a specific data source for members, use the **Remove Privileges** option.

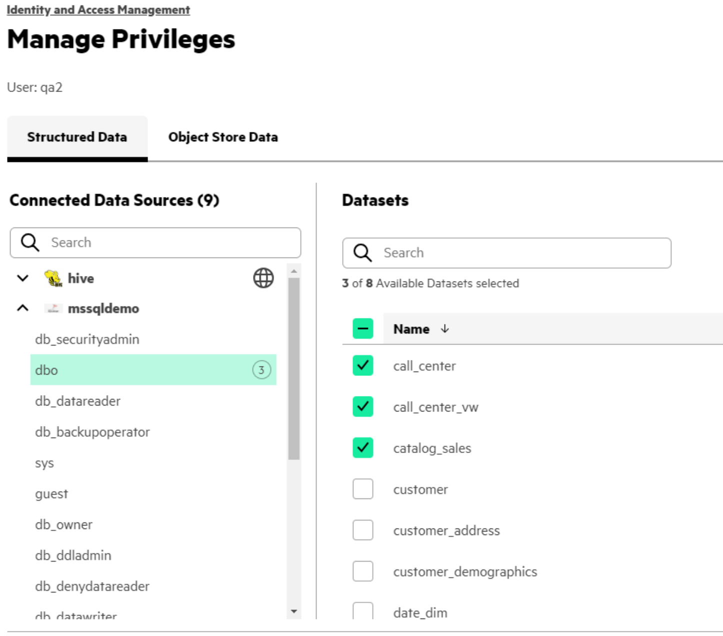
NOTE: You can use the Remove Privileges option only for private data sources.

Manage Privileges

To revoke member access to data, complete the following steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Administration > Identity & Access Management**.

3. On the **Identity and Access Management** screen, locate the user.
4. In the **Actions** column of the user row, click the three-dots and select **Manage Privileges**.
5. On the **Manage Privileges** screen, select the **Structured Data** or **Object Store Data** tab, depending on the type of data that you want to revoke access to.
6. Expand the data source and select the schema that contains the data you want to revoke access to.



Identity and Access Management

Manage Privileges

User: qa2

Structured Data | Object Store Data

Connected Data Sources (9)

Search

- hive
- mssqldemo
  - db_securityadmin
  - dbo (3)
  - db_datareader
  - db_backupoperator
  - sys
  - guest
  - db_owner
  - db_ddladmin
  - db_denydatareader
  - db_datawriter

Datasets

Search

3 of 8 Available Datasets selected

	Name ↓
☑	call_center
☑	call_center_vw
☑	catalog_sales
□	customer
□	customer_address
□	customer_demographics
□	date_dim

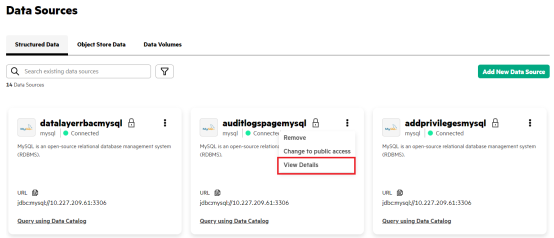
7. In the **Datasets** area, select the tables, views, or buckets that you want to revoke access to.
   - If you are only revoking access to one table, view, or bucket, select **No Access** in the **Access Type** column for the table, view, or bucket.
   - If you are revoking access to multiple tables, views, or buckets, select the tables, views, or buckets and then use the **Bulk Access** dropdown (to the right of the **Search** field) and select **No Access**.
8. Click **Update Privilege**. The system displays the message:

```
Updated privileges for the user:
<user-name>
```

Remove Privileges

To revoke all access to a specific data source for members, complete the following steps:

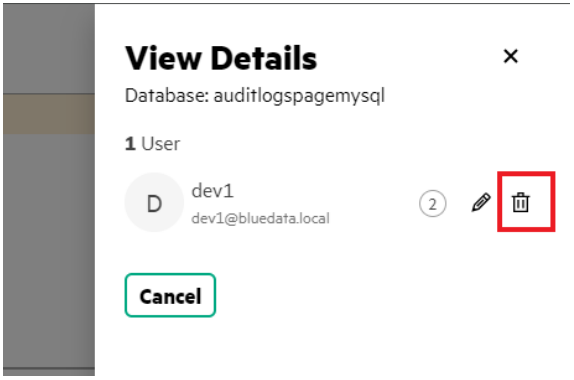
1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Data Engineering > Data Sources**.
3. Select the **Structured Data** or **Object Store Data** tab.
4. In the data source tile, click the three-dots.



MySQL is an open-source relational database management system (RDBMS).

jdbc:mysql://10.227.209.61:3306

5. Select **View Details**.



6. On the **View Details** screen, locate the user whose access to the data source you want to revoke.
7. Click **Remove Privileges (delete icon)**.

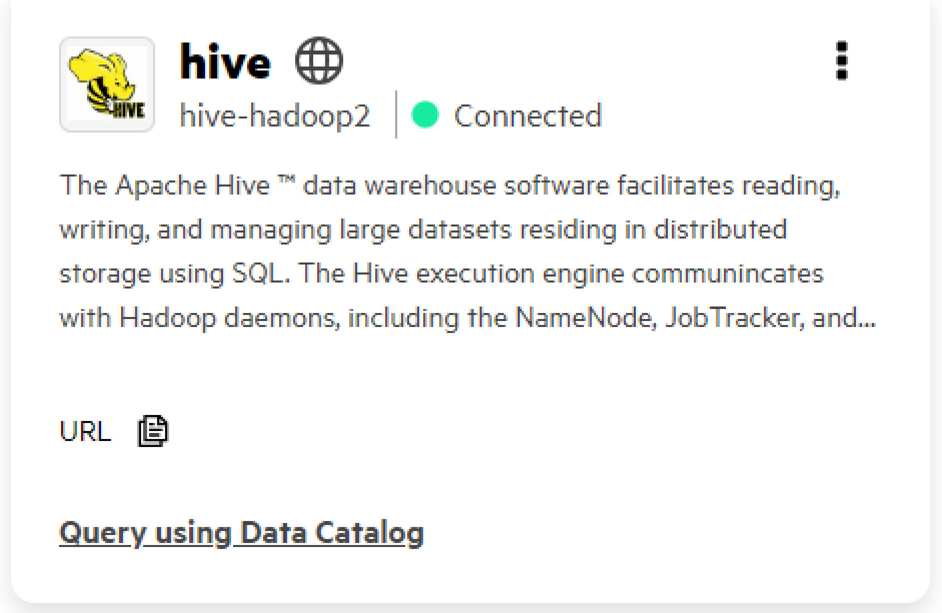
Access Indicator Labels

When users (admins and members) sign in to HPE Ezmeral Unified Analytics Software and go to **Data Engineering > Data Sources**, they see tiles for all of the connected data sources on the **Data Sources** screen. The tiles have icons and labels that indicate whether a data source is accessible or not.

The following sections describe the access indicators that admins and members see on data source tiles.

Admins

Admins have full access to all data sources regardless of the icon displayed. The icon that an admin sees in the tile indicates whether a data source is publicly accessible or not. If an admin makes a data source publicly accessible (read and write access for all members), the data source tile displays a globe icon next to the data source name, indicating global access.



hive

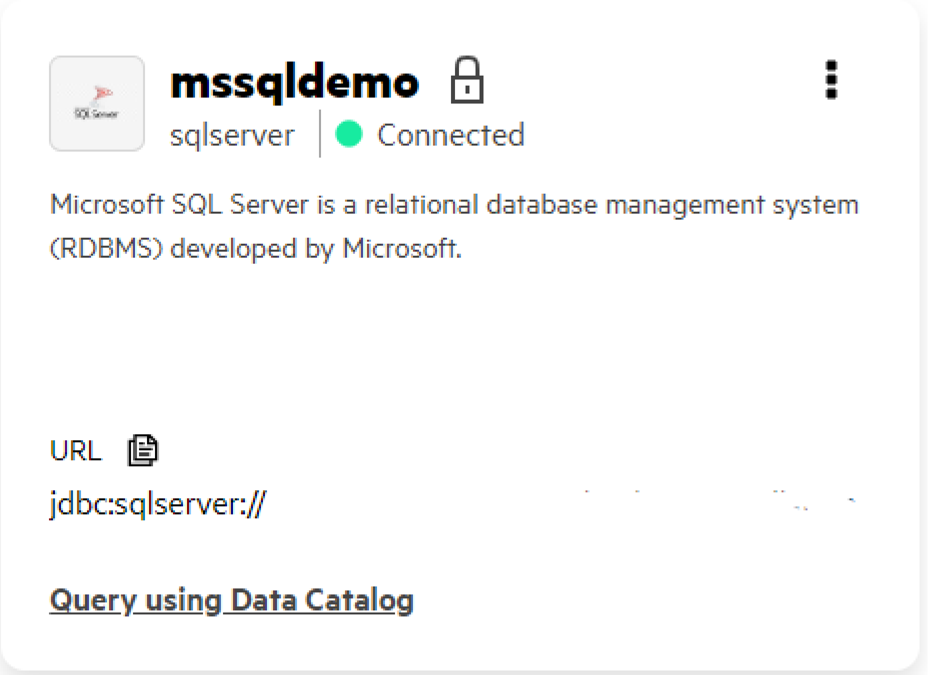
hive-hadoop2 | Connected

The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. The Hive execution engine communincates with Hadoop daemons, including the NameNode, JobTracker, and...

URL

Query using Data Catalog

Otherwise, a locked padlock icon displays.



mssqldemo

sqlserver | Connected

Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft.

URL

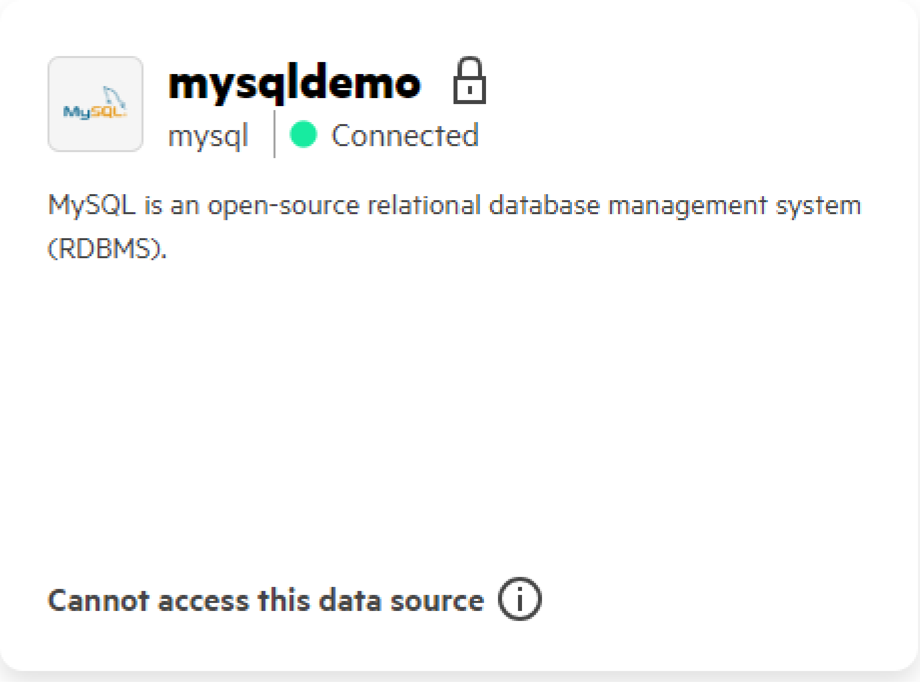
jdbc:sqlserver://

Query using Data Catalog

The locked padlock indicates that an admin must grant members access to the data source. All admins have access to the data source.

Members

When members **do not** have access to a data source, the data source tile shows a locked padlock icon and says *Cannot access this data source*.



mysqldemo

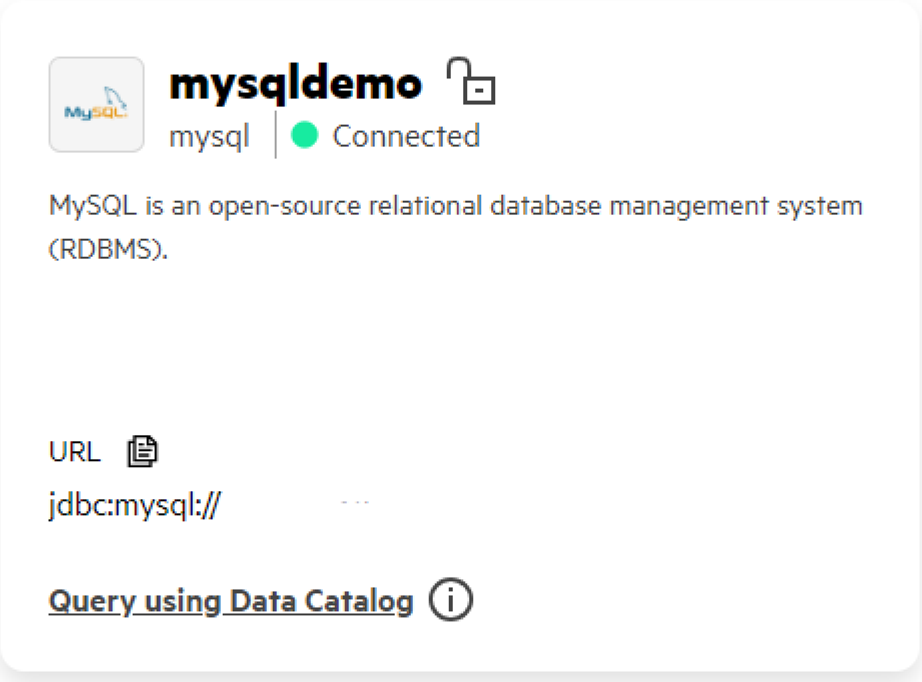
mysql | ● Connected

MySQL is an open-source relational database management system (RDBMS).

Cannot access this data source ⓘ

Any attempts to access the data results in an access denied error.

When members have access to a data source, the padlock icon in the data source tile is unlocked and the tile displays the *Query using Data Catalog* link.



mysqldemo

mysql | ● Connected

MySQL is an open-source relational database management system (RDBMS).

URL

jdbc:mysql://

Query using Data Catalog ⓘ

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 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.

Security Resources

Provides additional resources related to HPE Ezmeral Software security.

- HPE Support Center
- HPE Security Bulletin Library
- Get connected with updates from HPE
- HPE Sales and Support

Observability

Describes observability in HPE Ezmeral Unified Analytics Software.

Metering and Billing

Describes metering and billing in HPE Ezmeral Unified Analytics Software.

Metering involves the measurement and collection of metrics from monitored targets. The metering process is essential for billing purposes and to monitor the usage of key resources within the Kubernetes cluster, such as pods, PVCs, and jobs. Prometheus facilitates the monitoring of these components.

Prometheus is an open-source monitoring and alerting system designed for gathering time-series data. Prometheus provides a flexible querying language (PromQL) to retrieve and analyze metrics.

HPE Ezmeral Unified Analytics Software consists of platform components and application components such as Kubeflow, Airflow, and others. In HPE Ezmeral Unified Analytics Software, metering refers to tracking the resource usage (in particular, the CPU and GPU usage) of the components. Metering is used to support accurate billing and to observe the aggregated usage of these components over a specific time period.

There are four categories of component workloads (pods):

Application core workloads

The application core workloads are initialized by the application installer. These workloads allow the instantiation of user-initiated workloads, such as notebooks and jobs. Some examples of these workloads include notebook controllers, EzPresto core querying workloads, and inference job controllers.

Application user workloads

Application user workloads refers to the user-initiated workloads. Some examples of these workloads include notebooks, inference jobs, and Spark jobs.

Infrastructure or platform workloads

The platform workloads perform core-platform functions such as monitoring UI, managing users, connecting to Data Fabric, and so on.

Bring your own application workloads

Bring your own application workloads refers to the workloads that are imported using **Import Applications** functionality. You must manually specify the labels for workloads in the workload resource yaml files.

Configure the resource metadata with the following labels:

- `hpe-ezua/type="vendor-service"`
- `hpe-ezua/app="<name_of_the_app>"`

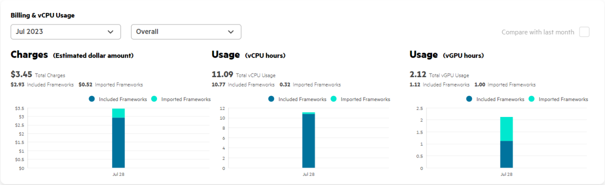
Pod labels are used to categorize each workload into the four types of workloads. CPU usage is measured every five seconds. After obtaining the initial data, application-related pod labels are used to retrieve the aggregate application usage. For example, the total Kubeflow usage can be calculated by summing up the usage of all pods associated with Kubeflow.

Billing and Metering

In HPE Ezmeral Unified Analytics Software, billing is performed on an hourly basis. A cron job runs at the top of each hour to compute the average of the aggregated application usage. Daily and monthly aggregations are also calculated, and custom metrics are pushed back to Prometheus. The billing feature leverages these custom metrics to calculate monthly charges.

Dashboard

An administrator can view the **Billing & vCPU Usage** dashboard on the HPE Ezmeral Unified Analytics Software homepage. An overview of the daily and monthly aggregated usages for applications is available through the **Billing & vCPU Usage** dashboard.



Billing & vCPU Usage

Jul 2023 | Overall | Compare with last month □

Charges (Estimated dollar amount)

$3.45 Total Charges
$2.93 Included Frameworks $0.52 Imported Frameworks

Usage (vCPU hours)

11.09 Total vCPU Usage
10.77 Included Frameworks 0.32 Imported Frameworks

Usage (vGPU hours)

2.12 Total vGPU Usage
1.12 Included Frameworks 1.00 Imported Frameworks

In this dashboard, you see the daily usage for the **Overall** category, which includes all applications, including imported applications. To view the resource usage charts for a specific application, select the application from the dropdown.

The left side provides the total estimated charges for the month with the daily breakdown. The right side displays the usage data.

In HPE Ezmeral Unified Analytics Software, GPU usage is metered per application, and each vGPU counts as an individual GPU for metering.

The GPU metric used for metering is `DCGM_FI_PROF_GR_ENGINE_ACTIVE`.

The GPU metrics sampling interval is every five seconds.

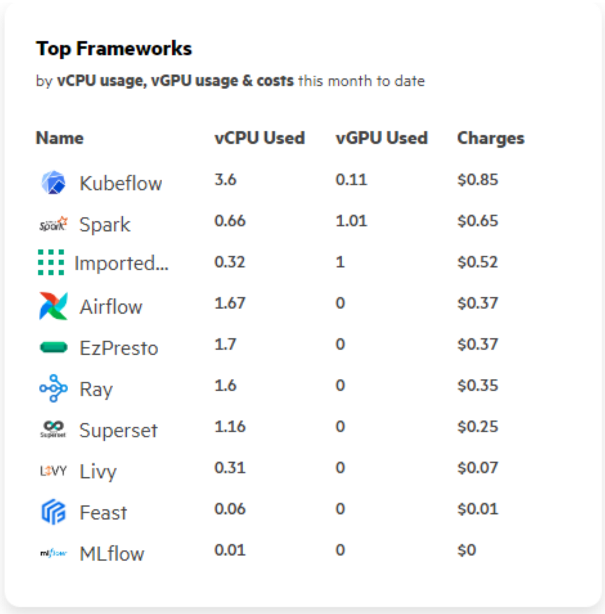
The hourly usage is the average GPU utilization over a one-hour period.

For example: With seven small vGPUs and four applications using six vGPUs as follows:

Applications	Pods -vGPU	Avg 1hr per vGPU	Avg 1hr utilization per application
App1	Pods11 - vGPU0	0.8	1.3
	Pods12 - vGPU1	0.5	
App2	Pods21 - vGPU2	0.8	0.8
App3	Pods31 - vGPU3	0.5	0.5
App4	Pods41 - vGPU4	0	0.5
	Pods42 - vGPU5	0.5	

Total GPU utilization for the billing record is 1.3 + 0.8 + 0.5 + 0.5 = 3.1 vGPU-hour

Top Frameworks provides the list of monthly aggregated usage charges data for all applications in HPE Ezmeral Unified Analytics Software.



Top Frameworks

by **vCPU usage, vGPU usage & costs** this month to date

Name	vCPU Used	vGPU Used	Charges
Kubeflow	3.6	0.11	$0.85
Spark	0.66	1.01	$0.65
Imported...	0.32	1	$0.52
Airflow	1.67	0	$0.37
EzPresto	1.7	0	$0.37
Ray	1.6	0	$0.35
Superset	1.16	0	$0.25
Livy	0.31	0	$0.07
Feast	0.06	0	$0.01
MLflow	0.01	0	$0

The total usage for billing is aggregated for all included and imported applications.

Monitoring

Describes monitoring in HPE Ezmeral Unified Analytics Software.

Monitoring and alerting play an integral role in the observability framework. They involve monitoring the health, performance, and resource utilization of a Kubernetes cluster and its components. Administrators receive alerts about potential issues, which helps maintain optimal cluster and application operations and enables prompt responses to critical events.

NOTE: You cannot configure notifications or turn off notifications. You must view alerts and notifications in HPE Ezmeral Unified Analytics Software.

Model Monitoring

Model monitoring is the process of continuously observing and analyzing the performance and behavior of machine learning models deployed in production environments. It is a critical aspect of the machine learning lifecycle that ensures models remain reliable, accurate, and aligned with the intended objectives.

Model monitoring involves the collection, analysis, and visualization of various metrics and data related to the model's performance and data characteristics. It is an iterative process that helps ensure model reliability and enables timely adjustments or updates to maintain optimal performance. Model monitoring plays a crucial role in building trust in machine learning systems and making informed decisions based on model outputs.

Model monitoring metrics are essential to track and measure the performance of the deployed models.

In HPE Ezmeral Unified Analytics Software, you can use KServe or MLflow for monitoring operational performance and whylogs for functional performance.

Collected Metrics

Knative metrics

Knative Serving does not have built-in native support for model monitoring metrics. You can integrate Kserve with other monitoring and observability tools to collect

	and analyze metrics related to the performance and behavior of your deployed models.(Prometheus, Grafana, Kiali, ESK etc) To learn more, see Importing dashboards to Grafana. The following metrics are collected via KServe: • Knative Serving: Revision HTTP Requests • Knative Serving: Scaling Debugging • Knative Serving: Revision CPU and Memory Usage • Knative: Reconciler • Knative Serving: Control Plane Efficiency
MLflow metrics	Use OTel to collect and export the telemetry data from MLflow applications, including metrics, and traces to third-party or external monitoring systems such as Prometheus, Jaeger, or Grafana for analysis and visualization. To learn more, see Configuring Endpoints on page 135. The following metrics are collected via MLflow: • `mlflow_http_request_total`: Total number of incoming HTTP requests. • `mlflow_http_request_duration_seconds_sum`: Total duration in seconds of all incoming HTTP requests. • `mlflow_http_request_duration_seconds_count`: Total count of all incoming HTTP requests.

Model Monitoring with whylogs

NOTE: This feature is presented as a developer preview. Developer previews are not tested for production environments, and should be used with caution.

whylogs is an open-source library for logging any kind of data. With whylogs, you can generate summaries of your datasets (data profiles) that you can use to:

- Track changes in the dataset and detect data drifts in the model input features.
- Create data constraints to validate data quality in model inputs or in a data pipeline.
- Detect training-serving skew, concept drift, and model performance degradation.
- Perform exploratory data analysis of massive datasets.
- Track data distributions and data quality for ML experiments.
- Standardize data documentation practices across the organization.
- Visualize the key summary statistics about the datasets in HTML and JSON file formats.

To learn more about whylogs, see whylogs documentation.

HPE Ezmeral Unified Analytics Software enables you to use an open-source library called whylogs in the preview environment. whylogs is integrated into the Notebook as a third-party package. You can access data from external S3 object store when using whylogs for monitoring. To learn more about accessing data, see Accessing Data in External S3 Object Stores on page 292.

The following applications and frameworks support whylogs in HPE Ezmeral Unified Analytics Software:

- Airflow. See Using whylogs with Airflow on page 363.
- MLflow. See Using whylogs with MLflow on page 419.

  NOTE: HPE Ezmeral Unified Analytics Software supports external data sources such as AWS, MinIO for whylogs with MLflow. You can not use S3 proxy as a data source.

- Ray. See Using whylogs with Ray on page 433.
- Spark. See Using whylogs with Spark on page 407.

Alerting

Describes alerting in HPE Ezmeral Unified Analytics Software.

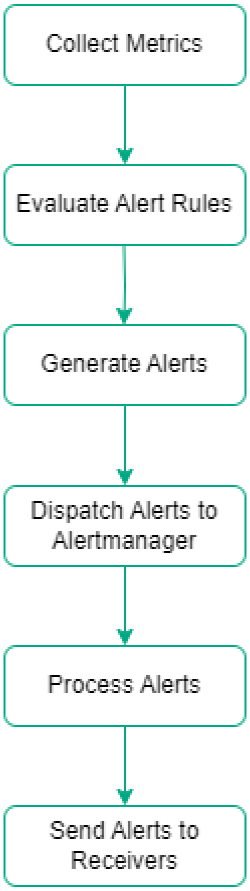
An alert in HPE Ezmeral Unified Analytics Software is a system notification that informs you of issues, warnings, and updates. Unified Analytics uses Prometheus to monitor and collect metrics from nodes, system processes, and applications that run in an HPE Ezmeral Unified Analytics Software cluster. Unified Analytics generates alerts based on the metrics collected. An Alertmanager in Unified Analytics enables you to control the behavior of alerts, for example, silence specific alerts or send notifications to a specific user when the system raises an alert.

To learn about Prometheus and Alertmanager in detail, see the Prometheus and Alertmanager documentation.

The alert system in HPE Ezmeral Unified Analytics Software is comprised of several components. The following sections include an architectural diagram, component descriptions, and alerting workflow.

Alerting Worflow

The following is an overview of the alerting workflow along with a detailed description in HPE Ezmeral Unified Analytics Software.



Collect Metrics

Prometheus scrapes metrics from targets exposed by exporters. For example, Prometheus collects the CPU usage metrics from servers via Node Exporter, and database query latency from MySQL via Mysqld Exporter.

Evaluate Alert Rules

Prometheus continuously evaluates the alerting rules that are defined in PromQL against the collected metrics.

Generate Alerts

If the condition for a rule is met, Prometheus generates an alert. For example:

- The following alert rules send notifications if an average API server error rate exceeds 5 per minute.

  ```
  avg(http_requests_total{job="api_se
  rver", status_code="500"}) by
  (job) > 5
  ```

- The following alert rules send notifications if the disk has less than 10GB free.

  ```
  node_filesystem_avail_bytes{mountpo
  int="/"} < 10 * 1024 * 1024 * 1024
  ```

Dispatch Alerts to Alertmanager

Prometheus sends the generated alerts to the configured Alertmanager.

Process Alerts	Alertmanager deduplicates, groups, and routes the alerts based on configured rules.
Send Notifications to Receivers	Alertmanager sends notifications to the appropriate recipients through the designated channels.

Resource Events Alerting

Alerts are triggered for the following events:

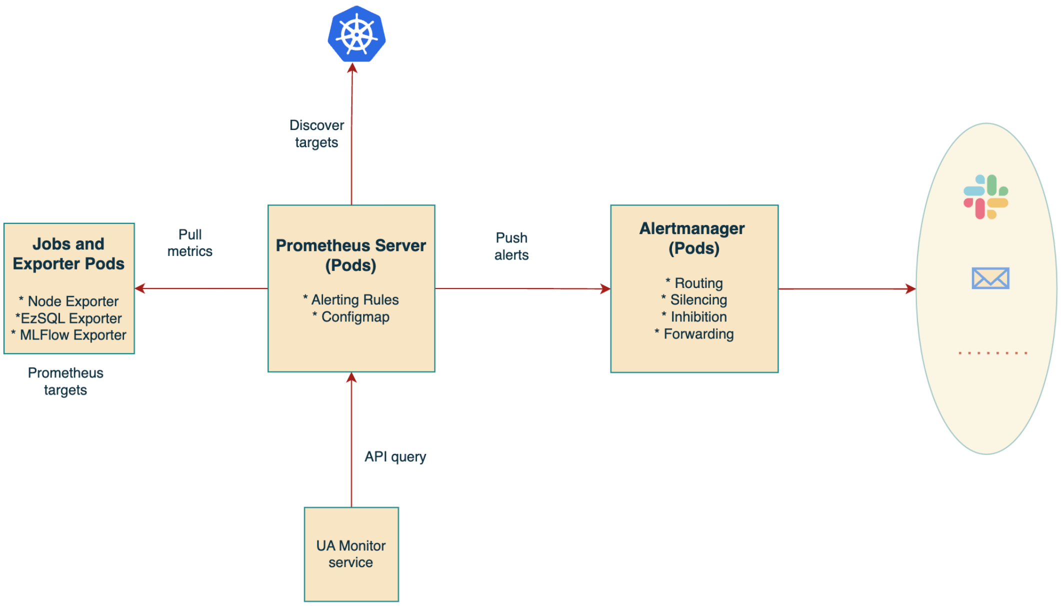
- High resource CPU usage
- High resource memory usage
- Unusual pod restart
- Pods not in running state
- PVC status not Bound
- Failed jobs
- Failed cronjobs
- Node failures
- Unsual node memory or CPU usage behavior
- Kubelet failures
- Node filesystem issues
- Node network issues
- Prometheus issues

To find the list of alerts generated in HPE Ezmeral Unified Analytics Software, see List of Alerts on page 278.

Alerting Architecture and Components

Shows the alerting architecture diagram and describes the alerting components included in HPE Ezmeral Unified Analytics Software.

The following image shows the alerting architecture in HPE Ezmeral Unified Analytics Software.



The following list describes the alerting components included in HPE Ezmeral Unified Analytics Software:

Prometheus

Prometheus is an open-source monitoring and alerting system that specializes in collection of the time-series data.

Pull-based monitoring: Prometheus actively scrapes metrics from the configured targets at regular intervals using pull-based monitoring.

PromQL: Prometheus uses the powerful query language called PromQL for data analysis and defining the alert rules. For example: The following alert rules send notifications if there are more than 100 HTTP requests with 500 status code on the API server at a specific time.

```
http_requests_total{job="api_server",
status_code="500"} > 100
```

Alertmanager

Alertmanager uses the metrics data and labels generated by Prometheus to enrich notifications with relevant information.

Alert routing: Alertmanager routes notifications to different communication channels such as email, Slack, webhooks, and others. For example, you can configure the routing rules to route the critical alerts to PagerDuty for immediate action and route all other alerts with low priority to a Slack channel.

Deduplication and grouping: Alertmanager groups alerts to reduce noise, resulting in an organized presentation of issues. For example, multiple alerts about high disk usage on different servers within a cluster are grouped as a single alert.

	Silencing and inhibition: Alertmanager suppresses alerts for a certain time. For example, you can silence alerts during the planned maintenance period. You can also inhibit alerts based on dependencies such as not sending out alerts on database issues when the primary network is down.
Exporters	Exporters are specialized software that exposes metrics from different systems and applications in a format Prometheus can understand. **Node Exporter:** Exposes hardware and operating system metrics such as CPU, memory, disk, and network. **Mysqld Exporter:** Exposes metrics from MySQL databases such as queries per second, connections, and replication status. **Kubernetes Exporter:** Exposes metrics from a Kubernetes cluster.

Configuring Alert Forwarding

Describes how to forward alerts via SMTP or Slack and configure the alert policies by updating the `alertmanager_config.yaml` file.

Prerequisites:

- Sign in to HPE Ezmeral Unified Analytics Software as an administrator.
- Have access to a terminal where `kubectl` is installed to interact with your Kubernetes cluster.

Steps:

To configure alert policies and forward alerts via SMTP or Slack, complete the following steps:

1. On the Prometheus namespace, run the following command to list secrets:

   ```
   kubectl get secrets -n prometheus
   ```

2. In the list of secrets, locate the secret named `alertmanager-prometheus-kube-prometheus-alertmanager`.

3. Extract the base64-encoded Alertmanager configuration from the `alertmanager_config.yaml` file:

   ```
   kubectl get secret
   alertmanager-prometheus-kube-prometheus-alertmanager -n prometheus -o
   jsonpath="{.data.alertmanager\.yaml}" | base64 -d >
   alertmanager_config.yaml
   ```

4. Open the `alertmanager_config.yaml` file in a text editor.

5. Update the email settings under the `email_configs` section and the Slack settings under the `slack_configs` section. You can also configure the additional notification channels as required. To learn more about these settings, see Understanding the Prometheus Alertmanager Configuration File on page 271.

6. Delete the existing Alertmanager configuration secret:

   ```
   kubectl delete secret
   alertmanager-prometheus-kube-prometheus-alertmanager -n prometheus
   ```

7. Create the updated Alertmanager configuration secret:

   ```
   kubectl create secret
   generic alertmanager-prometheus-kube-prometheus-alertmanager -n
   prometheus --from-file=alertmanager.yaml=alertmanager_config.yaml
   ```

8. Force restart the Alertmanager pod to reload its configuration:

   ```
   kubectl delete pod
   alertmanager-prometheus-kube-prometheus-alertmanager-0 -n prometheus
   ```

Results:

You have updated the `alertmanager_config.yaml` file to forward alerts via SMTP or Slack.

Configuring Templates and Filtering Alerts

Describes the process of configuring templates and filtering alerts in Alertmanager. You can follow these steps to customize the content and structure of email notifications generated by Alertmanager.

Prerequisites:

- Sign in to HPE Ezmeral Unified Analytics Software as an administrator.
- Have access to a terminal where `kubectl` is installed to interact with your Kubernetes cluster.

Steps:

To customize the content and structure of email notifications generated by Alertmanager, perform the following steps:

1. Create two separate template files: `custom_mail_subject.tmpl` and `custom_mail_html.tmpl`. You will use `custom_mail_subject.tmpl` to create a template of the subject and `custom_mail_html.tmpl` to edit the html.

   In `custom_mail_subject.tmpl`, add the following content:

   ```
   [Alerting] {{ .CommonLabels.alertname }}
   ```

   In `custom_mail_html.tmpl`, add the following content:

   ```
   <html>
   <body>
     <h3>{{ .CommonLabels.alertname }}</h3>
     <p>{{ .CommonAnnotations.description }}</p>
   </body>
   </html>
   ```

2. Create a ConfigMap in the `prometheus` namespace to store the templates:

   ```
   kubectl create configmap alertmanager-templates -n prometheus \
     --from-file=custom_mail_subject.tmpl \
     --from-file=custom_mail_html.tmpl
   ```

3. Modify the `prometheus-kube-prometheus-alertmanager` Custom Resource (CR) in the `prometheus` namespace to mount the ConfigMap:

   ```
   kubectl edit alertmanager prometheus-kube-prometheus-alertmanager -n
   prometheus -o yaml
   ```

   In the `spec` section, add the following:

   ```
   spec:
     volumes:
       - name: alertmanager-templates
         configMap:
           name: alertmanager-templates
     containers:
       - name: alertmanager
         volumeMounts:
           - name: alertmanager-templates
             mountPath: /etc/alertmanager/templates
   ```

4. Get the Secret containing the Alertmanager configuration:

   ```
   kubectl get secret
   alertmanager-prometheus-kube-prometheus-alertmanager -n prometheus -o
   yaml
   ```

5. Extract the base64-encoded Alertmanager configuration from the `alertmanager.yaml` file:

   ```
   echo '<base64_config_data>' | base64 -d > alertmanager.yaml
   ```

6. Open the `alertmanager.yaml` file in a text editor.

7. Add the template reference in `alertmanager.yaml`:

   ```
   templates:
     - '/etc/alertmanager/templates/*.tmpl'
   ```

8. Add or update the `email_configs` section in `alertmanager.yaml`:

   ```
   receivers:
   - name: 'email_receiver'
     email_configs:
     - to: 'email@xyz.com'
       html: '{{ template "custom_mail_html.tmpl" . }}'
       headers:
         subject: '{{ template "custom_mail_subject.tmpl" . }}'
   ```

   NOTE: This template is applied to alerts that are not specifically routed elsewhere.

9. Add additional receivers and routes in `alertmanager.yaml`:

   ```
   global:
     resolve_timeout: 5m
     smtp_from: <email_from>
     smtp_smarthost: <smtp_host>
     smtp_require_tls: <true/false>

   route:
     group_by: ['alertname']
     group_wait: 30s
     group_interval: 5m
     repeat_interval: 4h
     receiver: 'kubeflow_receiver'

     routes:
       - match:
           alertname: 'Kubeflow job failing'
         receiver: 'kubeflow_receiver'
       - match:
           alertname: 'Airflow job failing'
         receiver: 'airflow_receiver'

   receivers:
   - name: 'kubeflow_receiver'
     email_configs:
     - to: 'kubeflow_admin@hpe.com'
       html: '{{ template "kubeflow_html.tmpl" . }}'
       headers:
         subject: '{{ template "kubeflow_html.tmpl" . }}'
   - name: 'airflow_receiver'
     email_configs:
     - to: 'admin_airflow@hpe.com'
       html: '{{ template "airflow_html.tmpl" . }}'
       headers:
         subject: '{{ template "airflow_subject.tmpl" . }}'

   templates:
   - '/etc/alertmanager/templates/*.tmpl'
   ```

10. Encode the updated `alertmanager.yaml` file to Base64:

    ```
    cat alertmanager.yaml | base64 -w0
    ```

11. Replace the existing Base64 data in the Secret by editing the Secret:

    ```
    kubectl edit secret
    alertmanager-prometheus-kube-prometheus-alertmanager -n prometheus
    ```

12. Restart the Prometheus Operator pod and then the Alertmanager pod to apply the changes:

    ```
    kubectl delete pod -n prometheus -l
    alertmanager=prometheus-kube-prometheus-alertmanager
    kubectl delete pod -n prometheus -l app=kube-prometheus-stack-operator
    ```

Results:

You have updated the `alertmanager.yaml` file to forward alerts via different receivers.

More information

Understanding the Prometheus Alertmanager Configuration File on page 271
Provides an overview of the key components of the Prometheus Alertmanager configuration file, field descriptions, and configuration examples.

Understanding the Prometheus Alertmanager Configuration File

Provides an overview of the key components of the Prometheus Alertmanager configuration file, field descriptions, and configuration examples.

The `alertmanager.yaml` file stores Alertmanager configurations. Configurations in this file specify how Alertmanager routes and delivers the alerts received from Prometheus. You can configure the `alertmanager.yaml` file to send SMTP or Slack notifications.

To learn about acessing and configuring the `alertmanager.yaml` file to send notifications, see Configuring Alert Forwarding on page 267 and Configuring Templates and Filtering Alerts on page 268.

The following code block shows the sample `alertmanager.yaml` file:

```
global:
  # SMTP configuration for email notifications (if needed)
  smtp_smarthost: 'mailserver.example.com:587'
  smtp_from: 'alertmanager@example.com'
  smtp_auth_username: 'alertmanager'
  smtp_auth_password: 'your_password'

  # Other global settings like resolve_timeout, http_config, etc.

route:
  # Default receiver for alerts
  receiver: 'default-receiver'

  # Labels used for grouping alerts
  group_by: ['alertname', 'instance', 'severity']

  # Timing settings (group_wait, group_interval, repeat_interval)
  # You can have nested 'routes' for more complex routing logic

receivers:
- name: 'default-receiver'
  email_configs:
    to: 'ops-team@example.com'
    # ... other email settings ...

- name: 'slack-notifications'
  slack_configs:
    api_url: 'https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK'
    channel: '#alerts'
    # ... other Slack settings ...

- name: 'pagerduty-notifications'
  pagerduty_configs:
    service_key: 'your_pagerduty_service_key'
    # ... other PagerDuty settings ...

# ... more receivers as needed (webhooks, OpsGenie, etc.)

inhibit_rules:
  # Rules to suppress alerts based on other alerts
  # Example:
  - source_match:
      severity: 'critical'
    target_match:
      severity: 'warning'
    # Suppress 'warning' alerts if a 'critical' alert is also firing

templates:
  # Paths to template files for customizing notifications
  - '/etc/alertmanager/templates/*.tmpl'
```

Field Descriptions of the `alertmanager.yaml` File

The following table describe fields of the `alertmanager.yaml` file:

Fields	Descriptions
`global:`	Defines general settings for Alertmanager.
`resolve_timeout:`	Specifies the time to wait for alerts to be acknowledged as resolved. For example, e.g., `resolve_timeout: 5m`)
`smtp_smarthost:`	Address of the SMTP server. For example, `smtp_smarthost:'mail.example.com:25'`.
`smtp_from:`	Specifies the email address of sender.
`smtp_auth_username:`	Username for SMTP authentication.
`smtp_auth_password:`	Password for SMTP authentication.
`http_config:`	(Optional) For configuring TLS, authentication, and other in Alertmanager web interface.
`receivers:`	Defines how alerts are received. You can configure multiple receivers for different notification methods such as email, Slack, and others.
`name:`	Specifies a descriptive name for the receiver configuration. This is a unique name for your notification channel. This name is also used for routing alerts.
`email_configs:`	Specifies configurations for the email notifications.
`to:`	Specifies the comma-separated list of email addresses to receive alerts.
`from:`	Specifies the email address from which notifications are sent.
`smarthost:`	Specifies the hostname and port of your SMTP server. For example, smtp.example.com:587
`auth_username:`	Username for SMTP authentication.
`auth_password:`	Password for SMTP authentication.
`html:`	(Optional) Format email body as HTML.
`require_tls:`	(Optional) Enforce TLS for sending email.
`slack_configs:`	Specifies configurations for the Slack notifications.
`channel:`	Specifies the Slack channel to receive alerts.
`api_url:`	The webhook URL generated from your Slack Incoming Webhook integration.
`title:`	Specifies the title of the Slack notifications.

Fields	Descriptions
`text:`	Specifies the descriptive text for the Slack notifications.
`webhook_configs:`	Specifies configurations for a general receiver.
`url:`	Specifies the endpoint URL to send notifications.
`http_config:`	(Optional) Specifies configurations for HTTP authentication, proxies, and others.
`pagerduty_configs:`	Specifies configurations for sending the PagerDuty notifications.
`service_key:`	Specifies the PagerDuty integration key.
`route:`	(Optional) Specifies advanced routing rules based on alert characteristics (for example, severity labels). You can specify which receivers receive specific types of alerts.
`group_by:`	List of labels to groups similar alerts together. For example, `['alertname', 'cluster', 'service']`.
`group_wait:`	Specifies the time to wait before sending the initial notification. For example, `30s`.
`group_interval:`	Specifies the time between sending the grouped notifications. For example, `5m`.
`repeat_interval:`	Specifies time between repeat notifications for unresolved alerts. For example, `3h`.
`routes:`	(Optional) For setting the individual routing rules for different alert types.
`match:`	Filters alerts by matching a specific label to a value. For example, filter by `severity: critical`.
`match_re:`	Filters alerts by applying a regular expression to match a label for more advanced matching.
`receiver:`	Specifies the name of the previously defined receiver to determine the destination for the alert notifications.

Configuring Basic Email Alerts

The following example shows the basic email alerting configurations.

```
global:
  smtp_smarthost: 'smtp.example.com:587'
  smtp_from: 'alertmanager@example.com'
  smtp_auth_username: 'alertmanager'
  smtp_auth_password: 'your_password'

route:
  receiver: 'email-alerts'

receivers:
- name: 'email-alerts'
  email_configs:
    to: 'team@example.com'
```

This configuration sends all alerts to the specified email addresses. To customize,

- Modify the SMTP settings with your email server details such as, hostname, port, credentials.
- Set `to:` under `receivers:` with the email address where you want to receive alerts.

Configuring Slack Notifications

The following example shows the Slack notifications configurations.

```
route:
  receiver: 'slack-alerts'

receivers:
- name: 'slack-alerts'
  slack_configs:
    api_url: 'https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK'
    channel: '#alerts'
    text: "Firing: {{ .CommonAnnotations.summary }}"
```

This configuration sends all alerts to the specified Slack channel. To customize,

- Set `api_url:` with your Slack webhook URL.
- Set `channel:` with your target channel.
- Modify the `text:` property with the alert message using Go templating.

Configuring Multiple Receivers with Routing

The following example shows configurations for multiple receivers using routing.

```
route:
  group_by: ['alertname', 'severity']
  receiver: 'default-receivers'
  routes:
    - match:
        severity: critical
      receiver: 'pagerduty-notifications'

receivers:
- name: 'default-receivers'
  email_configs:
    to: 'team@example.com'
- name: 'pagerduty-notifications'
  pagerduty_configs:
    service_key: 'your_pagerduty_service_key'
```

This configuration routes the critical alerts to PagerDuty and all the other alerts to the email address. To customize,

- Set `service_key:` with the actual integration key.
- Modify `email_configs:` as required.
- Configure the routing rules based on your alert labels.

Configuring Alerts Silencing

The following example sends alerts to both email and Slack.

```
route:
  receiver: 'team-alerts'

receivers:
- name: 'team-alerts'
  email_configs:
    to: 'team@example.com'
  slack_configs:
    api_url: 'https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK'
    channel: '#alerts'

# Silence alerts during scheduled maintenance windows
inhibit_rules:
  - source_match:
      severity: 'warning'
    target_match:
      job: 'kubeflow'
    equal: ['maintenance']
```

This configuration suppresses alerts of type `warning` for the `database` job if they have the label `maintenance=true`. This is useful for avoiding unwanted noise during the planned maintenance period. To customize,

- Adjust the `source_match:` and `target_match:` sections to target specific alerts for silencing.
- Use different labels and values to match your alerting setup.

Configuring Inhibition Rules

The following example shows configurations for suppressing related alerts.

```
route:
  receiver: 'team-alerts'
  # ... receiver definitions for email, Slack, etc. ...

inhibit_rules:
  # Suppress node down alerts if the cluster is down
  - source_match:
      job: 'node'
      severity: 'critical'
    target_match:
      job: 'cluster'
      severity: 'critical'
```

This configuration prevents the `node down` alerts from being sent when the broader `cluster down` alerts are already being sent. You must carefully customize inhibition rules to avoid missing important alerts. To customize,

- Modify the `source_match:` and `target_match:` to specify alerts that suppress others.

Configuring Routing Rules

The following examples describe two routing rules of Alertmanager.

```
route:
  group_by: ['alertname']  # Example: Group alerts with the same name

  routes:
    # Always send critical alerts regardless of time
    - match:
        severity: critical
      receiver: 'critical-alerts'  # Send critical alerts to this receiver

    # Weekend silence for non-critical alerts (email & Slack)
    - match:
        severity: NOT critical  # All non-critical alerts
      receiver: 'weekend-silence'  # This receiver won't send alerts on
weekends
        # Define the time condition for weekends using weekday number
```

```
(0=Sunday)
        mute_intervals:
          - hours: 12-23  # Friday evening silence from 12 PM onwards
          - days:        # Saturday silence all day
              - 6
          - hours: 0-11  # Sunday morning silence until 11 AM
```

In this example, the `critical-alerts` and `weekend-silence` receivers are pre-configured in your `alertmanager.yaml` file with details specifying how they send notifications. You can include additional notification channels in the `weekend-silence` receiver, such as SMS, which might be necessary during weekends.

This example uses weekday numbers (0=Sunday) for the weekend schedule, allowing you to modify this number as needed.

Always send critical alerts

This configuration ensures that critical issues receive immediate attention regardless of the day or time.

This rule matches alerts with `severity: critical` and sends them to the `critical-alerts` receiver.

Silence alerts on weekends

This configuration disables the email and Slack notifications during weekends.

This rule matches all alerts that are `NOT critical` and sends them to the `weekend-silence` receiver. However, this receiver also includes a `mute_intervals` section, which is defined to silence notifications during specific times as follows:

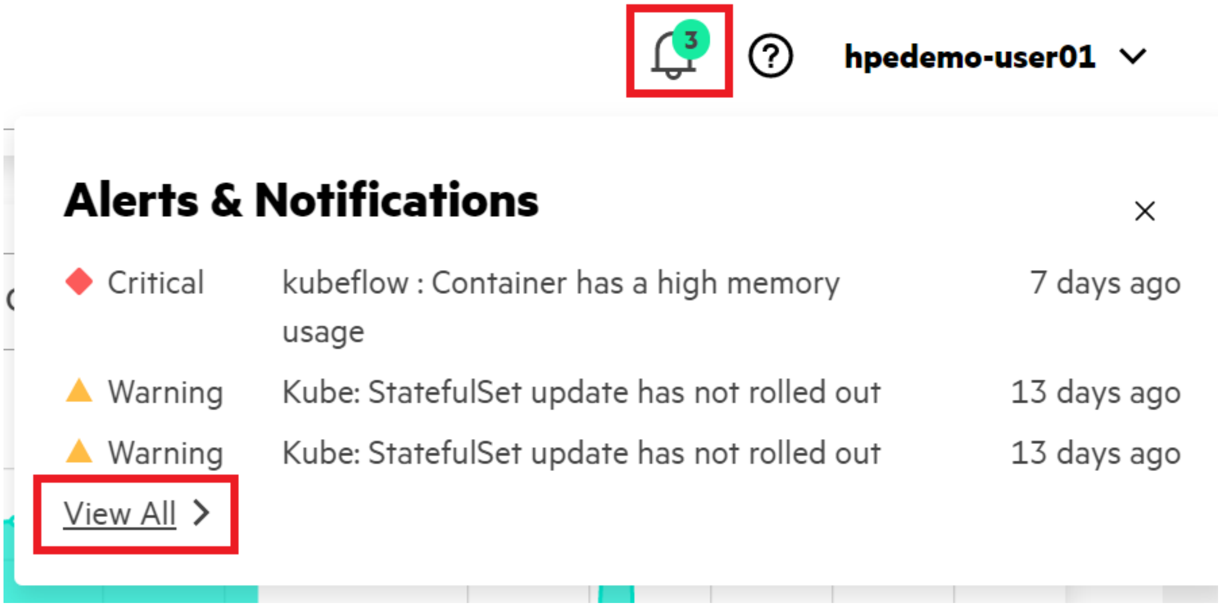
- Friday evenings from 12 PM onwards (hours: 12-23).
- All day Saturday (days: [6]).
- Sunday mornings until 11 AM (hours: 0-11).

Viewing Alerts and Notifications

Describes how to view alerts and notifications in HPE Ezmeral Unified Analytics Software.

To view the list of alerts and notifications in HPE Ezmeral Unified Analytics Software, perform:

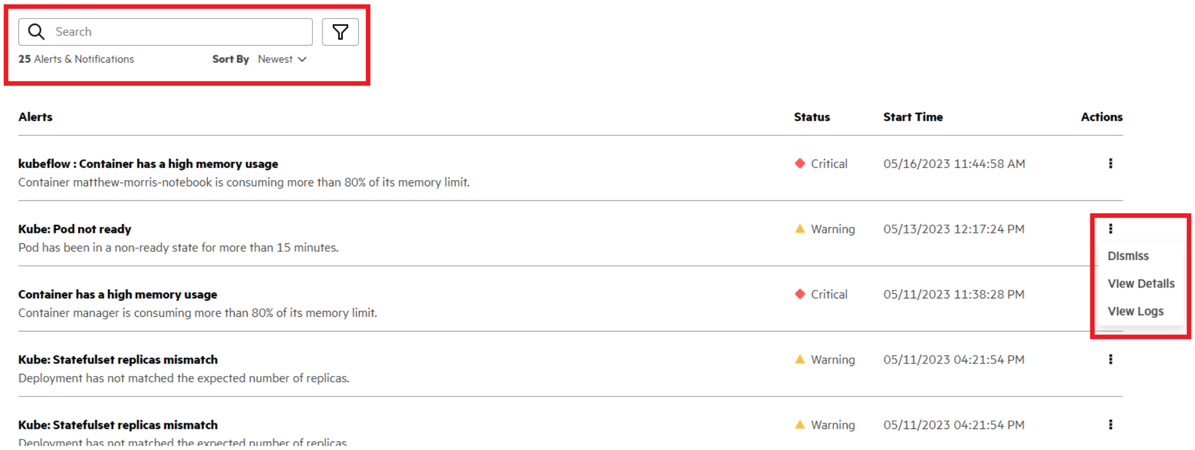
1. Sign in to HPE Ezmeral Unified Analytics Software.
2. Click the **bell** icon on top-right of HPE Ezmeral Unified Analytics Software homepage.



Alerts & Notifications

◆ Critical	kubeflow : Container has a high memory usage	7 days ago
▲ Warning	Kube: StatefulSet update has not rolled out	13 days ago
▲ Warning	Kube: StatefulSet update has not rolled out	13 days ago

3. To view all alerts and notifications, click **View All**.

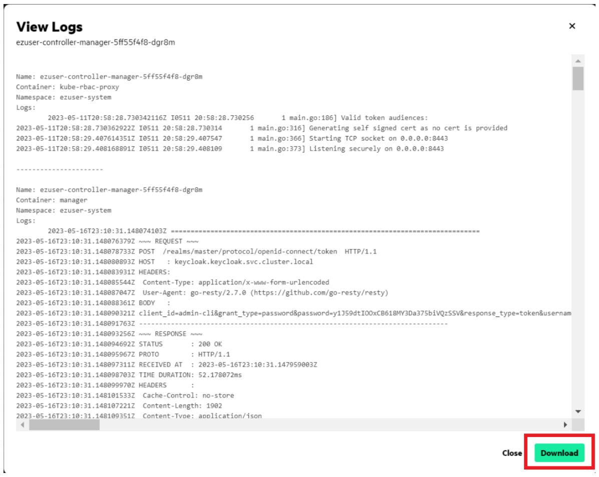


Alerts	Status	Start Time	Actions
kubeflow : Container has a high memory usage Container matthew-morris-notebook is consuming more than 80% of its memory limit.	◆ Critical	05/16/2023 11:44:58 AM	⋮
Kube: Pod not ready Pod has been in a non-ready state for more than 15 minutes.	▲ Warning	05/13/2023 12:17:24 PM	⋮ Dismiss View Details View Logs
Container has a high memory usage Container manager is consuming more than 80% of its memory limit.	◆ Critical	05/11/2023 11:38:28 PM	
Kube: Statefulset replicas mismatch Deployment has not matched the expected number of replicas.	▲ Warning	05/11/2023 04:21:54 PM	⋮
Kube: Statefulset replicas mismatch Deployment has not matched the expected number of replicas.	▲ Warning	05/11/2023 04:21:54 PM	⋮

.

4. To manage alerts, click the **Actions** menu.

Dismiss	To dismiss the alerts, select **Dismiss**.
View Details	To view the decsription and relevant metadata for alerts, select **View Details**.
View Logs	To view the pod logs, select **View Logs**.



```
Name: ezuser-controller-manager-5ff55f4f8-dgr8m
Container: kube-rbac-proxy
Namespace: ezuser-system
Logs:
        2023-05-11T20:58:28.730342116Z I0511 20:58:28.730256       1 main.go:186] Valid token audiences:
2023-05-11T20:58:28.730362922Z I0511 20:58:28.730314       1 main.go:316] Generating self signed cert as no cert is provided
2023-05-11T20:58:29.407614351Z I0511 20:58:29.407547       1 main.go:366] Starting TCP socket on 0.0.0.0:8443
2023-05-11T20:58:29.408168891Z I0511 20:58:29.408109       1 main.go:373] Listening securely on 0.0.0.0:8443

---------------------

Name: ezuser-controller-manager-5ff55f4f8-dgr8m
Container: manager
Namespace: ezuser-system
Logs:
        2023-05-16T23:10:31.148074103Z ==============================================================================
2023-05-16T23:10:31.148076379Z ~~~ REQUEST ~~~
2023-05-16T23:10:31.148078733Z POST  /realms/master/protocol/openid-connect/token  HTTP/1.1
2023-05-16T23:10:31.148080893Z HOST   : keycloak.keycloak.svc.cluster.local
2023-05-16T23:10:31.148083931Z HEADERS:
2023-05-16T23:10:31.148085544Z  Content-Type: application/x-www-form-urlencoded
2023-05-16T23:10:31.148087047Z  User-Agent: go-resty/2.7.0 (https://github.com/go-resty/resty)
2023-05-16T23:10:31.148088361Z BODY   :
2023-05-16T23:10:31.148090321Z client_id=admin-cli&grant_type=password&password=y1J59dtIOOxCB618MY3Da375biVQzSSV&response_type=token&usernam
2023-05-16T23:10:31.148091763Z ------------------------------------------------------------------------------
2023-05-16T23:10:31.148093256Z ~~~ RESPONSE ~~~
2023-05-16T23:10:31.148094692Z STATUS       : 200 OK
2023-05-16T23:10:31.148095967Z PROTO        : HTTP/1.1
2023-05-16T23:10:31.148097311Z RECEIVED AT  : 2023-05-16T23:10:31.147959003Z
2023-05-16T23:10:31.148098703Z TIME DURATION: 52.178072ms
2023-05-16T23:10:31.148099970Z HEADERS      :
2023-05-16T23:10:31.148101533Z  Cache-Control: no-store
2023-05-16T23:10:31.148107221Z  Content-Length: 1902
2023-05-16T23:10:31.148109351Z  Content-Type: application/json
```

To download logs, click **Download**.

5. To search for alerts, use the **Search** bar.

6. To sort alerts, use **Newest** or **Oldest** sort options.

7. To filter alerts, click the **filter** icon.

List of Alerts

Provides a list of platform alerts and application alerts in HPE Ezmeral Unified Analytics Software.

HPE Ezmeral Unified Analytics Software issues the following platform alerts:

- Node Alerts on page 279
- Etcd Alerts on page 280
- Resources Alerts on page 280
- Container Alerts on page 281
- GPU Alerts on page 281
- Storage Alerts on page 281
- Kubelet Alerts on page 282
- Billing Alerts on page 283
- Licensing Alerts on page 283
- Licensing Capacity Alerts on page 283
- Prometheus Alerts on page 283

HPE Ezmeral Unified Analytics Software issues the following application alerts:

- Airflow Alerts on page 285
- Kubeflow Alerts on page 285
- MLflow Alerts on page 285
- Ray Alerts on page 285

- Spark Alerts on page 286
- Superset Alerts on page 286

Node Alerts

Group	Title	Description	Severity
Node.Resources	NodeCPUUsageHigh	Alerts when a node's average CPU utilization over a five-minute window consistently exceeds 80% for 15 minutes.	warning
Node.Resources	NodeMemoryUsageHigh	Alerts if a node's memory usage surpasses 85% of its total memory for 10 minutes.	warning
Node.Requests	NodeMemoryRequestsVsAllocatableWarning80	Warns when memory requests from pods on a node reach 80-90% of the node's allocatable memory, indicating potential problems scheduling new pods.	warning
Node.Requests	NodeMemoryRequestsVsAllocatableCritical90	Triggers a critical alert if memory requests from pods reach or exceed 90% of the node's allocatable memory, indicating a high risk of new pods being stuck in a pending state.	critical
Node.Requests	NodeCPURequestsVsAllocatableWarning80	Warns when CPU requests from pods on a node reach 80-90% of the node's allocatable CPU resources, indicating potential problems scheduling new pods.	warning
Node.Requests	NodeCPURequestsVsAllocatableCritical90	Triggers a critical alert if CPU requests from pods reach or exceed 90% of the node's allocatable CPU resources, indicating a high risk of new pods being stuck in a pending state.	critical
node-exporter	NodeFilesystemSpaceFillingUp	Alerts when a filesystem's free space drops below a threshold, predicting potential exhaustion within 24 hours.	warning
node-exporter	NodeFilesystemSpaceFillingUp	Alerts if a filesystem's free space is critically low, predicting exhaustion within 4 hours.	critical
node-exporter	NodeFilesystemAlmostOutOfSpace	Alerts if a filesystem's free space falls below 5%.	warning
node-exporter	NodeFilesystemAlmostOutOfSpace	Alerts if a filesystem's free space falls below 3%.	critical
node-exporter	NodeFilesystemFilesFillingUp	Alerts when inode usage on a filesystem is predicted to reach exhaustion within 24 hours.	warning
node-exporter	NodeFilesystemFilesFillingUp	Alerts if inode usage is critically low, predicting exhaustion within 4 hours.	critical
node-exporter	NodeFilesystemAlmostOutOfFiles	Alerts if a filesystem's free inodes fall below 5%.	warning
node-exporter	NodeFilesystemAlmostOutOfFiles	Alerts if a filesystem's free inodes fall below 3%.	critical
node-exporter	NodeNetworkReceiveErrs	Alerts if a network interface reports a high rate of receive errors.	warning
node-exporter	NodeNetworkTransmitErrs	Alerts if a network interface reports a high rate of transmit errors.	warning

Group	Title	Description	Severity
node-exporter	NodeHighNumberConntrackEntriesUsed	Alerts if a large percentage of conntrack entries are in use.	warning
node-exporter	NodeTextFileCollectorScrapeError	Alerts if the Node Exporter's text file collector fails to scrape metrics.	warning
node-exporter	NodeClockSkewDetected	Alerts if the node's clock is significantly out of sync.	warning
node-exporter	NodeClockNotSynchronising	Alerts if the node's clock is not synchronizing with NTP.	warning
node-exporter	NodeRAIDDiskFailure	Alerts if a device in a RAID array has failed.	warning
node-exporter	NodeFileDescriptorLimit	Warns when file descriptors usage approaches a defined limit.	warning
node-exporter	NodeFileDescriptorLimit	Critically alerts when file descriptors usage breaches a limit.	critical
node-exporter	NodeCPUHighUsage	Warns when CPU usage exceeds 90% for a sustained period (15 minutes).	info
node-exporter	NodeSystemSaturation	Warns when the average system load per CPU core exceeds a high threshold.	warning
node-exporter	NodeMemoryMajorPagesFaults	Warns when the rate of major memory page faults is high.	N/A

Etcd Alerts

Alert Title	Description	Severity
etcdMembersDown	Alerts when members of the etcd cluster are down or experiencing network connectivity issues.	critical
etcdInsufficientMembers	Alerts when the etcd cluster doesn't have a sufficient number of members to reach quorum.	critical
etcdNoLeader	Alerts when the etcd cluster does not have a leader, indicating potential leadership election issues.	critical

Resources Alerts

Group Name	Title	Description	Severity
kubernetes-apps	KubePodCrashLooping	Alerts when a pod is repeatedly restarting due to crashes (CrashLoopBackOff state).	warning
kubernetes-apps	KubePodNotReady	Alerts when a pod remains in a "not ready" state for over 15 minutes.	warning
kubernetes-apps	KubeDeploymentGenerationMismatch	Alerts if a deployment's generation mismatch occurs, suggesting a failed rollback.	warning
kubernetes-apps	KubeDeploymentReplicasMismatch	Alerts if a deployment hasn't scaled to the desired number of replicas within 15 minutes.	warning
kubernetes-apps	KubeDeploymentRolloutStuck	Alerts if a deployment's rollout stalls for more than 15 minutes.	warning
kubernetes-apps	KubeStatefulSetReplicasMismatch	Alerts if a StatefulSet hasn't scaled to the desired number of replicas within 15 minutes.	warning
kubernetes-apps	KubeStatefulSetGenerationMismatch	Alerts if a StatefulSet's generation mismatch occurs, suggesting a failed rollback.	warning

Group Name	Title	Description	Severity
kubernetes-apps	KubeStatefulSetUpdateNotRolledOut	Alerts if a StatefulSet's update hasn't finished rolling out completely.	warning
kubernetes-apps	KubeDaemonSetRolloutStuck	Alerts if a DaemonSet rollout stalls or fails to progress within 15 minutes.	warning
kubernetes-apps	KubeContainerWaiting	Alerts if a container within a pod is in a waiting state for over an hour.	warning
kubernetes-apps	KubeDaemonSetNotScheduled	Alerts if one or more pods in a DaemonSet fail to be scheduled.	warning
kubernetes-apps	KubeDaemonSetMisScheduled	Alerts if one or more pods in a DaemonSet are scheduled on ineligible nodes.	warning
kubernetes-apps	KubeJobNotCompleted	Alerts if a Job takes longer than 12 hours (43200 seconds) to complete.	warning
kubernetes-apps	KubeJobFailed	Alerts if a Job fails to complete (enters failed state).	warning
kubernetes-apps	KubeHpaReplicasMismatch	Alerts if a HorizontalPodAutoscaler (HPA) hasn't scaled to the desired number of replicas within 15 minutes.	warning
kubernetes-apps	KubeHpaMaxedOut	Alerts if a HPA persistently operates at its maximum replica count for over 15 minutes.	warning

Container Alerts

Group Name	Title	Description	Severity
container.highmemoryusage.rules	Container has a high memory usage	Alerts when a container uses more than 80% of its memory limit. Includes details about the pod, container, namespace, etc.	warning
container.highcpuusage.rules	Container has a high CPU utilization rate	Alerts when a container uses more than 80% of its CPU limit. Includes details about the pod, container, namespace, etc.	warning
container.restarted.rules	Container has multiple restarts	Alerts on containers with multiple restarts, usually indicating instability. Includes relevant details about the container.	warning

GPU Alerts

Group Name	Title	Description	Severity
pod.gpu.evicted	Pod Preempted Due To Inactivity	Alerts on GPU-requesting pods evicted due to exceeding the inactivity limit in their PriorityClass. Guides troubleshooting.	warning
pod.gpu.pending	Pending Pods Due To GPU Requirement	Alerts when GPU-requesting pods are stuck in pending because of insufficient resources or scheduling constraints.	warning

Storage Alerts

Group Name	Title	Description	Severity
kubernetes-storage	KubePersistentVolumeFillingUp	Alerts when a PersistentVolume's free space falls below 3%.	critical

Group Name	Title	Description	Severity
kubernetes-storage	KubePersistentVolumeFillingUp	Warns when a PersistentVolume is predicted to fill up within 4 days, and currently has less than 15% space available.	warning
kubernetes-storage	KubePersistentVolumeInodesFillingUp	Alerts when a PersistentVolume's free inodes fall below 3%.	critical
kubernetes-storage	KubePersistentVolumeInodesFillingUp	Warns when a PersistentVolume is predicted to run out of inodes within 4 days, and has less than 15% of its inodes free.	warning
kubernetes-storage	KubePersistentVolumeErrors	Triggers when a PersistentVolume enters a "Failed" or "Pending" state, indicating potential provisioning issues.	critical

Kubelet Alerts

Group Name	Title	Description	Severity
kubernetes-system-kubelet	KubeNodeNotReady	Alerts when a Kubernetes node has been in the "Not Ready" state for more than 15 minutes.	warning
kubernetes-system-kubelet	KubeNodeUnreachable	Alerts when a Kubernetes node becomes unreachable, indicating potential workload rescheduling.	critical
kubernetes-system-kubelet	KubeletTooManyPods	Warns when a Kubelet is approaching its maximum pod capacity (95%).	info
kubernetes-system-kubelet	KubeNodeReadinessFlapping	Alerts when a node's readiness status frequently changes in a short period (more than twice in 15 minutes), suggesting instability.	warning
kubernetes-system-kubelet	KubeletPlegDurationHigh	Alerts when the Kubelet's Pod Lifecycle Event Generator (PLEG) takes a significant time to relist pods (99th percentile duration exceeding 10 seconds).	warning
kubernetes-system-kubelet	KubeletPodStartUpLatencyHigh	Alerts when the time for pods to reach full readiness becomes high (99th percentile exceeding 60 seconds)	warning
kubernetes-system-kubelet	KubeletClientCertificateExpiration	Warns when a Kubelet's client certificate is about to expire within a week.	warning
kubernetes-system-kubelet	KubeletClientCertificateExpiration	Alerts critically when a Kubelet's client certificate is about to expire within a day.	critical
kubernetes-system-kubelet	KubeletServerCertificateExpiration	Warns when a Kubelet's server certificate is about to expire within a week.	warning
kubernetes-system-kubelet	KubeletServerCertificateExpiration	Alerts critically when a Kubelet's server certificate is about to expire within a day.	critical
kubernetes-system-kubelet	KubeletClientCertificateRenewalErrors	Alerts when a Kubelet encounters repeated errors while attempting to renew its client certificate.	warning
kubernetes-system-kubelet	KubeletServerCertificateRenewalErrors	Alerts when a Kubelet encounters repeated errors while attempting to renew its server certificate.	warning
kubernetes-system-kubelet	KubeletDown	Alerts critically when a Kubelet disappears from Prometheus' target discovery, potentially indicating a serious issue.	critical

Billing Alerts

Group Name	Title	Description
ezbilling.clusterstate.rules	Cluster is in disabled state	Alerts when the EzBilling cluster is disabled. Suggests contacting HPE Support.
ezbilling.upload.rules	Billing usage records not uploaded	Alerts when billing usage records failed to upload for the past 24 hours.
ezbilling.activation.code.grace.peroid.rules	Activation code grace period started	Alerts when the activation code grace period begins, providing the expiration date.

Licensing Alerts

Group Name	Title	Description
ezlicense.license.rules	Cluster is in disabled state	Alerts when the EzLicense cluster enters a disabled state, suggesting the need to contact HPE Support.
ezlicense.expiry.tenday.rules	Activation key expiration	Alerts when an activation key is going to expire within the next 10 days, providing the expiration date.
ezlicense.expiry.thirtyday.rules	Activation key expiration	Alerts when an activation key is going to expire within the next 30 days, providing the expiration date.

Licensing Capacity Alerts

Group Name	Title	Description
ezlicense.capacity.vCPU.rules	Worker node capacity has exceeded vCPU license capacity	Alerts when the vCPU capacity of worker nodes surpasses the available vCPU license limit.
ezlicense.capacity.GPU.rules	Worker node capacity has exceeded GPU license capacity	Alerts when the GPU capacity of worker nodes surpasses the available GPU license limit.
ezlicense.capacity.no.gpu.license.rules	GPU worker node found but no GPU license exists	Alerts when a GPU worker node is detected, but there's no corresponding GPU license available.

Prometheus Alerts

Group Name	Title	Description	Severity
prometheus	PrometheusBadConfig	Alerts when Prometheus fails to reload its configuration.	critical
prometheus	PrometheusSDRefreshFailure	Alerts when Prometheus fails to refresh service discovery (SD) with a specific mechanism.	warning
prometheus	PrometheusNotificationQueueRunningFull	Alerts when the Prometheus alert notification queue is predicted to reach full capacity soon.	warning
prometheus	PrometheusErrorSendingAlertsToSomeAlertmanagers	Alerts when Prometheus encounters a significant error rate (> 1%) sending alerts to a specific Alertmanager.	warning
prometheus	PrometheusNotConnectedToAlertmanagers	Alerts when Prometheus is not connected to any configured Alertmanagers.	warning

Group Name	Title	Description	Severity
prometheus	PrometheusTSDBReloadsFailing	Alerts when Prometheus encounters repeated failures (>0) during the loading of data blocks from disk.	warning
prometheus	PrometheusTSDBCompactionsFailing	Alerts when Prometheus encounters repeated failures (>0) during block compactions.	warning
prometheus	PrometheusNotIngestingSamples	Alerts when a Prometheus instance stops ingesting new metric samples.	warning
prometheus	PrometheusDuplicateTimestamps	Alerts when Prometheus reports samples being dropped due to duplicate timestamps.	warning
prometheus	PrometheusOutOfOrderTimestamps	Alerts when Prometheus reports samples being dropped due to arriving out of order.	warning
prometheus	PrometheusRemoteStorageFailures	Alerts when Prometheus encounters a significant error rate (> 1%) when sending samples to configured remote storage.	critical
prometheus	PrometheusRemoteWriteBehind	Alerts when Prometheus remote write operations fall behind significantly (> 2 minutes).	critical
prometheus	PrometheusRemoteWriteDesiredShards	Alerts when the desired number of shards calculated for remote write exceeds the configured maximum.	warning
prometheus	PrometheusRuleFailures	Alerts when Prometheus encounters repeated failures during rule evaluations.	critical
prometheus	PrometheusMissingRuleEvaluations	Alerts when Prometheus misses rule group evaluations due to exceeding the allowed evaluation time.	warning
prometheus	PrometheusTargetLimitHit	Alerts when Prometheus drops targets because the number of targets exceeds a configured limit.	warning
prometheus	PrometheusLabelLimitHit	Alerts if Prometheus drops targets due to exceeding configured limits on label counts or label lengths.	warning
prometheus	PrometheusScrapeBodySizeLimitHit	Alerts if Prometheus fails scrapes due to targets exceeding the configured maximum scrape body size.	warning
prometheus	PrometheusScrapeSampleLimitHit	Alerts if Prometheus fails scrapes due to targets exceeding the configured maximum sample count.	warning
prometheus	PrometheusTargetSyncFailure	Alerts when Prometheus is unable to synchronize targets successfully due to configuration errors.	critical
prometheus	PrometheusHighQueryLoad	Alerts when the Prometheus query engine reaches close to full capacity, with less than 20% remaining.	warning
prometheus	PrometheusErrorSendingAlertsToAnyAlertmanager	Alerts when there's a persistent error rate (> 3%) while sending alerts from Prometheus to any configured Alertmanager.	critical

Airflow Alerts

Group Name	Title	Description
airflow.scheduler.healthy.rules	Airflow Scheduler Unresponsive	Airflow Scheduler is not responding to health checks.
airflow.dag.import.rules	Airflow DAG Import Errors	Errors detected during import of DAGs from the Git repository.
airflow.tasks.queued.rules	Airflow Tasks Queued and Not Running	Airflow tasks are queued and unable to be executed.
airflow.tasks.starving.rules	Airflow Tasks Starving for Resources	Airflow tasks cannot be scheduled due to lack of available resources in the pool.
airflow.dags.gitrepo.rules	Airflow DAG Git Repository Inaccessible	Airflow cannot access the Git repository containing DAGs.

Kubeflow Alerts

Group Name	Title	Description
kubeflow.katib.rules	Kubeflow katib stuck	Indicates a potential issue with Katib where it's not starting new experiments, trials, or successfully completing trials. Suggests restarting the Katib controller.

MLflow Alerts

Group Name	Title	Description
mlflow_http_request_total	High MLflow HTTP Request Rate without status 200	Alerts if more than 5% of HTTP requests to the MLflow server over a 5-minute window fail (don't have a status code of 200).
mlflow_http_request_duration_seconds_bucket	A histogram representation of the duration of the incoming HTTP requests	Alerts when the 95th percentile of MLflow HTTP request durations exceeds 5 seconds within a 5-minute window, indicating potential slowdowns.
mlflow_http_request_duration_seconds_sum	Total duration in seconds of all incoming HTTP requests	Alerts if the total time spent handling all MLflow HTTP requests exceeds 600 seconds over a 5-minute period, suggesting overload.

Ray Alerts

Group Name	Title	Description	Severity
ray.object.store.memory.high.pressure.alert	Ray: High Pressure on Object Store Memory	Alerts when 90% of Ray object store memory is used consistently for 5 minutes.	warning
ray.node.memory.high.pressure.alert	Ray: High Memory Pressure on Ray Nodes	Alerts when a Ray node's memory usage exceeds 90% of its capacity for 5 minutes.	warning
ray.node.cpu.utilization.high.pressure.alert	Ray: High CPU Pressure on Ray Nodes	Alerts when CPU utilization across Ray nodes exceeds 95% for 5 minutes.	warning
ray.autoscaler.failed.node.creation.alert	Ray: Autoscaler Failed to Create Nodes	Alerts when the Ray autoscaler has failed attempts at creating new nodes for 5 minutes.	warning
ray.scheduler.failed.worker.startup.alert	Ray: Scheduler Failed Worker Startup	Alerts when the Ray scheduler encounters failures during worker startup for 5 minutes.	warning
ray.node.low.disk.space.alert	Ray: Low Disk Space on Nodes	Alerts when a Ray node has less than 10% of disk space free for 5 minutes.	warning

Group Name	Title	Description	Severity
ray.node.network.high.usage.alert	Ray: High Network Usage on Ray Nodes	Alerts when network usage (receive + transmit) on Ray nodes exceeds a threshold for 5 minutes, indicating potential congestion.	warning

Spark Alerts

Group Name	Title	Description	Severity
spark.app.high.failed.rule	Spark Operator: High Failed App Count	Alerts when the number of failed Spark applications handled by the operator surpasses a threshold within a 5-minute window.	warning
spark.app.high.latency.rule	Spark Operator: High Average Latency for App Starting	Alerts when the average latency (time to start) for Spark applications exceeds 120 seconds for a 5-minute period.	warning
spark.app.submission.high.failed.percentage.rule	Spark Operator: High Percentage of Failed Spark App Submissions	Alerts when the failure rate of Spark application submissions exceeds 10% of total submissions for 15 minutes.	warning
spark.app.low.success.rate.rule	Spark Operator: Low Success Rate of Spark Applications	Alerts when the success rate of Spark applications drops below 80% of total submissions for a 20-minute period.	warning
spark.app.executor.low.success.rate.rule	Spark Operator: Low Success Rate of Spark Application Executors	Alerts when the success rate of Spark executors drops below 90% of total executors for a 20-minute period.	warning
spark.workload.high.memory.pressure.rule	Spark Workload: High Memory Pressure	Alerts when overall memory pressure on Spark's BlockManager exceeds a critical threshold for 1 minute.	warning
spark.workload.high.heap.memory.pressure.rule	Spark Workload: High On-Heap Memory Pressure	Alerts when on-heap memory pressure on Spark's BlockManager exceeds a critical threshold for 1 minute.	warning
spark.workload.high.cpu.usage.rule	Spark Workload: High JVM CPU Usage	Alerts when JVM CPU usage within Spark exceeds a critical threshold for 5 minutes.	warning

Superset Alerts

Group Name	Title	Description	Severity
superset.http.request.duration	A histogram representation of the duration of the incoming HTTP requests	Alerts when the 99th percentile of HTTP request duration exceeds 3 seconds for 5 minutes, indicating slow responses.	critical
superset.http.request.total	Superset total number of HTTP requests without status 200	Alerts if more than 5% of HTTP requests to Superset within a 5-minute window fail (don't have a status code of 200).	critical
superset.http.request.exceptions.total	Total number of HTTP requests which resulted in an exception	Alerts when more than 10 HTTP requests to Superset result in exceptions within a 5-minute window.	critical
superset.gc.objects.collected	Objects collected during gc	Alerts if more than 100 objects (generation 0) are collected during garbage collection within a 5-minute window.	warning
superset.gc.objects.uncollectable	Uncollectable objects found during GC	Alerts if more than 50 uncollectable objects (generation 0) are found during garbage collection within a 5-minute window.	warning

Group Name	Title	Description	Severity
supercet.gc.collections	Number of times this generation was collected	Alerts if the youngest generation (0) of garbage collection has run more than 100 times within a 5-minute window, indicating potential memory pressure.	warning

Logging

Describes how logging works in HPE Ezmeral Unified Analytics Software and how to access log files for the platform and applications.

Logging is crucial for monitoring and troubleshooting applications and the cluster infrastructure. Logs capture data generated by applications, containers, and Kubernetes components running in a Kubernetes cluster.

In HPE Ezmeral Unified Analytics Software, you can easily access log files to monitor and troubleshoot issues.

Log Rotation

To prevent storage issues, logs are automatically rotated to ensure that the `shared`[1] directory does not exceed the limit of 10 MB per file. When the file size surpasses 10 MB, the old copy is retained, and a new log file is created, effectively managing storage and keeping it under control.

NOTE: Only one old copy of 10 MB is retained at any time.

For example, the `airflow-ui.log` undergoes renaming as `airflow-ui.log.1` when it exceeds 10 MB in size. Simultaneously, a new log file named `airflow-ui.log` is created. Similarly, if the size of the new `airflow-ui.log` file exceeds the 10 MB threshold, the current `airflow-ui.log.1` log file is replaced with the new logs from the `airflow-ui.log` file. This log rotation process ensures efficient management of log files while maintaining the specified size limit.

[1] The `shared` directory is persistent volume storage shared by all users.

Accessing Log Files

To access the log files:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, go to **Data Engineering > Data Sources**.
3. Select the **Data Volumes** tab.
4. On the **Data Volumes** tab, select the **logs/** folder. Under the **logs/** folder is a folder named for the installation. The **logs/<installation-name>/** folder contains the following subdirectories with logging data:

<table>
<tr><td><b>apps/</b></td><td>Contains platform and application component logs. This directory contains the following subdirectories:
<table>
<tr><td><b>app-core/</b></td><td>Contains logs for core application pods.</td></tr>
<tr><td><b>app-user/</b></td><td>Contains logs for user-initiated pods such as notebooks, inference jobs, Spark jobs, and others.</td></tr>
</table>
</td></tr>
</table>

	platform/	Contains infrastructure pod logs.
audit/	Contains the native Kubernetes audit logs. The native Kubernetes audit logs record requests made to the Kubernetes API server, such as which users or services requested access to cluster resources and why the requests were authorized or rejected.	
system/	Contains system node service logs.	

TIP: The **Actions** column provides options that you can use on directories and log files, for example:

- Open, rename, or delete a folder or directory
- Rename, download, or delete log files

Audit Logging

Describes auditing in HPE Ezmeral Unified Analytics Software and how to access audit logs.

Auditing provides a chronological set of records that document the events that occur in an HPE Ezmeral Unified Analytics Software cluster.

Auditing records user, application, and control plane events (that occur through the UI and programmatic access via APIs or CLIs) in audit logs. Audit logs maintain records of actions for accountability, tracking, and compliance purposes.

Auditing provides the following information about actions in HPE Ezmeral Unified Analytics Software:

- Type of action
- User or application that triggered the action
- Timestamp (time the action occurred)
- Status of the action (Failed, Started, Success)

Audited Actions

The following tables lists the actions that auditing captures in the audit logs:

Area	Description
Platform	Captures successful and failed login attempts by users.
Administration	Captures the add/delete/modify user actions performed by a user assigned the administrator role.

Area	Description
Billing & Licensing	Captures the license related actions performed by the platform administrator.
	Captures the billing and activation related actions performed by the platform administrator, including: • Creation of billing credentials and signing-key • Creation billing and license credentials and signing-key in airgapped environments • Downloading of metering usage in airgapped environments • Uploading of metering usage • Renewal of billing and license credentials
Keycloak	Captures Keycloak realm updates when the product is deactivated or activated (triggered from enabled to disabled and vice versa).
Kubeflow	Captures the creation of a notebook in Kubeflow. The audit message contains information about the name/namespace and whether the API call was a dry run.
	Captures the deletion of a notebook in Kubeflow. The audit message contains information about the name/namespace and whether the API call was a dry run.
	Captures the creation of a Create KServe Inference Service in Kubeflow. The audit message contains information about the name/namespace and whether the API call was a dry run.
	Captures the deletion of a Create KServe Inference Service in Kubeflow. The audit message contains information about the name/namespace and whether the API call was a dry run.
Spark	Spark application submitted using Spark operator.
	Spark application deleted using Spark operator.
	Scheduled Spark application submitted using Spark operator.
	Scheduled spark application deleted using Spark operator.
	NOTE: Livy doesn't support audit logging.

Area	Description
Airflow	User disabled or enabled a DAG. Captures DAG ID and username.
	User started DAG execution. Captures DAG execution time, DAG ID, and username.
	DAG task scheduled after triggering the DAG. Captures DAG run ID, DAG ID, and task ID.
	DAG task running after scheduling. Captures DAG execution time, DAG ID, task ID, and username.
	DAG task succeeded after running. Captures DAG execution time, DAG ID, task ID, and username.
	DAG task failed after running. Captures DAG execution time, DAG ID, task ID, and username.
EzPresto	Query completed event. Audits the user, query, timestamp, status, type of query, and client-ip.
	Audits the data source name, data source type, user, timestamp, and status.
	Audits the user, create view query, timestamp, and status.
	Audtis the user, cache table details, remote table details, and status.

Accessing Audit Logs

Administrators can access audit logs by signing in to HPE Ezmeral Unified Analytics Software UI and selecting **Administration > Audit Logs** in the left navigation bar. The list of audit logs display on the **Audit Logs** page.

Viewing Audit Logs for a Period of Time

You can view the audit logs for a given time period by clicking into the dropdown field. The dropdown has the following options:

Option	Description
1 hour	See the audit logs recorded during the past hour.
6 hours	See the audit logs recorded during the past six hours.
Today	Today is the current date, starting at 12:00 am. For example, if you select Today and the date is July 19th and the time is 5:00 pm, you will see all the audit logs that were recorded between 12:00am and 5:00pm on July 19th. Date and time is based on local time. If two people are in different time zones, each person will see results based on their respective time zones when they select Today.
Custom	Click the calendar icon and select one or more days. To select multiple days, click the first day and then click the last day. Select the start and end times. For multiple days, the start time is the start time on day one and the end time is the end time on the last day.

Searching Audit Logs

You can search audit logs for records that match specified search criteria. For example, you can search on keywords and tags, including event type, users, date range, and failed attempts.

Filtering Audit Logs

Clicking the filter icon opens the **Filters** drawer where you can select one or more filter options. You can filter by:

- Actions
- Statuses
- Users

Clicking **Reset** clears the filters. Click **Apply** after you click **Reset** to save the update.

Downloading Audit Logs

Clicking **Download Logs** downloads the audit logs for the given time period to an Excel file.

Data Engineering

Data engineers can design and build pipelines that transform and transport data into usable formats for data consumers.

HPE Ezmeral Unified Analytics Software includes connectors for several data sources that facilitate data virtualization by providing a single point of uniform, controlled access to distributed data, regardless of the compute engine. You can use open-source tools, such as Apache Spark and Apache Airflow, to extract data from disparate sources and create transformed data sets for data consumption.

For example, you can run a Spark job to move data from one data source (such as Snowflake) into another data source (such as HPE Ezmeral Data Fabric) and then connect HPE Ezmeral Unified Analytics Software to the HPE Ezmeral Data Fabric data source. Once connected to HPE Ezmeral Data Fabric, you can work with the data (join and transform) to create consumable models for users and applications.

Data consumers with appropriate permissions can use data in their analytical workloads, data science workflows, dashboards, or for data modeling.

Working with Data

The Data Engineering space provides access to interfaces that enable you to use EzPresto on page 294, the SQL query engine in HPE Ezmeral Unified Analytics Software, to work with data.

The following list describes what you can do through each of the interfaces in the Data Engineering space:

Data Sources

Connect HPE Ezmeral Unified Analytics Software to external data sources. Each connected data source displays as a tile on the screen. You can also remove data sources or access the Query Editor from each data source tile. See Connecting Data Sources on page 297.

When you connect HPE Ezmeral Unified Analytics Software to various data sources, you can access the data in those data sources from Superset on page 364 and then visualize the data.

Data Catalog

Select data sets (tables and views) from one or more data sources and run federated queries. You can also cache data sets. Caching stores the data in

	a distributed caching layer within the data fabric for accelerated access to the data. See Caching Data on page 351.
Query Editor	Run queries against the selected data sets. You can also create views and new schemas.
Cached Assets	Lists the cached data sets (tables and views). See Caching Data on page 351.
Airflow Pipelines	Links to the Airflow interface where you can connect to data sets created in HPE Ezmeral Unified Analytics Software and use them in your data pipelines. See Airflow on page 353.

More information

Get Started on page 6
Describes how to get started with HPE Ezmeral Unified Analytics Software.

Accessing Data in External S3 Object Stores

Describes how to access data in external object stores from clients such as Spark and Kubeflow notebooks.

After an administrator connects HPE Ezmeral Unified Analytics Software to an external object store in AWS, MinIO, or HPE Ezmeral Data Fabric Object Store, you can access data in those data sources through clients such as Spark or Kubeflow notebooks, without providing an access key or secret key. Your HPE Ezmeral Unified Analytics Software administrator provides the access credentials when creating the data source connection. Your access to the data source is authorized through HPE Ezmeral Unified Analytics Software.

To connect a client to an object store, you provide the client with the following information:

- Data source name
- Endpoint URL
- Bucket that you want the client to access

You can find the data source name and endpoint URL on the data source tile in the HPE Ezmeral Unified Analytics Software UI.

Once connected, clients can:

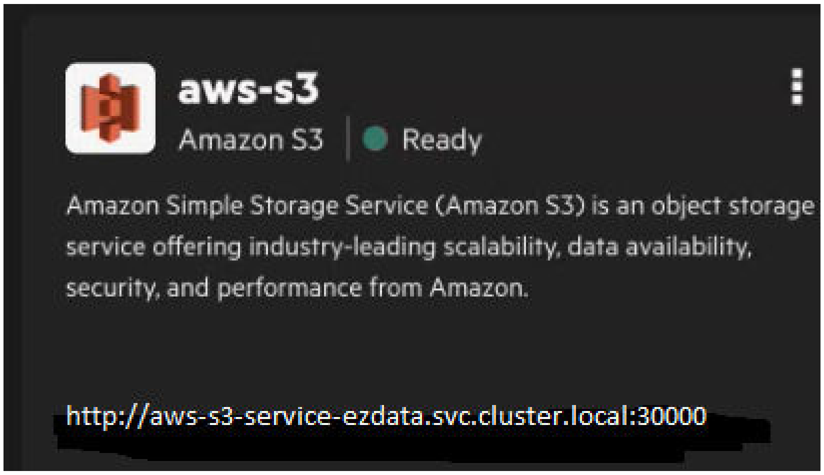
- Read and download files in a bucket
- Upload files from a bucket
- Create buckets

Getting the Data Source Name and S3 Proxy Endpoint URL

To get the data source name and S3 proxy endpoint URL:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Data Engineering > Data Sources**.
3. On the **Data Sources** page, find the tile for the object store that you want to connect to.

   The following image shows an example of a tile for an AWS S3 data source with the name aws-s3 and the enpoint URL:



aws-s3

Amazon S3 | Ready

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance from Amazon.

http://aws-s3-service-ezdata.svc.cluster.local:30000

[redacted]

NOTE: By default, a local-s3 Ezmeral Data Fabric tile also displays. This Ezmeral Data Fabric version of S3 is a local S3 version used internally by HPE Ezmeral Unified Analytics Software. Do not connect to this data source.

4. Note the *data source name* and *endpoint URL* and then use them to configure the client.

Using KServe to Deploy a Model on S3 Object Storage

Describes how to deploy a KServe model on S3 object storage from a Kubeflow notebook.

You can deploy a KServe model on the S3 object storage that your administrator connected to HPE Ezmeral Unified Analytics Software.

Add YAML configurations that perform the following actions and then run the code from a Kubeflow notebook:

- Create a service account
- Create a secret

  IMPORTANT:

  - When you create the secret, note that the `{os.environ['AUTH_TOKEN']}` option assigns a value to the `AWS_ACCESS_KEY_ID`. The value assigned is the value on the JWT for the current notebook user.
  - When accessing object store data via the S3 proxy in Unified Analytics, enter `"s3"` as the AWS_SECRET_ACCESS_KEY value. For additional information about the S3 proxy, see Configuring a Spark Application to Access External S3 Object Storage on page 383.

- Deploy a model (`InferenceService`)
- Apply the YAML (`!kubectl apply -f {yaml_name}`)

The following example shows a YAML configuration:

```
best_model_uri =    '<path_to_the_model>'  # for example 's3://
mlflow/2/0e4508d276a0427cb67da7630acb2e14/artifacts/model'
secret_name = 's3-proxy-kserve-secret'
sa_name = 's3-proxy-kserve-sa'
inference_service_name = "service-name"
yaml_name = './s3-proxy-kserve.yaml'
```

```
############################################

with open(yaml_name, 'w') as file:
    text = f"""---
apiVersion: v1
kind: Secret
metadata:
  name: "{secret_name}"
  annotations:
    serving.kserve.io/s3-cabundle: ""
    serving.kserve.io/s3-endpoint:
"local-s3-service.ezdata-system.svc.cluster.local:30000/"
    serving.kserve.io/s3-useanoncredential: "false"
    serving.kserve.io/s3-usehttps: "0"
    serving.kserve.io/s3-verifyssl: "0"
stringData:
  AWS_ACCESS_KEY_ID: "{os.environ['AUTH_TOKEN']}"
  AWS_SECRET_ACCESS_KEY: "s3"
type: Opaque

---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: "{sa_name}"
secrets:
  - name: "{secret_name}"

---
apiVersion: "serving.kserve.io/v1beta1"
kind: "InferenceService"
metadata:
  name: "{inference_service_name}"
spec:
  predictor:
    serviceAccountName: "{sa_name}"
    sklearn:
      protocolVersion: "v2"
      storageUri: "{best_model_uri}"
"""
    file.write(text)

############################################
!kubectl apply -f {yaml_name}
```

EzPresto

Describes the EzPresto SQL query engine and its featues.

EzPresto in HPE Ezmeral Unified Analytics Software

EzPresto is an SQL query engine based on the open-source, Linux foundation multi-parallel processing (MPP) query engine PrestoDB, that is optimized to run federated queries across various data sources. Enterprise BI applications such as Tableau, Power BI, and data processing engines, such as Spark, can leverage EzPresto for rapid query performance and prompt insights through federated data access.

You can easily connect EzPresto to multiple types of data sources from the Data Engineering space in HPE Ezmeral Unified Analytics Software by going to **Data Engineering > Data Sources**. Connections require a JDBC connection URL and user credentials.

Data sets available to the connected user display in the Data Catalog, which is accessible by going to **Data Engineering > Data Catalog**. In the Data Catalog, you select the data sets you want to work with. You can query or cache the selected datasets.

When you opt to cache data sets, you can modify the data sets prior to caching them. For example, you can edit table and column names, remove columns, and create new schema. Cached data sets (tables and views) are accessible in the Cached Assets space of HPE Ezmeral Unified Analytics Software. You can access cached assets by going to **Data Engineering > Cached Assets**.

When you opt to query data sets, you can run federated queries (query across data sets in multiple data sources) from the Query Editor. You can access the Query Editor by going to **Data Engineering > Query Editor**. Querying cached data sets accelerates queries for significant performance gains.

You can access the data in connected data sources from Superset and visualize the data that results from complex, federated queries. Superset is accessible in HPE Ezmeral Unified Analytics Software by going to **BI Reporting > Dashboards** or **Tools & Frameworks > Data Engineering** tab and clicking **Open** in the Superset tile. See Superset on page 364. You can also monitor the state of queries and query details, including the query plan and resource usage, by going to **Administration > EzSQL Cluster Monitoring**.

Refer to the following tutorials to get started with EzPresto in HPE Ezmeral Unified Analytics Software:

- Data Source Connectivity and Exploration on page 8
- BI Reporting (Superset) Basics on page 14
- Retail Store Analysis Dashboard (Superset) on page 43

EzPresto Key Features

EzPresto provides the following key benefits and features:

Data Source Connectivity

EzPresto includes connectors for several data sources, including:

- HPE Ezmeral Data Fabric
- HDFS
- Data Lakes
- Hive Metastore (including managed HMS services such as AWS Glue)
- Object Stores
- Relational Databases
- NoSQL Databases
- Streaming data platforms
- Data warehouses

Built-In Data Catalog

The built-in data catalog provides dynamic registration of new data sources. Data administrators can add new data sources as they become available without restarting any services. When a data administrator adds a new data source, the data catalog automatically refreshes so users, such as data analysts, can browse the new datasets and perform upstream activities, such as reporting and dashboarding.

Role-Based Access Controls

Role-based access controls isolate queries such that members (non-admin users) can only view, access, and cancel their own queries. Admin users have full access to all queries. For example, if a member runs a query that takes too long to complete or uses too many resources, any admin in HPE Ezmeral Unified Analytics Software can stop the query.

Optimized Federated Queries

Access data across disparate data sources in a single, optimized query. Query optimizations for accelerated performance include:

- Predicate pushdown - EzPresto pushes filters in the WHERE clause down to the data source for processing to reduce the number of rows returned.
- Projection pushdown - EzPresto pushes projects (scanning of selected columns) down to the data source for processing to reduce the amount of data returned.
- Dynamic filtering - EzPresto evaluates predicates on the right side of a join and pushes them to the left side of the join to reduce the number of rows scanned from the left table.
- Cost-based optimization - EzPresto uses table statistics to calculate the cost (resource usage) of various query plans and chooses the optimal plan (plan that uses the least resources) to run the query.

Distributed Caching

EzPresto accelerates federated queries through distributed caching of commonly used datasets. EzPresto currently supports *explicit caching* where you manually modify tables and select the data that you want cached for fast query access. You can use explicitly cached data for data modeling. EzPresto stores cached data in a data fabric volume. The cache expires based on the set TTL (time-to-live). See Connecting Data Sources on page 297 and Caching Data on page 351.

Explicit Caching

You manually modify tables and select the data in the tables that you want stored in the cache for fast query access. You can use explicitly cached data for data modeling. You can set a TTL (time to live) for the cache.

Self-Service Data Access

End-users can browse data sets they have access to and select the relevant data for their queries and analytical applications and workloads.

Run-Anywhere Architecture

EzPresto has a run-anywhere architecture; you can run EzPresto on-premises, on edge, in the cloud, or hybrid environments.

EzPresto Architecture

The EzPresto architecture consists of the following main components:

Presto

EzPresto uses a modified version of Presto as the query engine. Most of the modifications are in the query planning and optimizer areas, as well as support

	for different data sources, such as Teradata and Snowflake, and in-process caching based on Apache Geode, tuned for OLTP and OLAP access. The cache provides a tuple store with specialized columnar formats.
WebService	Provides the API.
Web UI	Provides the ability to access EzPresto in applications.
Client Connections	Provides the ability to connect to BI tools and external data sources via the JDBC client.
KeyCloak	KeyCloak provides the authentication mechanism and different authentication options, such as LDAP and JWT.

More information

Data Source Connectivity and Exploration on page 8
Provides basic steps for using the Data Engineering space within HPE Ezmeral Unified Analytics Software.

Connecting Data Sources

Provides instructions for connecting HPE Ezmeral Unified Analytics Software to external data sources.

Connecting data sources enables federated access to data for users with the appropriate permissions. HPE Ezmeral Unified Analytics Software includes PrestoDB and CSI connectors, enabling connections to multiple types of data sources. Connecting to an external data source is as simple as selecting the data source type and providing the required connection parameters and credentials.

IMPORTANT:

- Only HPE Ezmeral Unified Analytics Software administrators can create data source connections.
- Each data source connection that you create must have a unique name. For example, you can create multiple Hive data source connections, but each connection created must have a different name.
- EzPresto does not support underscores (_) in data source names. For example, hive_one is not supported; instead, use something like hiveone.
- Access to data in a data source is based on the username and password supplied when creating the data source connection. Data sources are accessible to all users with permission once they are connected.

Complete the following steps to connect a data source:

1. In the left navigation pane, select **Data Engineering > Data Sources**.
2. Select the tab that correlates with the type of data source that you want to connect:
   - **Structured Data** (relational databases, such as MySQL and Hive)
   - **Object Store Data** (S3 object stores, such as AWS S3 and MinIO)
   - **Data Volumes** (mount volumes in file storage, such as HPE Ezmeral Data Fabric File Store and HPE GreenLake for File Storage)
3. Complete the steps for the data type selected:

Structured Data	**a.** On the **Structured Data** tab, click **Add New Data Source**.

b. Locate the tile with the type of data source that you want to connect, and click **Create Connection**. For example, if you want to connect to a Hive data source, locate the Hive tile and click **Create Connection** in the Hive tile.

c. In the drawer that opens, enter the connection parameters and then click **Connect**.

> **TIP:** For every data source that you connect, you have the option to select the **Enable Local Snapshot Table** option. This option caches remote table data to accelerate queries on the tables. The cache is active for the duration of the configured TTL (time-to-live) or until the remote tables in the data source are altered.

Object Store Data

a. On the **Object Store Data** tab, click **Add New Data Source**.

b. Locate the tile with the type of data source that you want to connect, and click **Add <data-source>**. For example, if you want to connect to an Amazon S3 data source, locate the Amazon S3 tile and click **Add Amazon S3** in the tile.

c. In the drawer that opens, enter the connection parameters and then click **Add**.

Data Volumes

a. On the **Data Volumes** tab, click **New Volume**.

b. Locate the tile with the type of data source that you want to connect, and click **Add <data-source>**. For example, if you want to connect to an HPE GreenLake for File Storage data source, locate the HPE GreenLake for File Storage tile and click **Add HPE GreenLake for File Storage** in the tile.

c. In the drawer that opens, enter the connection parameters and then click **Add**.

Delta Connection Parameters

List of Delta connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Delta connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

NOTE: Delta connector values varies based on type of metastore. Seehttps://prestodb.io/docs/current/connector/deltalake.html.

Parameter	Description	Default Value	Data Type
Hive Metastore	The type of Hive metastore to use	thrift	STRING

Parameter	Description	Default Value	Data Type
Enable Local Snapshot Table	Enable Caching while querying	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Delta Parquet Dereference Pushdown Enabled	Enable pushing nested column dereferences into table scan so that only the required fields selected in a struct data type column are selected	true	BOOLEAN
Delta Max Splits Batch Size	Delta : Max split batch size	200	INTEGER
Delta Max Partitions Per Writer	Delta : Maximum number of partitions per writer	100	INTEGER
Hive Metastore	The type of Hive metastore to use	thrift	STRING
Hive Insert Overwrite Immutable Partitions Enabled	When enabled, insertion query will overwrite existing partitions when partitions are immutable. This config only takes effect with hive.immutable-partitions set to true	false	BOOLEAN
Hive Create Empty Bucket Files For Temporary Table	Create empty files when there is no data for temporary table buckets	false	BOOLEAN
Hive Enable Parquet Batch Reader Verification	Enable optimized parquet reader	false	BOOLEAN
Hive Create Empty Bucket Files For Temporary Table	Create empty files when there is no data for temporary table buckets	false	BOOLEAN
Hive Min Bucket Count To Not Ignore Table Bucketing	Ignore table bucketing when table bucket count is less than the value specified, otherwise, it is controlled by property hive.ignore-table-bucketing	0	INTEGER
Hive Partition Statistics Based Optimization Enabled	Enables partition statistics based optimization, including partition pruning and predicate stripping	false	BOOLEAN
Hive Experimental Optimized Partition Update Serialization Enabled	Serialize PartitionUpdate objects using binary SMILE encoding and compress with the ZSTD compression	false	BOOLEAN
Hive Materialized View Missing Partitions Threshold	Materialized views with missing partitions more than this threshold falls back to the base tables at read time	100	INTEGER
Hive S3select Pushdown Max Connections	The maximum number of client connections allowed for those operations from worker nodes	500	INTEGER
Hive Temporary Staging Directory Enabled	Should use (if possible) temporary staging directory for write operations	true	BOOLEAN
Hive Temporary Staging Directory Path	Location of temporary staging directory for write operations. Use ${USER} placeholder to use different location for each user.	/tmp/presto-${USER}	STRING
Hive Temporary Table Storage Format	The default file format used when creating new tables.	ORC	STRING
Hive Temporary Table Compression Codec	The compression codec to use when writing files for temporary tables	SNAPPY	STRING
Hive Use Pagefile For Hive Unsupported Type	Automatically switch to PAGEFILE format for materialized exchange when encountering unsupported types	true	BOOLEAN

Parameter	Description	Default Value	Data Type
Hive Parquet Pushdown Filter Enabled	Enable complex filter pushdown for Parquet	false	BOOLEAN
Hive Range Filters On Subscripts Enabled	Enable pushdown of range filters on subscripts (a[2] = 5) into ORC column readers	false	BOOLEAN
Hive Adaptive Filter Reordering Enabled	Enable adaptive filter reordering	true	BOOLEAN
Hive Parquet Batch Read Optimization Enabled	Is Parquet batch read optimization enabled	false	BOOLEAN
Hive Enable Parquet Dereference Pushdown	Is dereference pushdown expression pushdown into Parquet reader enabled	false	BOOLEAN
Hive Max Metadata Updater Threads	Maximum number of metadata updated threads	100	INTEGER
Hive Partial_aggregation_pushdown_enabled	Enable partial aggregation pushdown	false	BOOLEAN
Hive Manifest Verification Enabled	Enable verification of file names and sizes in manifest / partition parameters	false	BOOLEAN
Hive Undo Metastore Operations Enabled	Enable undo metastore operations	true	BOOLEAN
Hive Verbose Runtime Stats Enabled	Enable tracking all runtime stats. Note that this may affect query performance	false	BOOLEAN
Hive Prefer Manifests To List Files	Prefer to fetch the list of file names and sizes from manifests rather than storage	false	BOOLEAN
Hive Partition Lease Duration	Partition lease duration	0.00s	DURATION
Hive Size Based Split Weights Enabled	Enable estimating split weights based on size in bytes	true	BOOLEAN
Hive Minimum Assigned Split Weight	Minimum weight that a split can be assigned when size based split weights are enabled	0.05	DOUBLE
Hive Use Record Page Source For Custom Split	Use record page source for custom split. By default, true. Used to query MOR tables in Hudi.	true	BOOLEAN
Hive Split Loader Concurrency	Number of maximum concurrent threads per split source	4	INTEGER
Hive Domain Compaction Threshold	Maximum ranges to allow in a tuple domain without compacting it	100	INTEGER
Hive Max Concurrent File Renames	Maximum concurrent file renames	20	INTEGER
Hive Max Concurrent Zero Row File Creations	Maximum number of zero row file creations	20	INTEGER
Hive Recursive Directories	Enable reading data from subdirectories of table or partition locations. If disabled, subdirectories are ignored.	false	BOOLEAN
Hive User Defined Type Encoding Enabled	Enable user defined type	false	BOOLEAN
Hive Loose Memory Accounting Enabled	When enabled relaxes memory accounting for queries violating memory limits to run that previously honored memory thresholds	false	BOOLEAN

Parameter	Description	Default Value	Data Type
Hive Max Outstanding Splits Size	Maximum amount of memory allowed for split buffering for each table scan in a query, before the query is failed	256MB	DATASIZE
Hive Max Split Iterator Threads	Maximum number of iterator threads	1000	INTEGER
Hive Allow Corrupt Writes For Testing	Allow Hive connector to write data even when data will likely be corrupt	false	BOOLEAN
Hive Create Empty Bucket Files	Should empty files be created for buckets that have no data?	true	BOOLEAN
Hive Max Partitions Per Writers	Maximum number of partitions per writer	100	INTEGER
Hive Write Validation Threads	Number of threads used for verifying data after a write	16	INTEGER
Hive Orc Tiny Stripe Threshold	ORC: Threshold below which an ORC stripe or file will read in its entirety	8MB	DATASIZE
Hive Orc Lazy Read Small Ranges	ORC read small disk ranges lazily	true	BOOLEAN
Hive Orc Bloom Filters Enabled	ORC: Enable bloom filters for predicate pushdown	false	BOOLEAN
Hive Orc Default Bloom Filter Fpp	ORC Bloom filter false positive probability	0.05	DOUBLE
Hive Orc Optimized Writer Enabled	Experimental: ORC: Enable optimized writer	true	BOOLEAN
Hive Orc Writer Validation Percentage	Percentage of ORC files to validate after write by re-reading the whole file	0.0	DOUBLE
Hive Orc Writer Validation Mode	Level of detail in ORC validation. Lower levels require more memory	BOTH	STRING
Hive Rcfile Optimized Writer Enabled	Experimental: RCFile: Enable optimized writer	true	BOOLEAN
Hive Assume Canonical Partition Keys	Assume canonical parition keys?	false	BOOLEAN
Hive Parquet Fail On Corrupted Statistics	Fail when scanning Parquet files with corrupted statistics	true	BOOLEAN
Hive Parquet Max Read Block Size	Parquet: Maximum size of a block to read	16MB	DATASIZE
Hive Optimize Mismatched Bucket Count	Enable optimization to avoid shuffle when bucket count is compatible but not the same	false	BOOLEAN
Hive Zstd Jni Decompression Enabled	Use JNI based zstd decompression for reading ORC files	false	BOOLEAN
Hive File Status Cache Size	Hive file status cache size	0	LONG
Hive File Status Cache Expire Time	Hive file status cache : expiry time	0.00s	DURATION
Hive Per Transaction Metastore Cache Maximum Size	Maximum number of metastore data objects in the Hive metastore cache per transaction	1000	INTEGER

Parameter	Description	Default Value	Data Type
Hive Metastore Refresh Interval	Asynchronously refresh cached metastore data after access if it is older than this but is not yet expired, allowing subsequent accesses to see fresh data.	0.00s	DURATION
Hive Metastore Cache Maximum Size	Maximum number of metastore data objects in the Hive metastore cache	10000	INTEGER
Hive Metastore Refresh Max Threads	Maximum threads used to refresh cached metastore data	100	INTEGER
Hive Partition Versioning Enabled		false	BOOLEAN
Hive Metastore Impersonation Enabled	Should Presto user be impersonated when communicating with Hive Metastore	false	BOOLEAN
Hive Partition Cache Validation Percentage	Percentage of partition cache validation	0.0	DOUBLE
Hive Metastore Thrift Client Socks Proxy	Metastore thrift client socks proxy	null	STRING
Hive Metastore Timeout	Timeout for Hive metastore requests	10.00s	DURATION
Hive Dfs Verify Checksum	Verify checksum for data consistency	true	BOOLEAN
Hive Metastore Cache Ttl	Duration how long cached metastore data should be considered valid	0.00s	DURATION
Hive Metastore Recording Path	Metastore recording path	null	STRING
Hive Replay Metastore Recording	Replay metastore recording	false	BOOLEAN
Hive Metastore Recoding Duration	Metastore recording duration	0.00m	DURATION
Hive Dfs Require Hadoop Native	Hadoop native is required?	true	BOOLEAN
Hive Metastore Cache Scope	Metastore cache scope	ALL	STRING
Hive Metastore Authentication Type	Hive metastore authentication type.	NONE	STRING
Hive Hdfs Authentication Type	HDFS authentication type.	NONE	STRING
Hive Hdfs Impersonation Enabled	Should Presto user be impersonated when communicating with HDFS	false	BOOLEAN
Hive Hdfs Wire Encryption Enabled	Should be turned on when HDFS wire encryption is enabled	false	BOOLEAN
Hive Skip Target Cleanup On Rollback	Skip deletion of target directories when a metastore operation fails and the write mode is DIRECT_TO_TARGET_NEW_DIRECTORY	false	BOOLEAN
Hive Bucket Execution	Enable bucket-aware execution: only use a single worker per bucket	true	BOOLEAN
Hive Bucket Function Type For Exchange	Hash function type for exchange	HIVE_COMPATIBLE	STRING
Hive Ignore Unreadable Partition	Ignore unreadable partitions and report as warnings instead of failing the query	false	BOOLEAN
Hive Max Buckets For Grouped Execution	Maximum number of buckets to run with grouped execution	1000000	INTEGER

Parameter	Description	Default Value	Data Type
Hive Sorted Write To Temp Path Enabled	Enable writing temp files to temp path when writing to bucketed sorted tables	false	BOOLEAN
Hive Sorted Write Temp Path Subdirectory Count	Number of directories per partition for temp files generated by writing sorted table	10	INTEGER
Hive Fs Cache Max Size	Hadoop FileSystem cache size	1000	INTEGER
Hive Non Managed Table Writes Enabled	Enable writes to non-managed (external) tables	false	BOOLEAN
Hive Non Managed Table Creates Enabled	Enable non-managed (external) table creates	true	BOOLEAN
Hive Table Statistics Enabled	Enable use of table statistics	true	BOOLEAN
Hive Partition Statistics Sample Size	Specifies the number of partitions to analyze when computing table statistics.	100	INTEGER
Hive Ignore Corrupted Statistics	Ignore corrupted statistics rather than failing	false	BOOLEAN
Hive Collect Column Statistics On Write	Enables automatic column level statistics collection on write	false	BOOLEAN
Hive S3select Pushdown Enabled	Enable query pushdown to AWS S3 Select service	false	BOOLEAN
Hive Max Initial Splits	Max initial splits	200	INTEGER
Hive Max Initial Split Size	Max initial split size	null	DATASIZE
Hive Writer Sort Buffer Size	Write sort buffer size	64MB	DATASIZE
Hive Node Selection Strategy	Node affinity selection strategy	NO_PREFERENCE	STRING
Hive Max Split Size	Max split size	64MB	DATASIZE
Hive Max Partitions Per Scan	Maximum allowed partitions for a single table scan	100000	INTEGER
Hive Max Outstanding Splits	Target number of buffered splits for each table scan in a query, before the scheduler tries to pause itself	1000	INTEGER
Hive Metastore Partition Batch Size Min	Hive metastore : min batch size for partitions	10	INTEGER
Hive Metastore Partition Batch Size Max	Hive metastore : max batch size for partitions	100	INTEGER
Hive Config Resources	An optional comma-separated list of HDFS configuration files	[]	FILEPATH
Hive Dfs Ipc Ping Interval	The client will send ping when the interval is passed without receiving bytes	10.00s	DURATION
Hive Dfs Timeout	DFS timeout	60.00s	DURATION
Hive Dfs Connect Timeout	DFS connection timeout	500.00ms	DURATION
Hive Dfs Connect Max Retries	DFS - max retries in case of connection issue	5	INTEGER
Hive Storage Format	The default file format used when creating new tables.	ORC	STRING
Hive Compression Codec	The compression codec to use when writing files	GZIP	STRING
Hive Orc Compression Codec	The preferred compression codec to use when writing ORC and DWRF files	GZIP	STRING

Parameter	Description	Default Value	Data Type
Hive Respect Table Format	Should new partitions be written using the existing table format or the default PrestoDB format?	true	BOOLEAN
Hive Immutable Partitions	Can new data be inserted into existing partitions?	false	BOOLEAN
Hive Max Open Sort Files	Maximum number of writer temporary files to read in one pass	50	INTEGER
Hive Dfs Domain Socket Path	This is a path in the filesystem that allows the client and the DataNodes to communicate.	null	STRING
Hive S3 File System Type	s3 file system type	PRESTO	STRING
Hive Gcs Json Key File Path	JSON key file used to access Google Cloud Storage	null	FILEPATH
Hive Gcs Use Access Token	Use client-provided OAuth token to access Google Cloud Storage	false	BOOLEAN
Hive Orc Use Column Names	Access ORC columns using names from the file	false	BOOLEAN
Hive Orc Max Merge Distance	ORC: Maximum size of gap between two reads to merge into a single read	1MB	DATASIZE
Hive Orc Max Buffer Size	ORC: Maximum size of a single read	8MB	DATASIZE
Hive Orc Stream Buffer Size	ORC: Size of buffer for streaming reads	8MB	DATASIZE
Hive Orc Max Read Block Size	ORC: Soft max size of Presto blocks produced by ORC reader	16MB	DATASIZE
Hive Rcfile Writer Validate	Validate RCFile after write by re-reading the whole file	false	BOOLEAN
Hive Text Max Line Length	Maximum line length for text files	100MB	DATASIZE
Hive Parquet Use Column Names	Access Parquet columns using names from the file	false	BOOLEAN
Hive File Status Cache Tables	The tables that have file status cache enabled. Setting to '*' includes all tables		STRING
Hive Skip Deletion For Alter	Skip deletion of old partition data when a partition is deleted and then inserted in the same transaction	false	BOOLEAN
Hive Sorted Writing	Enable writing to bucketed sorted tables	true	BOOLEAN
Hive Ignore Table Bucketing	Ignore table bucketing to enable reading from unbucketed partitions	false	BOOLEAN
Hive Temporary Table Schema	Schema where to create temporary tables	default	STRING
Hive Pushdown Filter Enabled	Experimental: enable complex filter pushdown	false	BOOLEAN
Hive Pagefile Writer Stripe Max Size	PAGEFILE: Max stripe size	24MB	DATASIZE
Hive File_renaming_enabled	enable file renaming	false	BOOLEAN
Hive Partial_aggregation_pushdown_for_variable_length_datatypes_enabled	enable partial aggregation pushdown for variable length datatypes	false	BOOLEAN
Hive Time Zone	Sets the default time zone	null	STRING
Hive Orc Writer Stripe Min Size	ORC: Min stripe size	32MB	DATASIZE
Hive Orc Writer Stripe Max Size	ORC: Max stripe size	64MB	DATASIZE

Parameter	Description	Default Value	Data Type
Hive Orc Writer Stripe Max Rows	ORC: Max stripe row count	10000000	INTEGER
Hive Orc Writer Row Group Max Rows	ORC : Max rows in row group	10000	INTEGER
Hive Orc Writer Dictionary Max Memory	ORC: Max dictionary memory	16MB	DATASIZE
Hive Orc Writer String Statistics Limit	ORC: Maximum size of string statistics; drop if exceeding	64B	DATASIZE
Hive Orc Writer Stream Layout Type	ORC: Stream layout type	BY_COLUMN_SIZE	STRING
Hive Orc Writer Dwrf Stripe Cache Mode	Describes content of the DWRF stripe metadata cache.	INDEX_AND_FOOTER	STRING
Hive Orc Writer Max Compression Buffer Size	ORC : Max compression buffer size	256kB	DATASIZE
Hive Orc Writer Dwrf Stripe Cache Enabled	DWRF stripe cache enabled?	false	BOOLEAN
Hive Orc Writer Dwrf Stripe Cache Max Size	DWRF stripe cache max size	8MB	DATASIZE
Hive Parquet Optimized Writer Enabled	Parquet: Optimized writer enabled?	false	BOOLEAN
Hive Parquet Writer Block Size	Parquet: Writer block size	134217728B	DATASIZE
Hive Parquet Writer Page Size	Parquet: Writer page size	1048576B	DATASIZE
Hive Security	The type of access control to use	legacy	STRING
Generic Cache Enabled	Enable Caching while querying	true	BOOLEAN
Transparent Cache Enabled	Enable transparent caching while querying	true	BOOLEAN
Generic Cache Table Ttl	TTL for cache table expiry in minutes	1440	INTEGER

Delta Thrift Connection Parameters

List of Delta Thrift connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Delta Thrift connection parameters.

NOTE: Delta connector values varies based on type of metastore. Refer https://prestodb.io/docs/current/connector/deltalake.html.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Hive Metastore	The type of Hive metastore to use	thrift	STRING
Hive Metastore Uri	Hive metastore URIs (comma separated)	null	STRING
Hive S3 Aws Access Key	Default AWS access key to use for bucket access	null	STRING
Hive S3 Aws Secret Key	Default AWS secret key to use for bucket access	null	STRING
Enable Local Snapshot Table	Enable Caching while querying	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Generic Cache Table Ttl	TTL for cache table expiry in minutes	1440	INTEGER

Hive Connection Parameters

List of Hive connection parameters, descriptions, default values, and supported data types.

If you want to connect HPE Ezmeral Unified Analytics Software to a Hive data source that uses Kerberos for authentication, see Configuring a Hive Data Source with Kerberos Authentication on page 333.

The following sections list the required and optional Hive connection parameters.

NOTE: Hive connector values varies based on type of metastore. See https://prestodb.io/docs/current/connector/hive.html.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Hive Metastore	The type of Hive metastore to use.	thrift	STRING
Enable Local Snapshot Table	Enable Caching while querying.	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Hive Insert Overwrite Immutable Partitions Enabled	When enabled, insertion query will overwrite existing partitions when partitions are immutable. This config only takes effect when Hive Immutable Partitions is set to true.	false	BOOLEAN
Hive Create Empty Bucket Files For Temporary Table	Create empty files when there is no data for temporary table buckets.	false	BOOLEAN
Hive Enable Parquet Batch Reader Verification	Enable optimized parquet reader.	false	BOOLEAN
Hive Create Empty Bucket Files For Temporary Table	Create empty files when there is no data for temporary table buckets.	false	BOOLEAN
Hive Min Bucket Count To Not Ignore Table Bucketing	Ignore table bucketing when table bucket count is less than the value specified, otherwise, it is controlled by property hive.ignore-table-bucketing.	0	INTEGER
Hive Partition Statistics Based Optimization Enabled	Enables partition statistics based optimization, including partition pruning and predicate stripping.	false	BOOLEAN
Hive Experimental Optimized Partition Update Serialization Enabled	Serialize PartitionUpdate objects using binary SMILE encoding and compress with the ZSTD compression.	false	BOOLEAN

Parameter	Description	Default Value	Data Type
Hive Materialized View Missing Partitions Threshold	Materialized views with missing partitions more than this threshold falls back to the base tables at read time.	100	INTEGER
Hive S3select Pushdown Max Connections	The maximum number of client connections allowed for those operations from worker nodes.	500	INTEGER
Hive Temporary Staging Directory Enabled	Should use (if possible) temporary staging directory for write operations.	true	BOOLEAN
Hive Temporary Staging Directory Path	Location of temporary staging directory for write operations. Use ${USER} placeholder to use different location for each user.	/tmp/presto-${USER}	STRING
Hive Temporary Table Storage Format	The default file format used when creating new tables.	ORC	STRING
Hive Temporary Table Compression Codec	The compression codec to use when writing files for temporary tables.	SNAPPY	STRING
Hive Use Pagefile For Hive Unsupported Type	Automatically switch to PAGEFILE format for materialized exchange when encountering unsupported types.	true	BOOLEAN
Hive Parquet Pushdown Filter Enabled	Enable complex filter pushdown for Parquet.	false	BOOLEAN
Hive Range Filters On Subscripts Enabled	Enable pushdown of range filters on subscripts (a[2] = 5) into ORC column readers.	false	BOOLEAN
Hive Adaptive Filter Reordering Enabled	Enable adaptive filter reordering.	true	BOOLEAN
Hive Parquet Batch Read Optimization Enabled	Is Parquet batch read optimization enabled.	false	BOOLEAN
Hive Enable Parquet Dereference Pushdown	Is dereference pushdown expression pushdown into Parquet reader enabled.	false	BOOLEAN
Hive Max Metadata Updater Threads	Maximum number of metadata updated threads.	100	INTEGER
Hive Partial_aggregation_pushdown_enabled	Enable partial aggregation pushdown.	false	BOOLEAN
Hive Manifest Verification Enabled	Enable verification of file names and sizes in manifest / partition parameters.	false	BOOLEAN
Hive Undo Metastore Operations Enabled	Enable undo metastore operations.	true	BOOLEAN
Hive Verbose Runtime Stats Enabled	Enable tracking all runtime stats. Note that this may affect query performance.	false	BOOLEAN
Hive Prefer Manifests To List Files	Prefer to fetch the list of file names and sizes from manifests rather than storage	false	BOOLEAN
Hive Partition Lease Duration	Partition lease duration.	0.00s	DURATION
Hive Size Based Split Weights Enabled	Enable estimating split weights based on size in bytes	true	BOOLEAN
Hive Minimum Assigned Split Weight	Minimum weight that a split can be assigned when size based split weights are enabled.	0.05	DOUBLE
Hive Use Record Page Source For Custom Split	Use record page source for custom split. By default, true. Used to query MOR tables in Hudi.	true	BOOLEAN

Parameter	Description	Default Value	Data Type
Hive Split Loader Concurrency	Number of maximum concurrent threads per split source.	4	INTEGER
Hive Domain Compaction Threshold	Maximum ranges to allow in a tuple domain without compacting it.	100	INTEGER
Hive Max Concurrent File Renames	Maximum concurrent file renames	20	INTEGER
Hive Max Concurrent Zero Row File Creations	Maximum number of zero row file creations.	20	INTEGER
Hive Recursive Directories	Enable reading data from subdirectories of table or partition locations. If disabled, subdirectories are ignored.	false	BOOLEAN
Hive User Defined Type Encoding Enabled	Enable user defined type.	false	BOOLEAN
Hive Loose Memory Accounting Enabled	When enabled relaxes memory accounting for queries violating memory limits to run that previously honored memory thresholds.	false	BOOLEAN
Hive Max Outstanding Splits Size	Maximum amount of memory allowed for split buffering for each table scan in a query, before the query is failed.	256MB	DATASIZE
Hive Max Split Iterator Threads	Maximum number of iterator threads.	1000	INTEGER
Hive Allow Corrupt Writes For Testing	Allow Hive connector to write data even when data will likely be corrupt.	false	BOOLEAN
Hive Create Empty Bucket Files	Should empty files be created for buckets that have no data?	true	BOOLEAN
Hive Max Partitions Per Writers	Maximum number of partitions per writer.	100	INTEGER
Hive Write Validation Threads	Number of threads used for verifying data after a write.	16	INTEGER
Hive Orc Tiny Stripe Threshold	ORC: Threshold below which an ORC stripe or file will read in its entirety.	8MB	DATASIZE
Hive Orc Lazy Read Small Ranges	ORC read small disk ranges lazily.	true	BOOLEAN
Hive Orc Bloom Filters Enabled	ORC: Enable bloom filters for predicate pushdown.	false	BOOLEAN
Hive Orc Default Bloom Filter Fpp	ORC Bloom filter false positive probability.	0.05	DOUBLE
Hive Orc Optimized Writer Enabled	Experimental: ORC: Enable optimized writer.	true	BOOLEAN
Hive Orc Writer Validation Percentage	Percentage of ORC files to validate after write by re-reading the whole file.	0	DOUBLE
Hive Orc Writer Validation Mode	Level of detail in ORC validation. Lower levels require more memory.	BOTH	STRING
Hive Rcfile Optimized Writer Enabled	Experimental: RCFile: Enable optimized writer.	true	BOOLEAN
Hive Assume Canonical Partition Keys	Assume canonical parition keys?	false	BOOLEAN

Parameter	Description	Default Value	Data Type
Hive Parquet Fail On Corrupted Statistics	Fail when scanning Parquet files with corrupted statistics.	true	BOOLEAN
Hive Parquet Max Read Block Size	Parquet: Maximum size of a block to read.	16MB	DATASIZE
Hive Optimize Mismatched Bucket Count	Enable optimization to avoid shuffle when bucket count is compatible but not the same.	false	BOOLEAN
Hive Zstd Jni Decompression Enabled	Use JNI based zstd decompression for reading ORC files.	false	BOOLEAN
Hive File Status Cache Size	Hive file status cache size.	0	LONG
Hive File Status Cache Expire Time	Hive file status cache : expiry time.	0.00s	DURATION
Hive Per Transaction Metastore Cache Maximum Size	Maximum number of metastore data objects in the Hive metastore cache per transaction.	1000	INTEGER
Hive Metastore Refresh Interval	Asynchronously refresh cached metastore data after access if it is older than this but is not yet expired, allowing subsequent accesses to see fresh data.	0.00s	DURATION
Hive Metastore Cache Maximum Size	Maximum number of metastore data objects in the Hive metastore cache.	10000	INTEGER
Hive Metastore Refresh Max Threads	Maximum threads used to refresh cached metastore data.	100	INTEGER
Hive Partition Versioning Enabled		false	BOOLEAN
Hive Metastore Impersonation Enabled	Should Presto user be impersonated when communicating with Hive Metastore.	false	BOOLEAN
Hive Partition Cache Validation Percentage	Percentage of partition cache validation.	0	DOUBLE
Hive Metastore Thrift Client Socks Proxy	Metastore thrift client socks proxy.	null	STRING
Hive Metastore Timeout	Timeout for Hive metastore requests.	10.00s	DURATION
Hive Dfs Verify Checksum	Verify checksum for data consistency.	true	BOOLEAN
Hive Metastore Cache Ttl	Duration how long cached metastore data should be considered valid.	0.00s	DURATION
Hive Metastore Recording Path	Metastore recording path.	null	STRING
Hive Replay Metastore Recording	Replay metastore recording.	false	BOOLEAN
Hive Metastore Recoding Duration	Metastore recording duration.	0.00m	DURATION
Hive Dfs Require Hadoop Native	Hadoop native is required?	true	BOOLEAN
Hive Metastore Cache Scope	Metastore cache scope.	ALL	STRING
Hive Metastore Authentication Type	Hive metastore authentication type.	NONE	STRING
Hive Hdfs Authentication Type	HDFS authentication type.	NONE	STRING

Parameter	Description	Default Value	Data Type
Hive Hdfs Impersonation Enabled	Should Presto user be impersonated when communicating with HDFS.	false	BOOLEAN
Hive Hdfs Wire Encryption Enabled	Should be turned on when HDFS wire encryption is enabled.	false	BOOLEAN
Hive Skip Target Cleanup On Rollback	Skip deletion of target directories when a metastore operation fails and the write mode is DIRECT_TO_TARGET_NEW_DIRECTORY.	false	BOOLEAN
Hive Bucket Execution	Enable bucket-aware execution: only use a single worker per bucket.	true	BOOLEAN
Hive Bucket Function Type For Exchange	Hash function type for exchange.	HIVE_COMPATIBLE	STRING
Hive Ignore Unreadable Partition	Ignore unreadable partitions and report as warnings instead of failing the query.	false	BOOLEAN
Hive Max Buckets For Grouped Execution	Maximum number of buckets to run with grouped execution.	1000000	INTEGER
Hive Sorted Write To Temp Path Enabled	Enable writing temp files to temp path when writing to bucketed sorted tables.	false	BOOLEAN
Hive Sorted Write Temp Path Subdirectory Count	Number of directories per partition for temp files generated by writing sorted table.	10	INTEGER
Hive Fs Cache Max Size	Hadoop FileSystem cache size.	1000	INTEGER
Hive Non Managed Table Writes Enabled	Enable writes to non-managed (external) tables.	false	BOOLEAN
Hive Non Managed Table Creates Enabled	Enable non-managed (external) table creates.	true	BOOLEAN
Hive Table Statistics Enabled	Enable use of table statistics.	true	BOOLEAN
Hive Partition Statistics Sample Size	Specifies the number of partitions to analyze when computing table statistics.	100	INTEGER
Hive Ignore Corrupted Statistics	Ignore corrupted statistics rather than failing.	false	BOOLEAN
Hive Collect Column Statistics On Write	Enables automatic column level statistics collection on write.	false	BOOLEAN
Hive S3select Pushdown Enabled	Enable query pushdown to AWS S3 Select service.	false	BOOLEAN
Hive Max Initial Splits	Max initial splits.	200	INTEGER
Hive Max Initial Split Size	Max initial split size.	null	DATASIZE
Hive Writer Sort Buffer Size	Write sort buffer size.	64MB	DATASIZE
Hive Node Selection Strategy	Node affinity selection strategy.	NO_PREFERENCE	STRING
Hive Max Split Size	Max split size.	64MB	DATASIZE
Hive Max Partitions Per Scan	Maximum allowed partitions for a single table scan.	100000	INTEGER
Hive Max Outstanding Splits	Target number of buffered splits for each table scan in a query, before the scheduler tries to pause itself.	1000	INTEGER
Hive Metastore Partition Batch Size Min	Hive metastore : min batch size for partitions.	10	INTEGER

Parameter	Description	Default Value	Data Type
Hive Metastore Partition Batch Size Max	Hive metastore : max batch size for partitions.	100	INTEGER
Hive Config Resources	An optional comma-separated list of HDFS configuration files.	[]	FILEPATH
Hive Dfs Ipc Ping Interval	The client will send ping when the interval is passed without receiving bytes.	10.00s	DURATION
Hive Dfs Timeout	DFS timeout.	60.00s	DURATION
Hive Dfs Connect Timeout	DFS connection timeout.	500.00ms	DURATION
Hive Dfs Connect Max Retries	DFS - max retries in case of connection issue.	5	INTEGER
Hive Storage Format	The default file format used when creating new tables.	ORC	STRING
Hive Compression Codec	The compression codec to use when writing files.	GZIP	STRING
Hive Orc Compression Codec	The preferred compression codec to use when writing ORC and DWRF files.	GZIP	STRING
Hive Respect Table Format	Should new partitions be written using the existing table format or the default PrestoDB format?	true	BOOLEAN
Hive Immutable Partitions	Can new data be inserted into existing partitions?	false	BOOLEAN
Hive Max Open Sort Files	Maximum number of writer temporary files to read in one pass.	50	INTEGER
Hive Dfs Domain Socket Path	This is a path in the filesystem that allows the client and the DataNodes to communicate.	null	STRING
Hive S3 File System Type	S3 file system type.	PRESTO	STRING
Hive Gcs Json Key File Path	JSON key file used to access Google Cloud Storage.	null	FILEPATH
Hive Gcs Use Access Token	Use client-provided OAuth token to access Google Cloud Storage.	false	BOOLEAN
Hive Orc Use Column Names	Access ORC columns using names from the file.	false	BOOLEAN
Hive Orc Max Merge Distance	ORC: Maximum size of gap between two reads to merge into a single read	1MB	DATASIZE
Hive Orc Max Buffer Size	ORC: Maximum size of a single read.	8MB	DATASIZE
Hive Orc Stream Buffer Size	ORC: Size of buffer for streaming reads.	8MB	DATASIZE
Hive Orc Max Read Block Size	ORC: Soft max size of Presto blocks produced by ORC reader.	16MB	DATASIZE
Hive Rcfile Writer Validate	Validate RCFile after write by re-reading the whole file.	false	BOOLEAN
Hive Text Max Line Length	Maximum line length for text files.	100MB	DATASIZE
Hive Parquet Use Column Names	Access Parquet columns using names from the file.	false	BOOLEAN
Hive File Status Cache Tables	The tables that have file status cache enabled. Setting to '*' includes all tables.		STRING
Hive Skip Deletion For Alter	Skip deletion of old partition data when a partition is deleted and then inserted in the same transaction.	false	BOOLEAN
Hive Sorted Writing	Enable writing to bucketed sorted tables.	true	BOOLEAN

Parameter	Description	Default Value	Data Type
Hive Ignore Table Bucketing	Ignore table bucketing to enable reading from unbucketed partitions.	false	BOOLEAN
Hive Temporary Table Schema	Schema where to create temporary tables.	default	STRING
Hive Pushdown Filter Enabled	Experimental: enable complex filter pushdown.	false	BOOLEAN
Hive Pagefile Writer Stripe Max Size	PAGEFILE: Max stripe size.	24MB	DATASIZE
Hive File_renaming_enabled	Enable file renaming.	false	BOOLEAN
Hive partial_aggregation_pushdown_for_ variable_length_datatypes_enabled	Enable partial aggregation pushdown for variable length datatypes.	false	BOOLEAN
Hive Time Zone	Sets the default time zone.	null	STRING
Hive Orc Writer Stripe Min Size	ORC: Min stripe size.	32MB	DATASIZE
Hive Orc Writer Stripe Max Size	ORC: Max stripe size.	64MB	DATASIZE
Hive Orc Writer Stripe Max Rows	ORC: Max stripe row count.	10000000	INTEGER
Hive Orc Writer Row Group Max Rows	ORC : Max rows in row group.	10000	INTEGER
Hive Orc Writer Dictionary Max Memory	ORC: Max dictionary memory.	16MB	DATASIZE
Hive Orc Writer String Statistics Limit	ORC: Maximum size of string statistics; drop if exceeding.	64B	DATASIZE
Hive Orc Writer Stream Layout Type	ORC: Stream layout type.	BY_COLUMN_SIZE	STRING
Hive Orc Writer Dwrf Stripe Cache Mode	Describes content of the DWRF stripe metadata cache.	INDEX_AND_FOOTER	STRING
Hive Orc Writer Max Compression Buffer Size	ORC : Max compression buffer size.	256kB	DATASIZE
Hive Orc Writer Dwrf Stripe Cache Enabled	DWRF stripe cache enabled?	false	BOOLEAN
Hive Orc Writer Dwrf Stripe Cache Max Size	DWRF stripe cache max size.	8MB	DATASIZE
Hive Parquet Optimized Writer Enabled	Parquet: Optimized writer enabled?	false	BOOLEAN
Hive Parquet Writer Block Size	Parquet: Writer block size.	134217728B	DATASIZE
Hive Parquet Writer Page Size	Parquet: Writer page size.	1048576B	DATASIZE
Hive Security	The type of access control to use.	legacy	STRING
Generic Cache Table Ttl	TTL for cache table expiry in minutes.	1440	INTEGER

Hive Glue Metastore Connection Parameters

List of Hive Glue Metastore connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Hive Glue connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

NOTE: Hive connector values vary based on the type of metastore. See https://prestodb.io/docs/current/connector/hive.html.

Parameter	Description	Default Value	Data Type
Hive Metastore	The type of Hive metastore to use	thrift	STRING
Hive Metastore Glue Region	AWS region of the Glue Catalog	null	STRING
Hive Metastore Glue Aws Access Key	AWS access key to use to connect to the Glue Catalog. If specified along with hive.metastore.glue.aws-secret-key, this parameter takes precedence over hive.metastore.glue.iam-role.	null	STRING
Hive Metastore Glue Aws Secret Key	AWS secret key to use to connect to the Glue Catalog. If specified along with hive.metastore.glue.aws-access-key, this parameter takes precedence over hive.metastore.glue.iam-role.	null	STRING
Hive S3 Aws Access Key	Default AWS access key to use for bucket access	null	STRING
Hive S3 Aws Secret Key	Default AWS secret key to use for bucket access	null	STRING
Enable Local Snapshot Table	Enable Caching while querying	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Generic Cache Table Ttl	TTL for cache table expiry in minutes	1440	INTEGER

Hive Thrift Metastore Connection Parameters

List of Hive Thrift Metastore connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Hive Thrift Metastore connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

NOTE: The Hive connector values vary based on the type of metastore. See https://prestodb.io/docs/current/connector/hive.html.

Parameter	Description	Default Value	Data Type
Hive Metastore	The type of Hive metastore to use	thrift	STRING
Hive Metastore Uri	Hive metastore URIs (comma separated)	null	STRING
Enable Local Snapshot Table	Enable Caching while querying	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Generic Cache Table Ttl	TTL for cache table expiry in minutes	1440	INTEGER

Hive Discovery Metastore Connection Parameters

Lists Hive discovery metastore connection parameters, parameter descriptions, default values, and supported data types.

Use the Hive discovery metastore to query CSV and Parquet files. Hive discovery metastore automatically scans CSV files and Parquet footers in the specified directory to discover table schema. Hive discovery metastore does *not* require a Hive metastore service. For additional information, see Connecting to CSV and Parquet Data in an External S3 Data Source via Hive Connector on page 341.

The following sections list the required and optional Hive discovery metastore connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Data Dir	Location of the directory where files are stored.		STRING

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
File Type	Type of files stored CSV or Parquet.		STRING
Csv Header	Specifies that the file contains a header line with the names of each column in the file.	false	BOOLEAN
Csv Separator	Specifies the string that separates columns within each row (line) of the file.	,	STRING
Csv Date Format	Specifies the format for date fields	yyyy-MM-dd	STRING
Csv Timestamp Format	Specifies the format for timestamp fields	yyyy-MM-dd HH:mm:ss	STRING
Csv Row Count	Specifies the number of rows used for schema discovery.	1000	INTEGER
Csv Escape	Specifies the escape character used in the csv file.	\	STRING

MySQL Connection Parameters

List of MySQL connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional MySQL connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Connection Url	JDBC connection url.	null	STRING
Connection User	Specifies the login name of the user for the connection.	null	STRING
Connection Password	Specifies the password of the user for the connection.	null	STRING

Parameter	Description	Default Value	Data Type
Enable Local Snapshot Table	Enable Caching while querying.	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:S

Parameter	Description	Default Value	Data Type
Case Insensitive Name Matching	Match schema and table names case insensitively.	false	BOOLEAN
Case Insensitive Name Matching Cache Ttl	Duration for which remote dataset and table names will be cached. Set to 0ms to disable the cache.	1m	DURATION
Allow Drop Table	Allow connector to drop tables.	false	BOOLEAN
Mysql Auto Reconnect	When auto reconnect is enabled, presto tries to reconnect to the mysql server if it finds that connection is down. When it is disabled it will throw an error without retrying if connection is down.	true	BOOLEAN
Mysql Max Reconnects	Number of connection retries.	3	INTEGER
Mysql Connection Timeout	The time to wait while trying to establish a connection before terminating the attempt and generating an error.	10 sec	DURATION
Generic Cache Table Ttl	TTL for cache table expiry in minutes.	1440	INTEGER

Oracle Connection Parameters

List of Oracle connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Oracle connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Connection Url	JDBC connection url.	null	STRING
Connection User	Specifies the login name of the user for the connection.	null	STRING
Connection Password	Specifies the password of the user for the connection.	null	STRING
Enable Local Snapshot Table	Enable Caching while querying.	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Case Insensitive Name Matching	Match schema and table names case insensitively.	false	BOOLEAN
Case Insensitive Name Matching Cache Ttl	Duration for which remote dataset and table names will be cached. Set to 0ms to disable the cache.	1m	DURATION
Allow Drop Table	Allow connector to drop tables.	false	BOOLEAN
Oracle Synonyms Enabled	Synonyms feature enabled?	false	BOOLEAN

Parameter	Description	Default Value	Data Type
Oracle Number Default Scale	Default scale for number.	10	INTEGER
Oracle Number Rounding Mode	Default number rounding mode.	HALF_UP	STRING
Oracle Varchar Max Size	Max size for varchar datatype.	4000	INTEGER
Oracle Timestamp Precision	Specify the number of digits in the fractional second portion of the datetime.	6	INTEGER
Generic Cache Table Ttl	TTL for cache table expiry in minutes.	1440	INTEGER

Snowflake Connection Parameters

List of Snowflake connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Snowflake connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Connection Url	JDBC connection url.	null	STRING
Connection User	Specifies the login name of the user for the connection.	null	STRING
Connection Password	Specifies the password of the user for the connection.	null	STRING
Snowflake Db	Specifies the default database to use once connected.	null	STRING
Enable Local Snapshot Table	Enable Caching while querying.	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Case Insensitive Name Matching	Match schema and table names case insensitively.	false	BOOLEAN
Case Insensitive Name Matching Cache Ttl	Duration for which remote dataset and table names will be cached. Set to 0ms to disable the cache.	1m	DURATION
Snowflake Fetch Size	Gives the JDBC driver a hint as to the number of rows that should be fetched from the database when more rows are needed for ResultSet objects genrated by this Statement.	1000	INTEGER
Allow Drop Table	Allow connector to drop tables.	false	BOOLEAN
Generic Cache Table Ttl	TTL for cache table expiry in minutes.	1440	INTEGER

SQL Server Connection Parameters

List of SQL Server connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional SQL Server connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Connection Url	JDBC connection url.	null	STRING
Connection User	Specifies the login name of the user for the connection.	null	STRING
Connection Password	Specifies the password of the user for the connection.	null	STRING
Enable Local Snapshot Table	Enable Caching while querying.	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Case Insensitive Name Matching	Match schema and table names case insensitively.	false	BOOLEAN
Case Insensitive Name Matching Cache Ttl	Duration for which remote dataset and table names will be cached. Set to 0ms to disable the cache.	1m	DURATION
Allow Drop Table	Allow connector to drop tables.	false	BOOLEAN
Generic Cache Table Ttl	TTL for cache table expiry in minutes.	1440	INTEGER

Teradata Connection Parameters

List of Teradata connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Teradata connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Connection Url	JDBC connection url	null	STRING
Connection User	Specifies the login name of the user for the connection	null	STRING
Connection Password	Specifies the password of the user for the connection	null	STRING
Enable Local Snapshot Table	Enable Caching while querying	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Case Insensitive Name Matching	Match schema and table names case insensitively	false	BOOLEAN
Case Insensitive Name Matching Cache Ttl	Duration for which remote dataset and table names will be cached. Set to 0ms to disable the cache	1m	DURATION
Allow Drop Table	Allow connector to drop tables	false	BOOLEAN
Generic Cache Table Ttl	TTL for cache table expiry in minutes	1440	INTEGER

PostgreSQL Connection Parameters

List of PostgreSQL connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional PostgreSQL connection parameters.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type
Connection Url	JDBC connection url.	null	STRING
Connection User	Specifies the login name of the user for the connection.	null	STRING
Connection Password	Specifies the password of the user for the connection.	null	STRING
Enable Local Snapshot Table	Enable Caching while querying.	true	BOOLEAN

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type
Allow Drop Table	Allow connector to drop tables.	false	BOOLEAN
Case Insensitive Name Matching	Match schema and table names case insensitively.	false	BOOLEAN
Case Insensitive Name Matching Cache Ttl	Duration for which remote dataset and table names will be cached. Set to 0ms to disable the cache.	1m	DURATION
Generic Cache Table Ttl	TTL for cache table expiry in minutes.	1440	INTEGER

Iceberg Connection Parameters

List of Iceberg connection parameters, descriptions, default values, and supported data types.

The following sections list the required and optional Iceberg connection parameters.

IMPORTANT:

- Currently, Iceberg cannot use MAPRSASL to authenticate to an HPE Ezmeral Data Fabric cluster when the catalog type is hadoop; however, you can use the hive catalog type to connect Iceberg to an HPE Ezmeral Data Fabric cluster.

Required Connection Parameters

The following table lists the required connection parameters:

Parameter	Description	Default Value	Data Type	Possible Values
Name	Provide a unique name for the Iceberg data source connection.			
Iceberg Catalog Type	The catalog type for Iceberg tables.	hive	STRING	possibleValues(hive, hadoop)
Iceberg File Format	The storage file format for Iceberg tables.	PARQUET	STRING	possibleValues(PARQUET, ORC)

Parameter	Description	Default Value	Data Type	Possible Values
Iceberg Compression Codec	The compression codec to use when writing files. The available values are NONE, SNAPPY, GZIP, LZ4, and ZSTD	GZIP	STRING	possibleValues(NONE, SNAPPY, GZIP, LZ4, ZSTD)
Iceberg Catalog Cached Catalog Num	The number of Iceberg catalogs to cache, This property is required if the iceberg.catalog.type is hadoop	10	INTEGER	
Iceberg Max Partitions Per Writer	The Maximum number of partitions handled per writer.	100	INTEGER	
Iceberg Minimum Assigned Split Weight	A decimal value in the range (0, 1] used as a minimum for weights assigned to each split	0.05	DOUBLE	
Hive Metastore	The type of Hive metastore to use	thrift	STRING	possibleValues(thrift, file, glue)
Hive Metastore Catalog Dir	Hive file-based metastore catalog directory		STRING	
Hive Metastore Uri	Hive metastore URIs (comma separated).		STRING	
Hive Metastore Service Principal	The Kerberos principal of the Hive metastore service		STRING	
Hive Metastore Client Principal	The Kerberos principal that Presto will use when connecting to the Hive metastore service.		STRING	
Hive Metastore Client Keytab	Hive metastore client keytab location.		FILEPATH	
Hive Hdfs Presto Principal	The Kerberos principal that presto will use when connecting to HDFS		STRING	
Hive Hdfs Presto Keytab	HDFS client keytab location		FILEPATH	
Security Config File	Config file where rules are defined		STRING	
Security Refresh Period	Time after which rules will be refreshed from the file.		DURATION	Min(1ms)
Enable Local Snapshot Table	Enables local copy of database table for accelerated query performance	TRUE	BOOLEAN	

Optional Connection Parameters

The following table lists the optional connection parameters:

Parameter	Description	Default Value	Data Type	Possible Values
Iceberg Hadoop Config Resources	The path(s) for Hadoop configuration resources.		FILEPATH	
Iceberg Catalog Warehouse	The catalog warehouse root path for Iceberg tables.		STRING	
Hive Metastore User	Hive file-based metastore username for file access	presto	STRING	
Hive Metastore Glue Region	AWS region of the Glue Catalog.		STRING	
Hive Metastore Glue Endpoint Url	Glue API endpoint URL		STRING	
Hive Metastore Glue Pin Client To Current Region	Should the Glue client be pinned to the current EC2 region	FALSE	BOOLEAN	
Hive Metastore Glue Max Connections	Max number of concurrent connections to Glue	5	INTEGER	Min(1)
Hive Metastore Glue Max Error Retries	Maximum number of error retries for the Glue client	10	INTEGER	Min(0)
Hive Metastore Glue Default Warehouse Dir	Hive Glue metastore default warehouse directory		STRING	
Hive Metastore Glue Catalogid	The ID of the Glue Catalog in which the metadata database resides.		STRING	
Hive Metastore Glue Partitions Segments	Number of segments for partitioned Glue tables.	5	INTEGER	Min(1), Max(10)
Hive Metastore Glue Get Partition Threads	Number of threads for parallel partition fetches from Glue.	20	INTEGER	Min(1)
Hive Metastore Glue Iam Role	ARN of an IAM role to assume when connecting to the Glue Catalog.		STRING	
Hive Metastore Glue Aws Access Key	AWS access key to use to connect to the Glue Catalog. If specified along with hive.metastore.glue.aws-secret-key, this parameter takes precedence over hive.metastore.glue.iam-role.		STRING	

Parameter	Description	Default Value	Data Type	Possible Values
Hive Metastore Glue Aws Secret Key	AWS secret key to use to connect to the Glue Catalog. If specified along with hive.metastore.glue.aws-access-key, this parameter takes precedence over hive.metastore.glue.iam-role.		STRING	
Hive Metastore Username	Username for accessing the Hive metastore		STRING	
Hive Metastore Load Balancing Enabled	Enable load balancing between multiple Metastore instances	FALSE	BOOLEAN	
Hive Insert Overwrite Immutable Partitions Enabled	When enabled, insertion query will overwrite existing partitions when partitions are immutable. This config only takes effect with hive.immutable-partitions set to true	FALSE	BOOLEAN	
Hive Create Empty Bucket Files For Temporary Table	Create empty files when there is no data for temporary table buckets	FALSE	BOOLEAN	
Hive Enable Parquet Batch Reader Verification	enable optimized parquet reader	FALSE	BOOLEAN	
Hive Create Empty Bucket Files For Temporary Table	Create empty files when there is no data for temporary table buckets	FALSE	BOOLEAN	
Hive Min Bucket Count To Not Ignore Table Bucketing	Ignore table bucketing when table bucket count is less than the value specified, otherwise, it is controlled by property hive.ignore-table-bucketing	0	INTEGER	
Hive Partition Statistics Based Optimization Enabled	Enables partition statistics based optimization, including partition pruning and predicate stripping	FALSE	BOOLEAN	
Hive Experimental Optimized Partition Update Serialization Enabled	Serialize PartitionUpdate objects using binary SMILE encoding and compress with the ZSTD compression	FALSE	BOOLEAN	

Parameter	Description	Default Value	Data Type	Possible Values
Hive Materialized View Missing Partitions Threshold	Materialized views with missing partitions more than this threshold falls back to the base tables at read time	100	INTEGER	
Hive S3select Pushdown Max Connections	The maximum number of client connections allowed for those operations from worker nodes	500	INTEGER	Min(1)
Hive Temporary Staging Directory Enabled	Should use (if possible) temporary staging directory for write operations	TRUE	BOOLEAN	
Hive Temporary Staging Directory Path	Location of temporary staging directory for write operations. Use ${USER} placeholder to use different location for each user.	/tmp/presto-${USER}	STRING	
Hive Temporary Table Storage Format	The default file format used when creating new tables.	ORC	STRING	possibleValues(ORC, DWRF, PARQUET, AVRO, RCBINARY, RCTEXT, SEQUENCEFILE, JSON, TEXTFILE, CSV, PAGEFILE)
Hive Temporary Table Compression Codec	The compression codec to use when writing files for temporary tables	SNAPPY	STRING	possibleValues(NONE, SNAPPY, LZ4, ZSTD, GZIP)
Hive Use Pagefile For Hive Unsupported Type	Automatically switch to PAGEFILE format for materialized exchange when encountering unsupported types	TRUE	BOOLEAN	
Hive Parquet Pushdown Filter Enabled	Enable complex filter pushdown for Parquet	FALSE	BOOLEAN	
Hive Range Filters On Subscripts Enabled	enable pushdown of range filters on subscripts (a[2] = 5) into ORC column readers	FALSE	BOOLEAN	
Hive Adaptive Filter Reordering Enabled	Enable adaptive filter reordering	TRUE	BOOLEAN	
Hive Parquet Batch Read Optimization Enabled	Is Parquet batch read optimization enabled	FALSE	BOOLEAN	
Hive Enable Parquet Dereference Pushdown	Is dereference pushdown expression pushdown into Parquet reader enabled	FALSE	BOOLEAN	

Parameter	Description	Default Value	Data Type	Possible Values
Hive Max Metadata Updater Threads	Maximum number of metadata updated threads	100	INTEGER	Min(1)
Hive Partial_aggregation_pushdown_enabled	enable partial aggregation pushdown	FALSE	BOOLEAN	
Hive Manifest Verification Enabled	Enable verification of file names and sizes in manifest / partition parameters	FALSE	BOOLEAN	
Hive Undo Metastore Operations Enabled	Enable undo metastore operations	TRUE	BOOLEAN	
Hive Verbose Runtime Stats Enabled	Enable tracking all runtime stats. Note that this may affect query performance	FALSE	BOOLEAN	
Hive Prefer Manifests To List Files	Prefer to fetch the list of file names and sizes from manifests rather than storage	FALSE	BOOLEAN	
Hive Partition Lease Duration	Partition lease duration	0.00s	DURATION	
Hive Size Based Split Weights Enabled	Enable estimating split weights based on size in bytes	TRUE	BOOLEAN	
Hive Minimum Assigned Split Weight	Minimum weight that a split can be assigned when size based split weights are enabled	0.05	DOUBLE	Min(0, inclusive=false), Max(1)
Hive Use Record Page Source For Custom Split	Use record page source for custom split. By default, true. Used to query MOR tables in Hudi.	TRUE	BOOLEAN	
Hive Split Loader Concurrency	Number of maximum concurrent threads per split source	4	INTEGER	Min(1)
Hive Domain Compaction Threshold	Maximum ranges to allow in a tuple domain without compacting it	100	INTEGER	Min(1)
Hive Max Concurrent File Renames	Maximum concurrent file renames	20	INTEGER	
Hive Max Concurrent Zero Row File Creations	Maximum number of zero row file creations	20	INTEGER	Min(1)
Hive Recursive Directories	Enable reading data from subdirectories of table or partition locations. If disabled, subdirectories are ignored.	FALSE	BOOLEAN	

Parameter	Description	Default Value	Data Type	Possible Values
Hive User Defined Type Encoding Enabled	Enable user defined type	FALSE	BOOLEAN	
Hive Loose Memory Accounting Enabled	When enabled relaxes memory accounting for queries violating memory limits to run that previously honored memory thresholds	FALSE	BOOLEAN	
Hive Max Outstanding Splits Size	Maximum amount of memory allowed for split buffering for each table scan in a query, before the query is failed	256MB	DATASIZE	Min(1MB)
Hive Max Split Iterator Threads	Maximum number of iterator threads	1000	INTEGER	
Hive Allow Corrupt Writes For Testing	Allow Hive connector to write data even when data will likely be corrupt	FALSE	BOOLEAN	
Hive Create Empty Bucket Files	Should empty files be created for buckets that have no data?	TRUE	BOOLEAN	
Hive Max Partitions Per Writers	Maximum number of partitions per writer	100	INTEGER	Min(1)
Hive Write Validation Threads	Number of threads used for verifying data after a write	16	INTEGER	
Hive Orc Tiny Stripe Threshold	ORC: Threshold below which an ORC stripe or file will read in its entirety	8MB	DATASIZE	
Hive Orc Lazy Read Small Ranges	ORC read small disk ranges lazily	TRUE	BOOLEAN	
Hive Orc Bloom Filters Enabled	ORC: Enable bloom filters for predicate pushdown	FALSE	BOOLEAN	
Hive Orc Default Bloom Filter Fpp	ORC Bloom filter false positive probability	0.05	DOUBLE	
Hive Orc Optimized Writer Enabled	Experimental: ORC: Enable optimized writer	TRUE	BOOLEAN	
Hive Orc Writer Validation Percentage	Percentage of ORC files to validate after write by re-reading the whole file	0	DOUBLE	Min(0.0), Max(100.0)
Hive Orc Writer Validation Mode	Level of detail in ORC validation. Lower levels require more memory	BOTH	STRING	possibleValues(HASHED, DETAILED, BOTH)

Parameter	Description	Default Value	Data Type	Possible Values
Hive Rcfile Optimized Writer Enabled	Experimental: RCFile: Enable optimized writer	TRUE	BOOLEAN	
Hive Assume Canonical Partition Keys	Assume canonical parition keys?	FALSE	BOOLEAN	
Hive Parquet Fail On Corrupted Statistics	Fail when scanning Parquet files with corrupted statistics	TRUE	BOOLEAN	
Hive Parquet Max Read Block Size	Parquet: Maximum size of a block to read	16MB	DATASIZE	
Hive Optimize Mismatched Bucket Count	Enable optimization to avoid shuffle when bucket count is compatible but not the same	FALSE	BOOLEAN	
Hive Zstd Jni Decompression Enabled	use JNI based zstd decompression for reading ORC files	FALSE	BOOLEAN	
Hive File Status Cache Size	Hive file status cache size	0	LONG	
Hive File Status Cache Expire Time	Hive file status cache : expiry time	0.00s	DURATION	
Hive Per Transaction Metastore Cache Maximum Size	Maximum number of metastore data objects in the Hive metastore cache per transaction	1000	INTEGER	Min(1)
Hive Metastore Refresh Interval	Asynchronously refresh cached metastore data after access if it is older than this but is not yet expired, allowing subsequent accesses to see fresh data.	0.00s	DURATION	
Hive Metastore Cache Maximum Size	Maximum number of metastore data objects in the Hive metastore cache	10000	INTEGER	Min(1)
Hive Metastore Refresh Max Threads	Maximum threads used to refresh cached metastore data	100	INTEGER	Min(1)
Hive Partition Versioning Enabled		FALSE	BOOLEAN	
Hive Metastore Impersonation Enabled	Should Presto user be impersonated when communicating with Hive Metastore	FALSE	BOOLEAN	
Hive Partition Cache Validation Percentage	Percentage of partition cache validation	0	DOUBLE	Min(0.0), Max(100.0)

Parameter	Description	Default Value	Data Type	Possible Values
Hive Metastore Thrift Client Socks Proxy	metastore thrift client socks proxy		STRING	
Hive Metastore Timeout	Timeout for Hive metastore requests	10.00s	DURATION	
Hive Dfs Verify Checksum	Verify checksum for data consistency	TRUE	BOOLEAN	
Hive Metastore Cache Ttl	Duration how long cached metastore data should be considered valid	0.00s	DURATION	Min(0ms)
Hive Metastore Recording Path	metastore recording path		STRING	
Hive Replay Metastore Recording	replay metastore recording	FALSE	BOOLEAN	
Hive Metastore Recoding Duration	Metastore recording duration	0.00m	DURATION	
Hive Dfs Require Hadoop Native	hadoop native is required?	TRUE	BOOLEAN	
Hive Metastore Cache Scope	Metastore cache scope	ALL	STRING	possibleValues(ALL, PARTITION)
Hive Metastore Authentication Type	Hive metastore authentication type.	NONE	STRING	possibleValues(NONE, KERBEROS)
Hive Hdfs Authentication Type	HDFS authentication type.	NONE	STRING	possibleValues(NONE, KERBEROS)
Hive Hdfs Impersonation Enabled	Should Presto user be impersonated when communicating with HDFS	FALSE	BOOLEAN	
Hive Hdfs Wire Encryption Enabled	Should be turned on when HDFS wire encryption is enabled	FALSE	BOOLEAN	
Hive Skip Target Cleanup On Rollback	Skip deletion of target directories when a metastore operation fails and the write mode is DIRECT_TO_TARGET_NEW_DIRECTORY	FALSE	BOOLEAN	
Hive Bucket Execution	Enable bucket-aware execution: only use a single worker per bucket	TRUE	BOOLEAN	
Hive Bucket Function Type For Exchange	Hash function type for exchange	HIVE_COMPATIBLE	STRING	possibleValues(HIVE_COMPATIBLE, PRESTO_NATIVE)
Hive Ignore Unreadable Partition	Ignore unreadable partitions and report as warnings instead of failing the query	FALSE	BOOLEAN	

Parameter	Description	Default Value	Data Type	Possible Values
Hive Max Buckets For Grouped Execution	Maximum number of buckets to run with grouped execution	1000000	INTEGER	
Hive Sorted Write To Temp Path Enabled	Enable writing temp files to temp path when writing to bucketed sorted tables	FALSE	BOOLEAN	
Hive Sorted Write Temp Path Subdirectory Count	Number of directories per partition for temp files generated by writing sorted table	10	INTEGER	
Hive Fs Cache Max Size	Hadoop FileSystem cache size	1000	INTEGER	
Hive Non Managed Table Writes Enabled	Enable writes to non-managed (external) tables	FALSE	BOOLEAN	
Hive Non Managed Table Creates Enabled	Enable non-managed (external) table creates	TRUE	BOOLEAN	
Hive Table Statistics Enabled	Enable use of table statistics	TRUE	BOOLEAN	
Hive Partition Statistics Sample Size	Specifies the number of partitions to analyze when computing table statistics.	100	INTEGER	Min(1)
Hive Ignore Corrupted Statistics	Ignore corrupted statistics rather than failing	FALSE	BOOLEAN	
Hive Collect Column Statistics On Write	Enables automatic column level statistics collection on write	FALSE	BOOLEAN	
Hive S3select Pushdown Enabled	Enable query pushdown to AWS S3 Select service	FALSE	BOOLEAN	
Hive Max Initial Splits	Max initial splits	200	INTEGER	
Hive Max Initial Split Size	Max initial split size	null	DATASIZE	
Hive Writer Sort Buffer Size	Write sort buffer size	64MB	DATASIZE	Min(1MB), Max(1GB)
Hive Node Selection Strategy	Node affinity selection strategy	NO_PREFERENCE	STRING	possibleValues(HARD_AFFINITY, SOFT_AFFINITY, NO_PREFERENCE)
Hive Max Split Size	Max split size	64MB	DATASIZE	
Hive Max Partitions Per Scan	Maximum allowed partitions for a single table scan	100000	INTEGER	Min(1)

Parameter	Description	Default Value	Data Type	Possible Values
Hive Max Outstanding Splits	Target number of buffered splits for each table scan in a query, before the scheduler tries to pause itself	1000	INTEGER	Min(1)
Hive Metastore Partition Batch Size Min	hive metastore : min batch size for partitions	10	INTEGER	Min(1)
Hive Metastore Partition Batch Size Max	hive metastore : max batch size for partitions	100	INTEGER	Min(1)
Hive Config Resources	An optional comma-separated list of HDFS configuration files	[]	FILEPATH	
Hive Dfs Ipc Ping Interval	The client will send ping when the interval is passed without receiving bytes	10.00s	DURATION	
Hive Dfs Timeout	DFS timeout	60.00s	DURATION	Min(1ms)
Hive Dfs Connect Timeout	DFS connection timeout	500.00ms	DURATION	Min(1ms)
Hive Dfs Connect Max Retries	DFS - max retries in case of connection issue	5	INTEGER	Min(0)
Hive Storage Format	The default file format used when creating new tables.	ORC	STRING	possibleValues(ORC, DWRF, PARQUET, AVRO, RCBINARY, RCTEXT, SEQUENCEFILE, JSON, TEXTFILE, CSV, PAGEFILE)
Hive Compression Codec	The compression codec to use when writing files	GZIP	STRING	possibleValues(NONE, SNAPPY, LZ4, ZSTD, GZIP)
Hive Orc Compression Codec	The preferred compression codec to use when writing ORC and DWRF files	GZIP	STRING	possibleValues(NONE, SNAPPY, LZ4, ZSTD, GZIP)
Hive Respect Table Format	Should new partitions be written using the existing table format or the default PrestoDB format?	TRUE	BOOLEAN	
Hive Immutable Partitions	Can new data be inserted into existing partitions?	FALSE	BOOLEAN	
Hive Max Open Sort Files	Maximum number of writer temporary files to read in one pass	50	INTEGER	Min(2), Max(1000)

Parameter	Description	Default Value	Data Type	Possible Values
Hive Dfs Domain Socket Path	This is a path in the filesystem that allows the client and the DataNodes to communicate.	null	STRING	
Hive S3 File System Type	s3 file system type	PRESTO	STRING	possibleValues(PRESTO, EMRFS, HADOOP_DEFAULT)
Hive S3 Use Instance Credentials	Use the EC2 metadata service to retrieve API credentials (defaults to true). This works with IAM roles in EC2.	FALSE	BOOLEAN	
Hive S3 Encryption Materials Provider	Use a custom encryption materials provider for S3 data encryption		STRING	
Hive S3 Multipart Min File Size	Minimum file size for an S3 multipart upload	16MB	DATASIZE	
Hive S3 Multipart Min Part Size	Minimum part size for an S3 multipart upload	5MB	DATASIZE	
Hive S3 Pin Client To Current Region	Pin S3 requests to the same region as the EC2 instance where Presto is running	FALSE	BOOLEAN	
Hive S3 Upload Acl Type	Canned ACL type for S3 uploads	PRIVATE	STRING	possibleValues(AUTHENTICATED_READ, AWS_EXEC_READ, BUCKET_OWNER_FULL_CONTROL, BUCKET_OWNER_READ, LOG_DELIVERY_WRITE, PRIVATE, PUBLIC_READ, PUBLIC_READ_WRITE)
Hive S3 User Agent Prefix	The user agent prefix to use for S3 calls		STRING	
Hive S3 Skip Glacier Objects	Ignore Glacier objects rather than failing the query. This will skip data that may be expected to be part of the table or partition	FALSE	BOOLEAN	
Hive S3 Sse Enabled	Use S3 server-side encryption	FALSE	BOOLEAN	
Hive S3 Sse Type	The type of key management for S3 server-side encryption	S3	STRING	possibleValues(S3, KMS)

Parameter	Description	Default Value	Data Type	Possible Values
Hive S3 Max Client Retries	Maximum number of read attempts to retry	5	INTEGER	Min(0)
Hive S3 Max Error Retries	Maximum number of error retries, set on the S3 client	10	INTEGER	Min(0)
Hive S3 Max Backoff Time	Use exponential backoff starting at 1 second up to this maximum value when communicating with S3	10.00m	DURATION	Min(1s)
Hive S3 Max Retry Time	Maximum time to retry communicating with S3	10.00m	DURATION	Min(1ms)
Hive S3 Connect Timeout	The default timeout for creating new connections.	5.00s	DURATION	Min(1ms)
Hive S3 Socket Timeout	The default timeout for reading from a connected socket.	5.00s	DURATION	Min(1ms)
Hive S3 Max Connections	Sets the maximum number of allowed open HTTP connections	500	INTEGER	Min(1)
Hive S3 Staging Directory	Local staging directory for data written to S3.		STRING	
Hive S3 Aws Access Key	Default AWS access key to use.		STRING	
Hive S3 Aws Secret Key	Default AWS secret key to use.		STRING	
Hive S3 Endpoint	The S3 storage endpoint server.		STRING	
Hive S3 Storage Class	The S3 storage class to use when writing the data.	STANDARD	STRING	possibleValues(STANDARD, INTELLIGENT_TIERING)
Hive S3 Signer Type	Specify a different signer type for S3-compatible storage		STRING	possibleValues(S3SignerType, AWS3SignerType, AWS4SignerType, AWSS3V4SignerType, CloudFrontSignerType, QueryStringSignerType)
Hive S3 Path Style Access	Use path-style access for all requests to the S3-compatible storage	FALSE	BOOLEAN	
Hive S3 Iam Role	IAM role to assume		STRING	

Parameter	Description	Default Value	Data Type	Possible Values
Hive S3 Iam Role Session Name	AWS STS session name when IAM role to assume to access S3 buckets	presto-session	STRING	
Hive S3 Ssl Enabled	Use HTTPS to communicate with the S3 API	TRUE	BOOLEAN	
Hive S3 Kms Key Id	If set, use S3 client-side encryption and use the AWS KMS to store encryption keys and use the value of this property as the KMS Key ID for newly created objects		STRING	
Hive S3 Sse Kms Key Id	The KMS Key ID to use for S3 server-side encryption with KMS-managed keys		STRING	
Hive Gcs Json Key File Path	JSON key file used to access Google Cloud Storage		FILEPATH	
Hive Gcs Use Access Token	Use client-provided OAuth token to access Google Cloud Storage	FALSE	BOOLEAN	
Hive Orc Use Column Names	Access ORC columns using names from the file	FALSE	BOOLEAN	
Hive Orc Max Merge Distance	ORC: Maximum size of gap between two reads to merge into a single read	1MB	DATASIZE	
Hive Orc Max Buffer Size	ORC: Maximum size of a single read	8MB	DATASIZE	
Hive Orc Stream Buffer Size	ORC: Size of buffer for streaming reads	8MB	DATASIZE	
Hive Orc Max Read Block Size	ORC: Soft max size of Presto blocks produced by ORC reader	16MB	DATASIZE	
Hive Rcfile Writer Validate	Validate RCFile after write by re-reading the whole file	FALSE	BOOLEAN	
Hive Text Max Line Length	Maximum line length for text files	100MB	DATASIZE	Min(1B), Max(1GB)
Hive Parquet Use Column Names	Access Parquet columns using names from the file	FALSE	BOOLEAN	
Hive File Status Cache Tables	The tables that have file status cache enabled. Setting to '*' includes all tables		STRING	

Parameter	Description	Default Value	Data Type	Possible Values
Hive Skip Deletion For Alter	Skip deletion of old partition data when a partition is deleted and then inserted in the same transaction	FALSE	BOOLEAN	
Hive Sorted Writing	Enable writing to bucketed sorted tables	TRUE	BOOLEAN	
Hive Ignore Table Bucketing	Ignore table bucketing to enable reading from unbucketed partitions	FALSE	BOOLEAN	
Hive Temporary Table Schema	Schema where to create temporary tables	default	STRING	
Hive Pushdown Filter Enabled	Experimental: enable complex filter pushdown	FALSE	BOOLEAN	
Hive Pagefile Writer Stripe Max Size	PAGEFILE: Max stripe size	24MB	DATASIZE	
Hive File_renaming_enabled	enable file renaming	FALSE	BOOLEAN	
Hive Partial_aggregation_pushdown_for_variable_length_datatypes_enabled	enable partial aggregation pushdown for variable length datatypes	FALSE	BOOLEAN	
Hive Time Zone	Sets the default time zone		STRING	
Hive Orc Writer Stripe Min Size	ORC: Min stripe size	32MB	DATASIZE	
Hive Orc Writer Stripe Max Size	ORC: Max stripe size	64MB	DATASIZE	
Hive Orc Writer Stripe Max Rows	ORC: Max stripe row count	10000000	INTEGER	
Hive Orc Writer Row Group Max Rows	ORC : Max rows in row group	10000	INTEGER	
Hive Orc Writer Dictionary Max Memory	ORC: Max dictionary memory	16MB	DATASIZE	
Hive Orc Writer String Statistics Limit	ORC: Maximum size of string statistics; drop if exceeding	64B	DATASIZE	
Hive Orc Writer Stream Layout Type	ORC: Stream layout type	BY_COLUMN_SIZE	STRING	possibleValues(BY_STREAM_SIZE, BY_COLUMN_SIZE)
Hive Orc Writer Dwrf Stripe Cache Mode	Describes content of the DWRF stripe metadata cache.	INDEX_AND_FOOTER	STRING	possibleValues (NONE, INDEX, FOOTER, INDEX_AND_FOOTER)

Parameter	Description	Default Value	Data Type	Possible Values
Hive Orc Writer Max Compression Buffer Size	ORC : Max compression buffer size	256kB	DATASIZE	
Hive Orc Writer Dwrf Stripe Cache Enabled	DWRF stripe cache enabled?	FALSE	BOOLEAN	
Hive Orc Writer Dwrf Stripe Cache Max Size	DWRF stripe cache max size	8MB	DATASIZE	
Hive Parquet Optimized Writer Enabled	Parquet: Optimized writer enabled?	FALSE	BOOLEAN	
Hive Parquet Writer Block Size	Parquet: Writer block size	134217728B	DATASIZE	
Hive Parquet Writer Page Size	Parquet: Writer page size	1048576B	DATASIZE	
Hive Allow Add Column	Allow Hive connector to add column	FALSE	BOOLEAN	
Hive Allow Drop Column	Allow Hive connector to drop column	FALSE	BOOLEAN	
Hive Allow Drop Table	Allow Hive connector to drop table	FALSE	BOOLEAN	
Hive Allow Rename Table	Allow Hive connector to rename table	FALSE	BOOLEAN	
Hive Allow Rename Column	Allow Hive connector to rename column	FALSE	BOOLEAN	
Hive Security	The type of access control to use	legacy	STRING	possibleValues(legacy, file, read-only, sql-standard)
Generic Cache Table Ttl	TTL for cache table expiry in minutes	1440	INTEGER	

Configuring a Hive Data Source with Kerberos Authentication

Describes the required prerequisite steps to complete before you connect HPE Ezmeral Unified Analytics Software to a Hive data source that uses Kerberos authentication.

You can connect HPE Ezmeral Unified Analytics Software to a Hive data source that uses a Hive metastore and Kerberos for authentication. However, before you create the connection, manually complete the following steps:

Step 1 - Upload a krb5 configuration file to the shared location on page 333

Step 2 - Configure EzPresto to use the krb5.conf file on page 334

Step 3 - Connect HPE Ezmeral Unified Analytics Software to the Hive data source on page 337

Step 1 - Upload a krb5 configuration file to the shared location

The `krb5.conf` file contains Kerberos configuration information, including the locations of the KDCs and admin servers for the Kerberos realms used in the Hive configuration. To upload the `krb5.conf` file to a shared location, complete the following steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, go to **Data Engineering > Data Sources > Data Volumes.**.

3. Select the **shared** directory.

4. Upload a `krb5.conf file` to the shared directory.

   TIP: The name of the file must be `krb5.conf`.

Step 2 - Configure EzPresto to use the krb5.conf file

1. In the left navigation bar, go to **Tools & Frameworks > Data Engineering > EzPresto**.

2. Click on the **three dots** and select **Configure**.

3. In the window that appears, remove the entire `cmnConfigMaps` section and replace it with the following JVM properties:

```
cmnConfigMaps:
  # Configmaps common to both Presto Master and Worker
  logConfig:
    log.properties: |
      # Enable verbose logging from Presto
      #com.facebook.presto=DEBUG
  prestoMst:
    cmnPrestoCoordinatorConfig:
      config.properties: |

http-server.http.port={{ tpl .Values.ezsqlPresto.locatorService.locatorSv
cPort $ }}

discovery.uri=http://{{ tpl .Values.ezsqlPresto.locatorService.fullname
$ }}:{{ tpl .Values.ezsqlPresto.locatorService.locatorSvcPort $ }}
        coordinator=true
        node-scheduler.include-coordinator=false
        discovery-server.enabled=true
        catalog.config-dir =
{{ .Values.ezsqlPresto.stsDeployment.volumeMount.mountPathCatalog }}

catalog.disabled-connectors-for-dynamic-operation=drill,parquet,csv,sales
force,sharepoint,prestodb,raptor,kudu,redis,accumulo,elasticsearch,redshi
ft,localfile,bigquery,prometheus,mongodb,pinot,druid,cassandra,kafka,atop
,presto-thrift,ampool,hive-cache,memory,blackhole,tpch,tpcds,system,examp
le-http,jmx
        generic-cache-enabled=true
        transparent-cache-enabled=false
        generic-cache-catalog-name=cache
        generic-cache-change-detection-interval=300
        catalog.config-dir.shared=true
        node.environment=production
        plugin.dir=/usr/lib/presto/plugin
        log.output-file=/data/presto/server.log
        log.levels-file=/usr/lib/presto/etc/log.properties
        query.max-history=1000
        query.max-stage-count=1000
        query.max-memory={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        query.max-total-memory={{ mulf 0.7
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        # query.max-memory-per-node={{ mulf 0.5
( tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapSize . ) |
floor }}MB
        # query.max-total-memory-per-node={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapSize . ) |
floor }}MB
        # memory.heap-headroom-per-node={{ mulf 0.3
( tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapSize . ) |
floor }}MB
        experimental.spill-enabled=false
        experimental.spiller-spill-path=/tmp
        orm-database-url=jdbc:sqlite:/data/cache/metadata.db

plugin.disabled-connectors=accumulo,atop,cassandra,example-http,kafka,kud
u,localfile,memory,mongodb,pinot,presto-bigquery,prestodb,presto-druid,pr
esto-elasticsearch,prometheus,raptor,redis,redshift
        log.max-size=100MB
        log.max-history=10
        discovery.http-client.max-requests-queued-per-destination=10000
        dynamic.http-client.max-requests-queued-per-destination=10000
        event.http-client.max-requests-queued-per-destination=10000
        exchange.http-client.max-requests-queued-per-destination=10000

failure-detector.http-client.max-requests-queued-per-destination=10000

memoryManager.http-client.max-requests-queued-per-destination=10000

node-manager.http-client.max-requests-queued-per-destination=10000
        scheduler.http-client.max-requests-queued-per-destination=10000
        workerInfo.http-client.max-requests-queued-per-destination=10000
    jvmConfig:
      jvm.config: |
        -server
        -Xms{{ tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.minHeapS
ize . | floor }}M
        -Xmx{{ tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapS
ize . | floor }}M
        -XX:-UseBiasedLocking
        -XX:+UseG1GC
        -XX:G1HeapRegionSize={{ .Values.ezsqlPresto.configMapProp.mst.jvm
Prop.G1HeapRegionSize }}
        -XX:+ExplicitGCInvokesConcurrent
        -XX:+HeapDumpOnOutOfMemoryError
        -XX:+UseGCOverheadLimit
        -XX:+ExitOnOutOfMemoryError
        -XX:ReservedCodeCacheSize={{ .Values.ezsqlPresto.configMapProp.ms
t.jvmProp.ReservedCodeCacheSize }}
        -XX:PerMethodRecompilationCutoff=10000
        -XX:PerBytecodeRecompilationCutoff=10000
        -Djdk.attach.allowAttachSelf=true
        -Djdk.nio.maxCachedBufferSize={{ .Values.ezsqlPresto.configMapPro
p.jvmProp.maxCachedBufferSize }}
        -Dcom.amazonaws.sdk.disableCertChecking=true
        -Djava.security.krb5.conf=/data/shared/krb5.conf
  prestoWrk:
    prestoWorkerConfig:
      config.properties: |
        coordinator=false

http-server.http.port={{ tpl .Values.ezsqlPresto.locatorService.locatorSv
cPort $ }}

discovery.uri=http://{{ tpl .Values.ezsqlPresto.locatorService.fullname
$ }}:{{ tpl .Values.ezsqlPresto.locatorService.locatorSvcPort $ }}
        catalog.config-dir =
{{ .Values.ezsqlPresto.stsDeployment.volumeMount.mountPathCatalog }}

catalog.disabled-connectors-for-dynamic-operation=drill,parquet,csv,sales
force,sharepoint,prestodb,raptor,kudu,redis,accumulo,elasticsearch,redshi
ft,localfile,bigquery,prometheus,mongodb,pinot,druid,cassandra,kafka,atop
,presto-thrift,ampool,hive-cache,memory,blackhole,tpch,tpcds,system,examp
le-http,jmx
        generic-cache-enabled=true
        transparent-cache-enabled=false
        generic-cache-catalog-name=cache
        catalog.config-dir.shared=true
        node.environment=production
        plugin.dir=/usr/lib/presto/plugin
        log.output-file=/data/presto/server.log
        log.levels-file=/usr/lib/presto/etc/log.properties
        query.max-memory={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        query.max-total-memory={{ mulf 0.7
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . )
( .Values.ezsqlPresto.stsDeployment.wrk.replicaCount ) | floor }}MB
        query.max-memory-per-node={{ mulf 0.5
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . ) |
floor }}MB
        query.max-total-memory-per-node={{ mulf 0.6
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . ) |
floor }}MB
        memory.heap-headroom-per-node={{ mulf 0.2
( tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . ) |
floor }}MB
        experimental.spill-enabled=false
        experimental.spiller-spill-path=/tmp
        orm-database-url=jdbc:sqlite:/data/cache/metadata.db

plugin.disabled-connectors=accumulo,atop,cassandra,example-http,kafka,kud
u,localfile,memory,mongodb,pinot,presto-bigquery,prestodb,presto-druid,pr
esto-elasticsearch,prometheus,raptor,redis,redshift
        log.max-size=100MB
        log.max-history=10
        discovery.http-client.max-requests-queued-per-destination=10000
        event.http-client.max-requests-queued-per-destination=10000
        exchange.http-client.max-requests-queued-per-destination=10000

node-manager.http-client.max-requests-queued-per-destination=10000
        workerInfo.http-client.max-requests-queued-per-destination=10000
    jvmConfig:
      jvm.config: |
        -server
        -Xms{{ tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.minHeapS
ize . | floor }}M
        -Xmx{{ tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapS
ize . | floor }}M
        -XX:-UseBiasedLocking
        -XX:+UseG1GC
        -XX:G1HeapRegionSize={{ .Values.ezsqlPresto.configMapProp.wrk.jvm
Prop.G1HeapRegionSize }}
        -XX:+ExplicitGCInvokesConcurrent
        -XX:+HeapDumpOnOutOfMemoryError
        -XX:+UseGCOverheadLimit
        -XX:+ExitOnOutOfMemoryError
        -XX:ReservedCodeCacheSize={{ .Values.ezsqlPresto.configMapProp.wr
k.jvmProp.ReservedCodeCacheSize }}
        -XX:PerMethodRecompilationCutoff=10000
        -XX:PerBytecodeRecompilationCutoff=10000
        -Djdk.attach.allowAttachSelf=true
        -Djdk.nio.maxCachedBufferSize={{ .Values.ezsqlPresto.configMapPro
p.jvmProp.maxCachedBufferSize }}
        -Dcom.amazonaws.sdk.disableCertChecking=true
        -Djava.security.krb5.conf=/data/shared/krb5.conf
### values_cmn_configmap.yaml contents END
```

4. Click **Configure**. This updates the configuration on each of the presto pods and restarts the pods. This operation can take a few minutes.

Step 3 - Connect HPE Ezmeral Unified Analytics Software to the Hive data source

1. In the left navigation bar, go to **Data Engineering > Data Sources**.

2. Click **Add New Data Source**.

3. In the Hive tile, click **Create Connection**.

4. Using the following connection properties as an example, add the connection properties for your environment and then Connect.

```
Name = kdchive
Hive Metastore = Thrift
Hive Metastore Uri = thrift://m2-dev.mip.storage.mycorp.net:9083
Hive Metastore Authentication Type=KERBEROS
Hive Metastore Service Principal=hive/_HOST@MYCORP.NET
Hive Metastore Client Principal=supergroup@MYCORP.NET
Hive Metastore Client Keytab=<Uploaded the keytab file for supergroup
user>
Hive Hdfs Authentication Type=KERBEROS
Hive Hdfs Presto Principal=supergroup@MYCORP.NET
Hive Hdfs Presto Keytab=<Uploaded the keytab file for supergroup user>
```

Using MAPRSASL to Authenticate to Hive Metastore on HPE Ezmeral Data Fabric

Describes how to create a Hive data source connection that uses MAPRSASL to authenticate to a Hive Metastore on HPE Ezmeral Data Fabric.

You can connect Unified Analytics to a Hive Metastore in HPE Ezmeral Data Fabric that uses MAPRSASL to authenticate users. In Unified Analytics, create a Hive data source connection and provide the required connection details.

Prerequisites

If you want Unified Analytics to authenticate to the Hive Metastore in an external HPE Ezmeral Data Fabric cluster via MAPRSASL, you must provide the Hive connector (in Unified Analytics) with specific details about the HPE Ezmeral Data Fabric cluster.

To get the HPE Ezmeral Data Fabric cluster details, perform the following tasks before you complete the steps in Creating the Connection to Hive Metastore in HPE Ezmeral Data Fabric on page 338

- Get the HPE Ezmeral Data Fabric cluster details.

- Generate an impersonation ticket.

- Create a configuration file.

- Verify that ports used by HPE Ezmeral Data Fabric are available. For details, see Port Information.

> **TIP:** When you create a connection to Hive Metastore in HPE Ezmeral Data Fabric from Unified Analytics, you only have to provide this information once because the information is stored in designated configuration files. Subsequent connections can automatically use the cluster details and ticket information stored in the configuration files to access the Hive Metastore in the HPE Ezmeral Data Fabric cluster. For example, if you create subsequent Hive, Iceberg, or Delta Lake data source connections in Unified Analytics, you do not have to enter values in the **DF Cluster Details** and **Hive HDFS DF Ticket** fields when you configure the connections. For details, see Modifying the HPE Ezmeral Data Fabric Configuration Files on page 341.

Creating the Connection to Hive Metastore in HPE Ezmeral Data Fabric

To create a Unified Analytics connection to Hive Metastore in HPE Ezmeral Data Fabric that uses MAPRSASL to authenticate users, complete the following steps:

1. Sign in to Unified Analytics.

2. In the left navigation panel, go to **Data Engineering > Data Sources**.
3. On the **Data Sources** screen, click **Add New Data Source** on the **Structured Data** tab.
4. In the **Hive** tile, click **Create Connection**.
5. In the drawer that opens, add the following fields so they appear in the drawer:
   a. In the **Hive Advanced Settings** search field, type **auth** and select **Hive Metastore Authentication Type** when it appears. The field is added to the drawer.

   b. Repeat step a, but now select **Hive HDFS Authentication Type** when it appears. The field is added to the drawer.

   c. In the **Hive Advanced Settings** search field, type **DF** and select **DF Cluster Details**. The field is added to the drawer.

   d. Repeat step c, but now select **DF Cluster Name**. The field is added to the drawer.

   e. (*Optional for Hive Metastore Discovery Only*) In the **Hive Advanced Settings** search field, type **impersonation** and select **Hive HDFS Impersonation Enabled**. Also select the **Hive HDFS Presto Principal** field.
6. Complete the following fields in the drawer:

Field	Description
Name	Enter a unique name for the Hive connection.
Hive Metastore	Select **Thrift** or **Discovery**
Hive Metastore URI	Enter the Hive Metastore URI, for example: `thrift://a2-dev.mip.storage.mycorp.net:9083`
Hive Metastore Authentication Type	Select **MAPRSASL**.
Hive HDFS Authentication Type	Select **MAPRSASL**.
DF Cluster Name	Enter the name of the HPE Ezmeral Data Fabric cluster.
Hive Config Resources	Upload the configuration file. For information about how to create the file, see Creating a Configuration File on page 341.

DF Cluster Details	Enter cluster details from the `mapr-clusters.conf` file, for example:
	`bob123 secure=true`
	`a2-ab1-dev-vm123456.mip.storage.mycompany.net:7222`
	For information about how to access the `mapr-clusters.conf` file, see Getting the HPE Ezmeral Data Fabric Cluster Details on page 340.

Hive HDFS DF Ticket	Enter the impersonation ticket content, for example:
	`bob123`
	`rjB4HAbce... =`
	For information about how to generate a ticket or get ticket content, see Generating an HPE Ezmeral Data Fabric Impersonation Ticket on page 340.

Field	Description
Hive HDFS Impersonation Enabled	(*Optional for Hive Metastore Discovery Only*) Selecting this option enables HDFS impersonation. If you select this option, you must also provide the username for impersonation in the **Hive HDFS Presto Principal** field.
Hive HDFS Presto Principal	(*Optional for Hive Metastore Discovery Only*) Enter the username for impersonation.

IMPORTANT:

- If the Hive configuration with the `fs.defaultFS` property was not properly specified, the Hive connection must be deleted and recreated after restarting the EzPresto master and worker pods.
- You must use the actual name of the HPE Ezmeral Data Fabric cluster in the **DF Cluster Name**, **DF Cluster Details**, and **Hive HDFS DF Ticket** fields, and also in the `fs.defaulFS` property in the configuration file.

7. Click **Connect**.

Getting the HPE Ezmeral Data Fabric Cluster Details

To get the cluster details, complete the following steps:

1. SSH in to the HPE Ezmeral Data Fabric cluster.
2. Open the `mapr-clusters.conf` file:

```
cat /opt/mapr/conf/mapr-clusters.conf
```

3. Copy the information from the `mapr-clusters.conf` file and paste it into the **DF Cluster Name** and **DF Cluster Details** fields when you complete the fields in the drawer.

For additional information, see mapr-clusters.conf.

Generating an HPE Ezmeral Data Fabric Impersonation Ticket

For Unified Analytics to access the Hive Metastore in HPE Ezmeral Data Fabric, Unified Analytics must impersonate a user that has permission to access the Hive Metastore. Unified Analytics can only impersonate a user with a valid impersonation ticket from HPE Ezmeral Data Fabric.

To generate an impersonation ticket, complete the following steps:

1. If you are not already signed in to the cluster, SSH in to the HPE Ezmeral Data Fabric cluster.
2. Complete the steps to generate an impersonation ticket, as described in Generating an Impersonation Ticket with Ticket Generation Privileges.
3. Copy the contents of the impersonation ticket and paste it into the **Hive HDFS DF Ticket** field when you complete the fields in the drawer.

Creating a Configuration File

You must provide Unified Analytics with the file path to the HPE Ezmeral Data Fabric cluster. To do so, create a file named `config.xml` with the following Hadoop configuration property, providing the file path to the HPE Ezmeral Data Fabric cluster:

```
<configuration>
<property>
    <name>fs.defaultFS</name>
    <value>maprfs://<mapr_cluster_name>/</value>
</property>
</configuration>
```

When you complete the fields in the drawer, upload this file to the **Hive Config Resources** field.

Modifying the HPE Ezmeral Data Fabric Configuration Files

Describes how to access and edit the HPE Ezmeral Data Fabric files that Unified Analytics generates when you configure a data source connection that authenticates to HPE Ezmeral Data Fabric via MAPRSASL.

Unified Analytics stores the HPE Ezmeral Data Fabric cluster details and impersonation ticket in configuration files in the following directory:

```
/etc/presto/catalog/maprconf
```

If you modify these files, you must restart the EzPresto pods in the Unified Analytics cluster for the changes to take effect. Restarting the pods removes the in-memory cached configuration.

Modifying the Configuration Files

1. To connect to the primary `ezpresto` pod, run the following command:

   ```
   kubectl exec -i -t -n ezpresto ezpresto-sts-mst-0  -- bash
   ```

2. In the `/etc/presto/catalog/maprconf` directory, modify and save the configuration files.

   - Do not enter blank or empty lines in the files.
   - Only one entry per HPE Ezmeral Data Fabric cluster is allowed in the configuration files.

3. Restart the `ezpresto` pods:

   ```
   kubectl rollout restart statefulset -n ezpresto ezpresto-sts-mst
   kubectl rollout restart statefulset -n ezpresto ezpresto-sts-wrk
   ```

Connecting to CSV and Parquet Data in an External S3 Data Source via Hive Connector

Describes how to use the Hive connector with Presto in HPE Ezmeral Unified Analytics Software to connect to CSV and Parquet data in S3-based external data sources.

You can connect HPE Ezmeral Unified Analytics Software to any external S3-based data source through the Hive connector and Presto to access CSV and Parquet data. For example, you can connect HPE Ezmeral Unified Analytics Software to an external Data Fabric, Iceberg, or Spark cluster to access CSV and Parquet data in S3 storage within these data sources.

Connection Requirements

Connecting to an S3-based external data source has the following requirements:

- You must have read/write access on the S3 data source.

- You must provide the required information to connect, including the:
  - Access key
  - Secret key
  - S3 directory where files are stored
  - S3 endpoint

Connection fields vary depending on the type of data source you connect to (Data Fabric, Iceberg, Spark, etc.); however, the S3-related fields (credentials, directory, and endpoint) are always required regardless of the data source you are accessing S3 data through.

IMPORTANT: A Hive Metastore is not required. HPE Ezmeral Unified Analytics Software scans the files in the S3 data source to get metadata and determine data types.

For more information about the PrestoDB Hive connector, see Hive Connector.

Connecting to an S3-Based Data Source

To connect to an S3-based data source:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Data Engineering > Data Sources**.
3. On the **Structured Data** tab, click **Add New**.
4. In the **Hive** tile, click **Create Connection**.
5. In the drawer that opens, enter values in the required fields. An asterisk (*) denotes a required field.
   - For **Hive Metastore**, select **Discovery**.
   - In the **Optional Fields** search field, find and add the following fields:

Field Name	Description
Hive S3 AWS Access Key	Enter the AWS access key.
Hive S3 AWS Secret Key	Enter the AWS secret key.
Hive S3 Endpoint	Enter the S3 endpoint.
Hive S3 Path Style Access	Select the checkbox.

6. Click **Connect**. Upon successful connection, the data source is available on the **Data Sources** page.

   TROUBLE: If the connection fails, see Hive Data Source Connection Failure (S3-Based External Data Souce) on page 164.

7. In the left navigation bar, select **Data Engineering > Data Sources**.
8. In the data source tile, click **Query using Data Catalog** to access the S3 data.
9. On the **Data Catalog** page, identify the data source and select the schema to view the data.

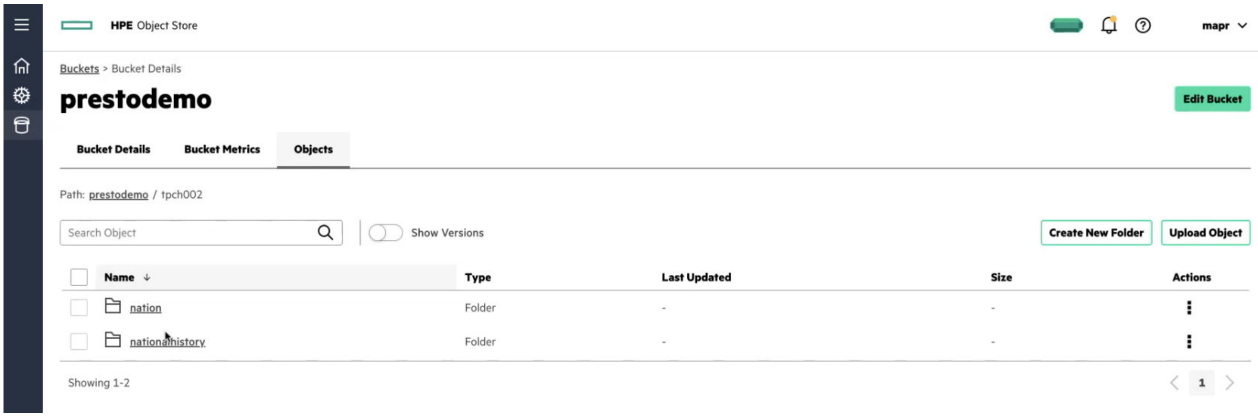
Example: Connect to HPE Ezmeral Data Fabric Object Store to access Parquet files

This example demonstrates how to connect HPE Ezmeral Unified Analytics Software to Parquet data in HPE Ezmeral Data Fabric Object Store (S3-based data store).

> **TIP:** To connect to HPE Ezmeral Data Fabric Object Store, you need working knowledge of HPE Ezmeral Data Fabric Object Store, read/write access to a bucket (granted via IAM policies), and the ability to generate the access key and secret key in HPE Ezmeral Data Fabric Object Store.

In this example, a bucket named *prestodemo* exists in HPE Ezmeral Data Fabric Object Store. Inside the bucket is a *tpch002* directory with *nation* and *nationalhistory* sub-directories that contain Parquet files.

The following image shows the HPE Ezmeral Data Fabric Object Store *prestodemo* bucket and directories within the bucket:



Name	Type	Last Updated	Size	Actions
nation	Folder	-	-	⋮
nationalhistory	Folder	-	-	⋮

Creating the HPE Ezmeral Unified Analytics Software connection to HPE Ezmeral Data Fabric Object Store requires the following information from HPE Ezmeral Data Fabric Object Store:

- Access key
- Secret key
- S3 directory where files are stored (Example: `s3://prestodemo/tpch002`)
- S3 endpoint (This is the HPE Ezmeral Data Fabric Object Store IP address and port, for example: `https://10.10.10.100:9000`

To connect HPE Ezmeral Unified Analytics Software to an S3 Directory in an external HPE Ezmeral Data Fabric Object Store:

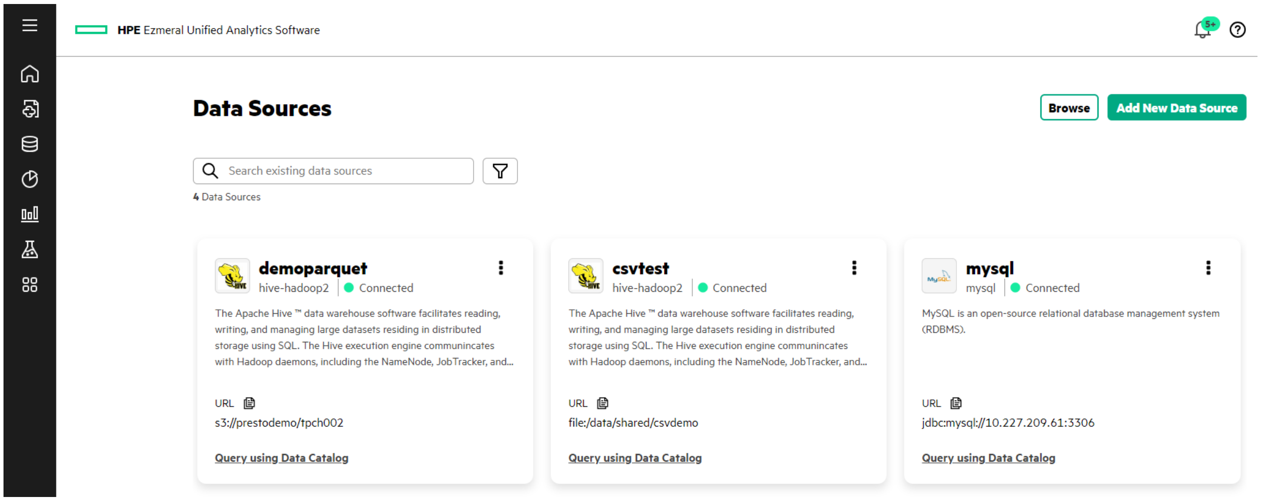
1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Data Engineering > Data Sources**.
3. On the **Structured Data** tab, click **Add New**.
4. In the **Hive** tile, click **Create Connection**.
5. In the drawer that opens, complete the required fields:

   > **TIP:**
   >
   > - In the drawer, search for and add the fields you do not see. Use the search field under **Optional Fields** to find and add these fields.
   > - No external Hive Metastore is required; HPE Ezmeral Unified Analytics Software internally parses and scans the files to get the data types and metadata.

Field Name	Example	Notes
Name	demoparquet	Unique name for the data source connection.
Hive Metastore	Discovery	System scans files to discover metadata and data types.
Data Dir	s3://prestodemo/tpch002	S3 directory where files are stored.
File Type	Parquet	You can select Parquet or CSV depending on the type of data in the S3 directory.
Hive S3 AWS Access Key	The access key generated by the Data Fabric Object Store.	Under Optional Fields, search for this field in the search box and select it.
Hive S3 AWS Secret Key	The secret key generated by the Data Fabric Object Store.	Under Optional Fields, search for this field in the search box and select it.
Hive S3 Endpoint	https://10.10.10.100:9000	The Data Fabric Object Store connection URL. Under Optional Fields, search for this field in the search box and select it.
Hive S3 Path Style Access	Select the checkbox.	Under Optional Fields, search for this field in the search box and select it.

6. Click **Connect**. The system displays the message, `"Successfully added data source,"` and the new data source (*demoparquet*) tile appears on the Data Sources page.

   TROUBLE: If the connection fails, see Hive Data Source Connection Failure (S3-Based External Data Souce) on page 164.



HPE Ezmeral Unified Analytics Software
5+
Browse
Add New Data Source
Search existing data sources

Data Sources

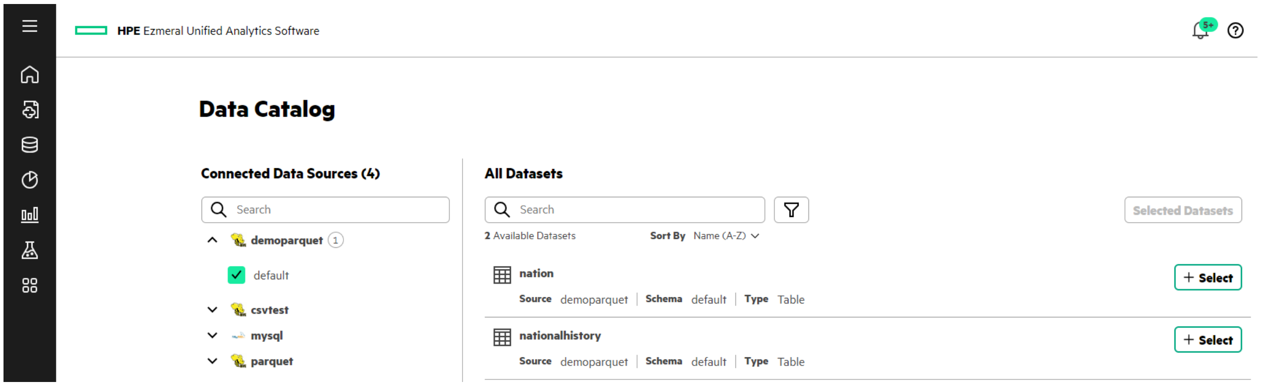
4 Data Sources

demoparquet
hive-hadoop2 | Connected
The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. The Hive execution engine communincates with Hadoop daemons, including the NameNode, JobTracker, and...
URL
s3://prestodemo/tpch002
Query using Data Catalog

csvtest
hive-hadoop2 | Connected
The Apache Hive ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. The Hive execution engine communincates with Hadoop daemons, including the NameNode, JobTracker, and...
URL
file:/data/shared/csvdemo
Query using Data Catalog

mysql
mysql | Connected
MySQL is an open-source relational database management system (RDBMS).
URL
jdbc:mysql://10.227.209.61:3306
Query using Data Catalog

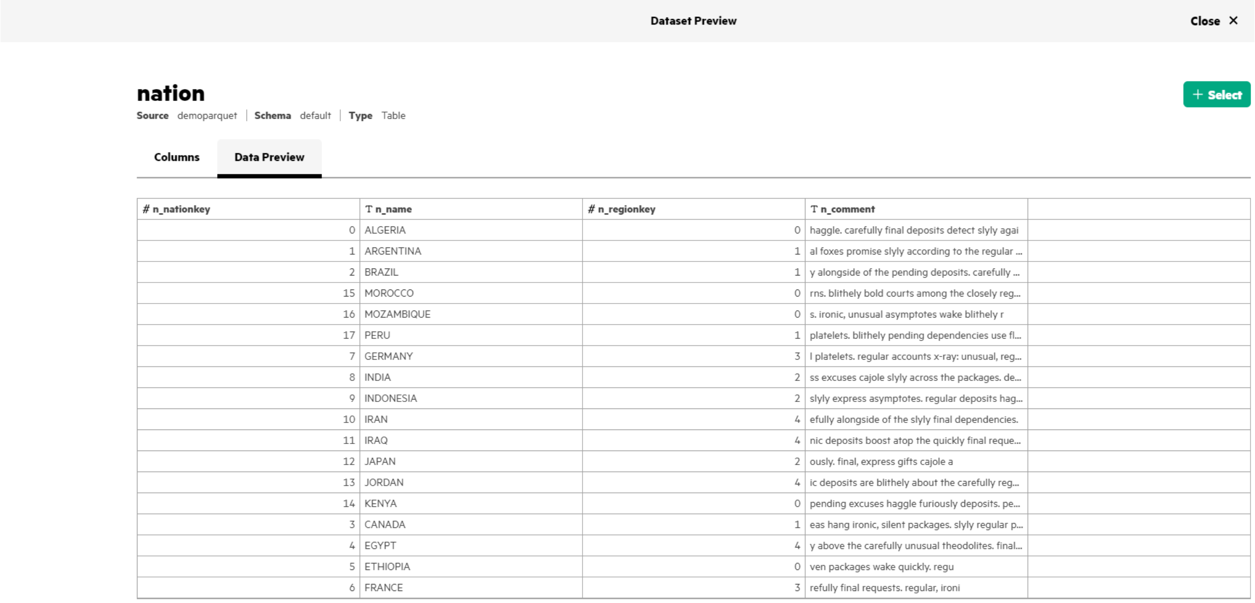
7. In the new data source tile (*demoparquet*), click **Query using Data Catalog**. You can see the `demoparquet` source listed.



8. Select the schema (*default*) under the data source to view the data sets.

NOTE: Each subdirectory in the S3 directory displays as a table in HPE Ezmeral Unified Analytics Software, and each Parquet file in the subdirectory is a row in the table.

9. Click on a table (*nation*) and then click the **Data Preview** tab to view the Parquet data.



# n_nationkey	T n_name	# n_regionkey	T n_comment	
0	ALGERIA	0	haggle. carefully final deposits detect slyly agai	
1	ARGENTINA	1	al foxes promise slyly according to the regular ...	
2	BRAZIL	1	y alongside of the pending deposits. carefully ...	
15	MOROCCO	0	rns. blithely bold courts among the closely reg...	
16	MOZAMBIQUE	0	s. ironic, unusual asymptotes wake blithely r	
17	PERU	1	platelets. blithely pending dependencies use fl...	
7	GERMANY	3	l platelets. regular accounts x-ray: unusual, reg...	
8	INDIA	2	ss excuses cajole slyly across the packages. de...	
9	INDONESIA	2	slyly express asymptotes. regular deposits hag...	
10	IRAN	4	efully alongside of the slyly final dependencies.	
11	IRAQ	4	nic deposits boost atop the quickly final reque...	
12	JAPAN	2	ously. final, express gifts cajole a	
13	JORDAN	4	ic deposits are blithely about the carefully reg...	
14	KENYA	0	pending excuses haggle furiously deposits. pe...	
3	CANADA	1	eas hang ironic, silent packages. slyly regular p...	
4	EGYPT	4	y above the carefully unusual theodolites. final...	
5	ETHIOPIA	0	ven packages wake quickly. regu	
6	FRANCE	3	refully final requests. regular, ironi	

Connecting External Applications to EzPresto via JDBC

Describes how to connect external applications and BI tools, such as Tableau and PowerBI, to EzPresto through the EzPresto JDBC endpoint.

Connecting applications to EzPresto provides users with the convenience of using their preferred applications and the ability to leverage the high performance SQL query engine to quickly build out powerful executive charts and dashboards from massive amounts of data.

You can connect external applications to EzPresto using the JDBC connection URL or code.

Getting the JDBC Endpoint

You can get the JDBC endpoint in Unified Analytics by going to **Administration > Settings** in the left navigation bar. The **JDBC Endpoint** is displayed on the **Configurations** tab.

JDBC Connection URL

Use the following URL to connect EzPresto to external applications, replacing the domain with your Unified Analytics cluster domain:

```
jdbc:presto://ezpresto.<unified-analytics-cluster-domain>:443
```

JDBC Connection Code

To programmatically connect external applications to EzPresto, use the following code and enter values specific to your Unified Analytics cluster domain and user:

```
String url = "jdbc:presto://ezpresto.unified-analytics-cluster-domain:443";
Properties properties = new Properties();
properties.setProperty("user", "ua-user");
properties.setProperty("password", "ua-user-password");
properties.setProperty("SSL", "true");
Connection connection = DriverManager.getConnection(url, properties);
```

Tableau Connection Example

The following example shows you how to connect Tableau to EzPresto:

```
Server : ezpresto.<unified-analytics-cluster-domain-name>
Port : 443
Catalog : <catalog-from-presto to which user wants to connect>
Authentication : LDAP
Username : <ua-user-name>
Password: <ua-user-password>
Require SSL : true
```

NOTE: The Tableau connector does not support SSO. The username and password are required

Using Spark to Query EzPresto

Describes how to use Spark to query EzPresto.

Using Spark to Query EzPresto via EzPresto Plugin

Spark images packaged with HPE Ezmeral Unified Analytics Software include the Presto client library required to connect Spark to EzPresto.

The EzPresto plugin retrieves and processes vast datasets quickly and efficiently compared to the legacy method, making it suitable for large-scale data operations.

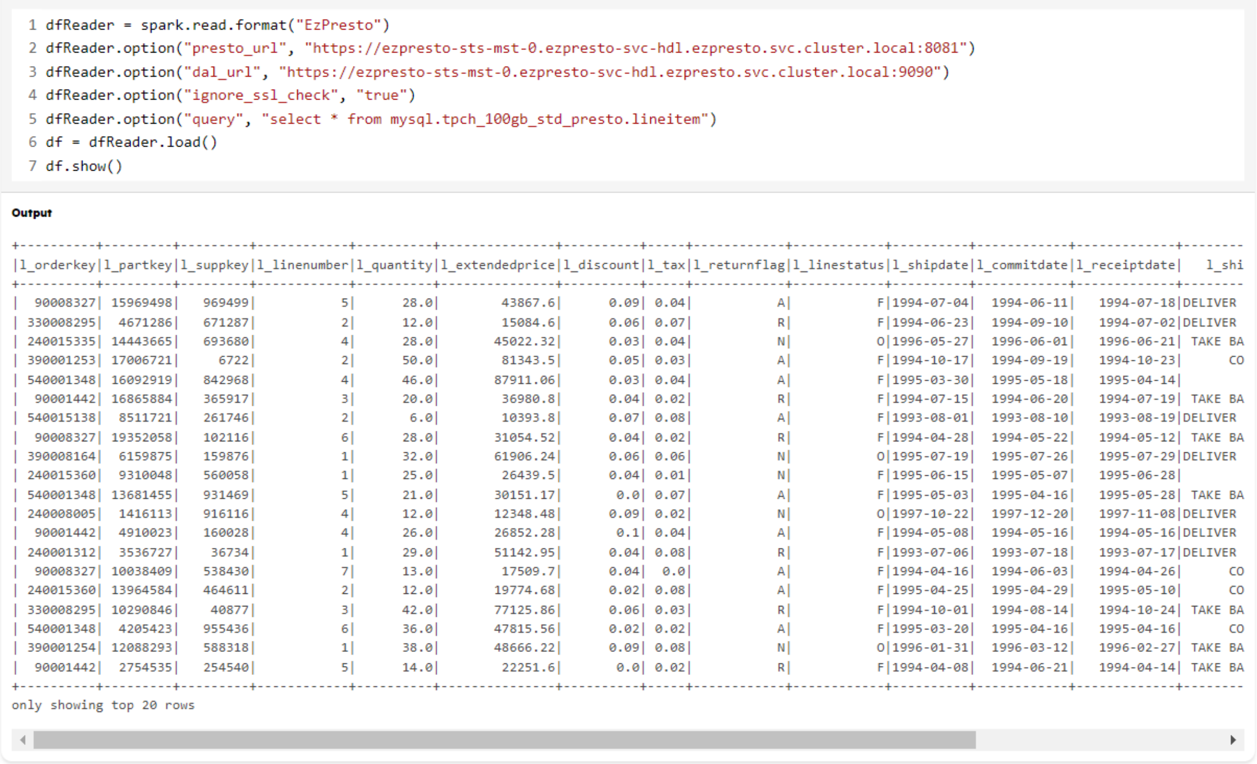
The bundled EzPresto client library automatically sets and manages credential renewal, removing the need to explicitly provide username and password as connection parameters.

Spark runtimes do not have the EzPresto certificate in the truststore; therefore, when you connect Spark to EzPresto, you must set the `ignore_ssl_check` option to `true`.

The following example shows you how to use Spark to submit a dataframe to query EzPresto:

```
dfReader = spark.read.format("EzPresto")
dfReader.option("presto_url", "https://
ezpresto-sts-mst-0.ezpresto-svc-hdl.ezpresto.svc.cluster.local:8081")
dfReader.option("dal_url", "https://
ezpresto-sts-mst-0.ezpresto-svc-hdl.ezpresto.svc.cluster.local:9090")
dfReader.option("ignore_ssl_check", "true")
dfReader.option("query", "select * from
mysql.tpch_100gb_std_presto.lineitem")
```

```
df = dfReader.load()
df.show()
```



```
dfReader = spark.read.format("EzPresto")
dfReader.option("presto_url", "https://ezpresto-sts-mst-0.ezpresto-svc-hdl.ezpresto.svc.cluster.local:8081")
dfReader.option("dal_url", "https://ezpresto-sts-mst-0.ezpresto-svc-hdl.ezpresto.svc.cluster.local:9090")
dfReader.option("ignore_ssl_check", "true")
dfReader.option("query", "select * from mysql.tpch_100gb_std_presto.lineitem")
df = dfReader.load()
df.show()
```

Output

```
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+-------------+--------
|l_orderkey|l_partkey|l_suppkey|l_linenumber|l_quantity|l_extendedprice|l_discount|l_tax|l_returnflag|l_linestatus|l_shipdate|l_commitdate|l_receiptdate|   l_shi
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+-------------+--------
|  90008327| 15969498|   969499|           5|      28.0|        43867.6|      0.09| 0.04|           A|           F|1994-07-04|  1994-06-11|   1994-07-18|DELIVER
| 330008295|  4671286|   671287|           2|      12.0|        15084.6|      0.06| 0.07|           R|           F|1994-06-23|  1994-09-10|   1994-07-02|DELIVER
| 240015335| 14443665|   693680|           4|      28.0|       45022.32|      0.03| 0.04|           N|           O|1996-05-27|  1996-06-01|   1996-06-21| TAKE BA
| 390001253| 17006721|     6722|           2|      50.0|        81343.5|      0.05| 0.03|           A|           F|1994-10-17|  1994-09-19|   1994-10-23|      CO
| 540001348| 16092919|   842968|           4|      46.0|       87911.06|      0.03| 0.04|           A|           F|1995-03-30|  1995-05-18|   1995-04-14|
|  90001442| 16865884|   365917|           3|      20.0|        36980.8|      0.04| 0.02|           R|           F|1994-07-15|  1994-06-20|   1994-07-19| TAKE BA
| 540015138|  8511721|   261746|           2|       6.0|        10393.8|      0.07| 0.08|           A|           F|1993-08-01|  1993-08-10|   1993-08-19|DELIVER
|  90008327| 19352058|   102116|           6|      28.0|       31054.52|      0.04| 0.02|           R|           F|1994-04-28|  1994-05-22|   1994-05-12| TAKE BA
| 390008164|  6159875|   159876|           1|      32.0|       61906.24|      0.06| 0.06|           N|           O|1995-07-19|  1995-07-26|   1995-07-29|DELIVER
| 240015360|  9310048|   560058|           1|      25.0|        26439.5|      0.04| 0.01|           N|           F|1995-06-15|  1995-05-07|   1995-06-28|
| 540001348| 13681455|   931469|           5|      21.0|       30151.17|       0.0| 0.07|           A|           F|1995-05-03|  1995-04-16|   1995-05-28| TAKE BA
| 240008005|  1416113|   916116|           4|      12.0|       12348.48|      0.09| 0.02|           N|           O|1997-10-22|  1997-12-20|   1997-11-08|DELIVER
|  90001442|  4910023|   160028|           4|      26.0|       26852.28|       0.1| 0.04|           A|           F|1994-05-08|  1994-05-16|   1994-05-16|DELIVER
| 240001312|  3536727|    36734|           1|      29.0|       51142.95|      0.04| 0.08|           R|           F|1993-07-06|  1993-07-18|   1993-07-17|DELIVER
|  90008327| 10038409|   538430|           7|      13.0|        17509.7|      0.04|  0.0|           A|           F|1994-04-16|  1994-06-03|   1994-04-26|      CO
| 240015360| 13964584|   464611|           2|      12.0|       19774.68|      0.02| 0.08|           A|           F|1995-04-25|  1995-04-29|   1995-05-10|      CO
| 330008295| 10290846|    40877|           3|      42.0|       77125.86|      0.06| 0.03|           R|           F|1994-10-01|  1994-08-14|   1994-10-24| TAKE BA
| 540001348|  4205423|   955436|           6|      36.0|       47815.56|      0.02| 0.02|           A|           F|1995-03-20|  1995-04-16|   1995-04-16|      CO
| 390001254| 12088293|   588318|           1|      38.0|       48666.22|      0.09| 0.08|           N|           O|1996-01-31|  1996-03-12|   1996-02-27| TAKE BA
|  90001442|  2754535|   254540|           5|      14.0|        22251.6|       0.0| 0.02|           R|           F|1994-04-08|  1994-06-21|   1994-04-14| TAKE BA
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+-------------+--------
only showing top 20 rows
```

Using Spark to Query EzPresto via Legacy Method

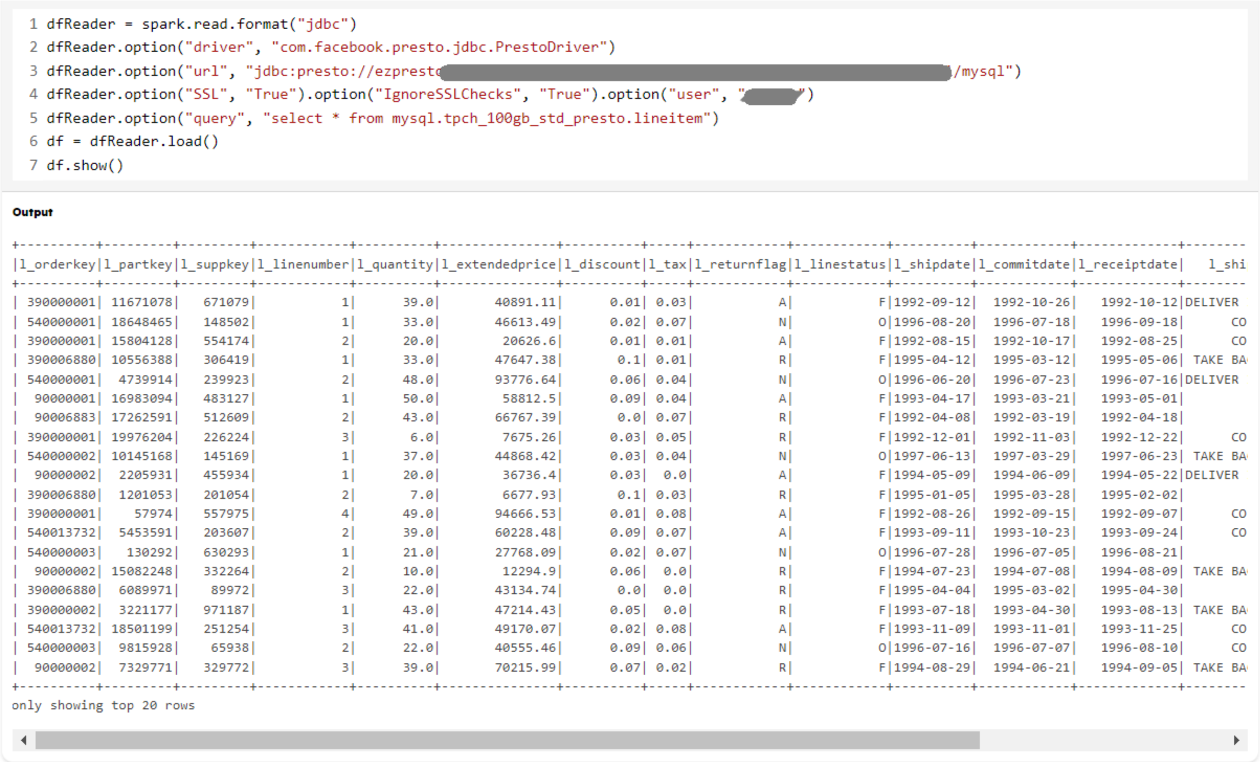
Spark images packaged with HPE Ezmeral Unified Analytics Software include the Presto client library required to connect Spark to EzPresto.

Spark runtimes do not have the EzPresto certificate in the truststore; therefore, when you connect Spark to EzPresto, you must set the `IgnoreSSLChecks` option to `true`.

For open-source Spark use cases, the CA certificate must be available in the JVM truststore. You can set a custom path to the truststore using the jdbc `SSLTrustStorePath` option. Note that you must use JKS format.

The following example shows you how to use Spark to submit a dataframe to query EzPresto:

```
DOMAIN='<your-unified-analytics-domain-name>'

df = spark.read.format("jdbc").\
      option("driver", "com.facebook.presto.jdbc.PrestoDriver").\
      option("url", f"jdbc:presto://ezpresto.{DOMAIN}.com:443/cache/
default>").\
      option("user", "<username>").\
      option("SSL", "true").\
      option("IgnoreSSLChecks", "true").\
      option("query", "select * from
mysql.tpch_100gb_std_presto.lineitem").\
      load().show()
```



```
dfReader = spark.read.format("jdbc")
dfReader.option("driver", "com.facebook.presto.jdbc.PrestoDriver")
dfReader.option("url", "jdbc:presto://ezpresto[redacted]/mysql")
dfReader.option("SSL", "True").option("IgnoreSSLChecks", "True").option("user", "[redacted]")
dfReader.option("query", "select * from mysql.tpch_100gb_std_presto.lineitem")
df = dfReader.load()
df.show()
```

Output

```
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+-------------+--------
|l_orderkey|l_partkey|l_suppkey|l_linenumber|l_quantity|l_extendedprice|l_discount|l_tax|l_returnflag|l_linestatus|l_shipdate|l_commitdate|l_receiptdate|   l_shi
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+-------------+--------
| 390000001| 11671078|   671079|           1|      39.0|       40891.11|      0.01| 0.03|           A|           F|1992-09-12|  1992-10-26|   1992-10-12|DELIVER
| 540000001| 18648465|   148502|           1|      33.0|       46613.49|      0.02| 0.07|           N|           O|1996-08-20|  1996-07-18|   1996-09-18|      CO
| 390000001| 15804128|   554174|           2|      20.0|        20626.6|      0.01| 0.01|           A|           F|1992-08-15|  1992-10-17|   1992-08-25|      CO
| 390006880| 10556388|   306419|           1|      33.0|       47647.38|       0.1| 0.01|           R|           F|1995-04-12|  1995-03-12|   1995-05-06| TAKE BA
| 540000001|  4739914|   239923|           2|      48.0|       93776.64|      0.06| 0.04|           N|           O|1996-06-20|  1996-07-23|   1996-07-16|DELIVER
|  90000001| 16983094|   483127|           1|      50.0|        58812.5|      0.09| 0.04|           A|           F|1993-04-17|  1993-03-21|   1993-05-01|
|  90006883| 17262591|   512609|           2|      43.0|       66767.39|       0.0| 0.07|           R|           F|1992-04-08|  1992-03-19|   1992-04-18|
| 390000001| 19976204|   226224|           3|       6.0|        7675.26|      0.03| 0.05|           R|           F|1992-12-01|  1992-11-03|   1992-12-22|      CO
| 540000002| 10145168|   145169|           1|      37.0|       44868.42|      0.03| 0.04|           N|           O|1997-06-13|  1997-03-29|   1997-06-23| TAKE BA
|  90000002|  2205931|   455934|           1|      20.0|        36736.4|      0.03|  0.0|           A|           F|1994-05-09|  1994-06-09|   1994-05-22|DELIVER
| 390006880|  1201053|   201054|           2|       7.0|        6677.93|       0.1| 0.03|           R|           F|1995-01-05|  1995-03-28|   1995-02-02|
| 390000001|    57974|   557975|           4|      49.0|       94666.53|      0.01| 0.08|           A|           F|1992-08-26|  1992-09-15|   1992-09-07|      CO
| 540013732|  5453591|   203607|           2|      39.0|       60228.48|      0.09| 0.07|           A|           F|1993-09-11|  1993-10-23|   1993-09-24|      CO
| 540000003|   130292|   630293|           1|      21.0|       27768.09|      0.02| 0.07|           N|           O|1996-07-28|  1996-07-05|   1996-08-21|
|  90000002| 15082248|   332264|           2|      10.0|        12294.9|      0.06|  0.0|           R|           F|1994-07-23|  1994-07-08|   1994-08-09| TAKE BA
| 390006880|  6089971|    89972|           3|      22.0|       43134.74|       0.0|  0.0|           R|           F|1995-04-04|  1995-03-02|   1995-04-30|
| 390000002|  3221177|   971187|           1|      43.0|       47214.43|      0.05|  0.0|           R|           F|1993-07-18|  1993-04-30|   1993-08-13| TAKE BA
| 540013732| 18501199|   251254|           3|      41.0|       49170.07|      0.02| 0.08|           A|           F|1993-11-09|  1993-11-01|   1993-11-25|      CO
| 540000003|  9815928|    65938|           2|      22.0|       40555.46|      0.09| 0.06|           N|           O|1996-07-16|  1996-07-07|   1996-08-10|      CO
|  90000002|  7329771|   329772|           3|      39.0|       70215.99|      0.07| 0.02|           R|           F|1994-08-29|  1994-06-21|   1994-09-05| TAKE BA
+----------+---------+---------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+-------------+--------
only showing top 20 rows
```

Limitations

The EzPresto connector for Spark has the following limitations:

- Does not support write operations.
- Does not support queries that require ordering of results, such as sort by or order by queries. While the query does not fail, the ordering is not maintained.
- You must always use aliases in SQL aggregations to ensure proper functionality. For example, replace `COUNT()` with `COUNT() AS col_name`.

Connecting to EzPresto via Python Client

Provides information for connecting to EzPresto from a Python client.

Connecting to EzPresto from a Python client is useful for Notebooks. Use the `presto-python-client` and `presto` packages to connect to EzPresto from a Python client.

Required Packages

Run the following commands to install the `presto-python-client` and `presto` packages:

```
pip install presto-python-client

pip install presto
```

Example

The following code example shows you how to connect to EzPresto from a Python client:

```
import urllib3
import uuid
import requests
from requests.packages.urllib3.exceptions import InsecureRequestWarning
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
import warnings
warnings.filterwarnings('ignore')  #This will ignore the warnings. Warnings
will not display in the notebook.
import prestodb
import getpass


class DBComponentEzsql(object):
    def __init__(self, **args):
        self._db_version = str
        self._http_scheme = args['http_scheme']
        self._schema = args['schema']
        self._catelog = args['catelog']
        self._host = args['host']
        self._user = args['user']
        self._pwd = args['password']
        self._port = args['port']
        self._test_query= "select database();"
        self._cursor = object
        self._connection = object
        self._err = "Exception while connecting to PrestoDB, there, check
with your Administrator !!!"

    # this is the prestodb connect component user defined function.
    def _connect(self)->object:
        try:
            with prestodb.dbapi.connect(host=self._host,
port=self._port, user=self._user, catalog=self._catelog,
schema= self._schema, http_scheme=self._http_scheme,
auth=prestodb.auth.BasicAuthentication(self._user, self._pwd)) as
self._connection:
                self._connection._http_session.verify = False

            if self._connection:
                return self._connection

        except Exception as e:
            print(self._err, e)
            exit(0)

        finally:
            if self._connection:
                self._connection.close()

    # this is the prestodb connect component user defined function.
    def _old_connect(self)->object:
        try:
            with prestodb.dbapi.connect(host=self._host,
port=self._port, user=self._user, catalog=self._catelog,
schema=self._schema, http_scheme=self._http_scheme,
auth=prestodb.auth.BasicAuthentication(self._user, self._pwd)) as
self._connection:
                self._connection._http_session.verify = False

            if self._connection:
                return self._connection

        except Exception as e:
            print(self._err, e)
            exit(0)
        finally:
            if self._connection:
                self._connection.close()

    #returns sql schema consisted table details
    def _get_sql_schema(self, **args)->list:
        try:
            self._cursor = self._connection.cursor()
            # self._cursor.execute('show catalogs')
            self._cursor.execute('SHOW SCHEMAS')
            _db_list = self._cursor.fetchall()
            return _db_list
        except Exception as e:
            print(self._err, e)
        finally:
            if self._connection:
                self._connection.close()
                self._cursor.close()

    #returns sql schema consisted table details
    def _get_sql_tables(self, **args)->list:
        try:
            self._connection.close()
            if self._schema != None and args["run_schema"] != None:
                self._schema = args["run_schema"]
                self._connection = self._connect()
                self._cursor = self._connection.cursor()
                self._cursor.execute('show tables')
                _table_list = self._cursor.fetchall()
                return _table_list
        except Exception as e:
            print(self._err, e)
        finally:
            if self._connection:
                self._connection.close()
                self._cursor.close()

    #returns sql table persisted data
    def _get_data(self,**args)->list:
        try:
            if args['table_name']!= None:
                ''' This generalized sql query we must need to extend '''
                str_query = f"SELECT * FROM {args['table_name']}"
                self._cursor = self._connection.cursor()
                self._cursor.execute(str_query)
                res_data = self._cursor.fetchall()
                return res_data
        except Exception as e:
            print(self._err, e)
        finally:
            if self._connection:
                self._connection.close()
                self._cursor.close()

if __name__ == "__main__":
    try:
        # config to validate the schema name:
        config = {
            "host":"ezsql.hpe-qa1-ezaf.com",
            "catelog":"mysql",
            "user":"hpedemo-user01",
            "password":"Hpepoc@123",
            "schema":"retailstore",
            "http_scheme":"https",
            "port":443,
            "table": "call_center"
        }

        ezobj = DBComponentEzsql(
            host=config.get("host"),
            catelog=config.get("catelog"),
            schema="default",
            user= config.get("user"),
            password=config.get("password"),
            http_scheme = config.get("http_scheme"),
            port=config.get("port"))
        conn = ezobj._connect()
        print("-"*100, conn)

        ''' How we can use the developed core Ezmeral unified analytics
Ezsql component explained bellow !!!'''
        if conn:
            print("-"*100," print list of schams ", ezobj._get_sql_schema())
            for item in range(0, len(ezobj._get_sql_schema())):
                # validate desired scema:
                if config.get("schema") in ezobj._get_sql_schema()[item]:

print(ezobj._get_sql_tables(run_schema=config.get("schema")))
                    print(ezobj._get_data(table_name=config.get("table")))

    except Exception as e:
        print(ezobj._err, e)
```

Caching Data

Describes data caching and provides the steps for caching data in HPE Ezmeral Unified Analytics Software.

Caching data reduces latency. You can pre-load frequently accessed data into the cache to improve the performance of queries on the data. Caching is useful when network latency is an issue due to firewalls.

When queries run against data or tables, the query engine automatically checks for cached data and uses it if present. Cache optimization works when queries reference remote tables. Queries issued against cached data do not require optimization.

The cache lasts for the duration of the TTL (time-to-live). The user that connects HPE Ezmeral Unified Analytics Software to a data source selects the caching option (**Enable Local Snapshot Table**) and sets the TTL for the cache. The default TTL is one day (set in minutes).

HPE Ezmeral Unified Analytics Software stores cached data in an HPE Ezmeral Data Fabric volume. You can view and access cached data in the HPE Ezmeral Unified Analytics Software UI by going to **Data Engineering > Cached Assets** in the left navigation bar.

When you cache data, you can modify the data sets before caching them. The following list describes some of the changes that you can make to a data set:

- Edit the data set name
- Remove columns from the data set
- Edit column names
- Change the schema or add a new schema
- Apply a schema to the selected data sets

IMPORTANT:

- Cached data is only available to the user that cached the data. Other users that sign in to HPE Ezmeral Unified Analytics Software cannot access the data that you cache.
- If data in the underlying data sources change, HPE Ezmeral Unified Analytics Software does not automatically update the cache. You must cache the data again to refresh the cache.

How to Cache Data

HPE Ezmeral Unified Analytics Software must be connected to the data sources with the data sets that you want to cache. See Connecting Data Sources on page 297.

To cache data, complete the following steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Data Engineering > Data Catalog.**
3. In the **Connected Data Sources** area, select the data sources with the data that you want to cache. The data sets available to you in the selected data sources displays in the **All Datasets** area.
4. Optionally, search or filter the data sets to find the data set(s) that you want to cache.
5. Click **+ Select** for each of the data sets that you want to cache.
6. Click **Selected Datasets**. The **Selected Datasets** drawer opens and displays the selected data sets.
7. Click **Cache Datasets**. The **Manage Datasets** screen appears. Each data set that you selected appears on its own tab.
8. Optionally, modify the data set(s).
   - Use the **pencil icon** to modify data set and column names.
   - Use the **check boxes** next to the column names to remove columns from the data set.
   - Use the **Schema** dropdown to change the schema or add a new schema.
   - If you have selected multiple data sets, use the **connector icon** next to the schema dropdown to apply the schema to all of the selected data sets.
9. Click **Cache Overview** and compare the original data sets (Input Assets) to the modified data sets (Output Assets) to verify the changes.
10. If the changes to a data set are incorrect, click the **pencil icon** to edit the data set.
11. To cache the data set(s), click **Save to cache**. The system displays the following message:

    ```
    Successfully initiating cache
    ```

    If an error appears, correct the issue and continue.
12. To view the cached data sets, go to **Data Engineering > Cached Assets** in the left navigation bar of the HPE Ezmeral Unified Analytics Software UI.

    NOTE: Depending on the size of the data sets, it may take a minute or so for them to appear as cached assets.

Enable or Disable a Cache

You can enable or disable caching through the **Enable Local Snapshot Table** option when you create a data source connection. See Connecting Data Sources on page 297.

You cannot disable caching by setting the TTL to zero. If the TTL is set to zero, the cache expires immediately but still consumes resources.

Submitting Presto Queries from Notebook

Describes how to submit Presto queries from the notebook.

In HPE Ezmeral Unified Analytics Software, you can connect to SQL databases and submit queries through EzPresto using the `%sql` or `%%sql` magic. See Notebook Magic Functions on page 445%sql and %%sql on page 450

Airflow

Provides an overview of Apache Airflow in HPE Ezmeral Unified Analytics Software.

You can use Airflow to author, schedule, or monitor workflows and data pipelines.

A workflow is a Directed Acyclic Graph (DAG) of tasks used to handle big data processing pipelines. The workflows are started on a schedule or triggered by an event. DAGs define the order to run tasks or rerun tasks in case of failures. The tasks define the actions to be performed, such as ingest, monitor, report, and others.

To learn more, see Airflow documentation.

Airflow Functionality

Airflow in HPE Ezmeral Unified Analytics Software supports the following functionality:

- Extracting data from multiple data sources and running Spark jobs or other data transformations.
- Training machine learning models.
- Automated generation of reports.
- Backups and other DevOps tasks.

Airflow Architecture

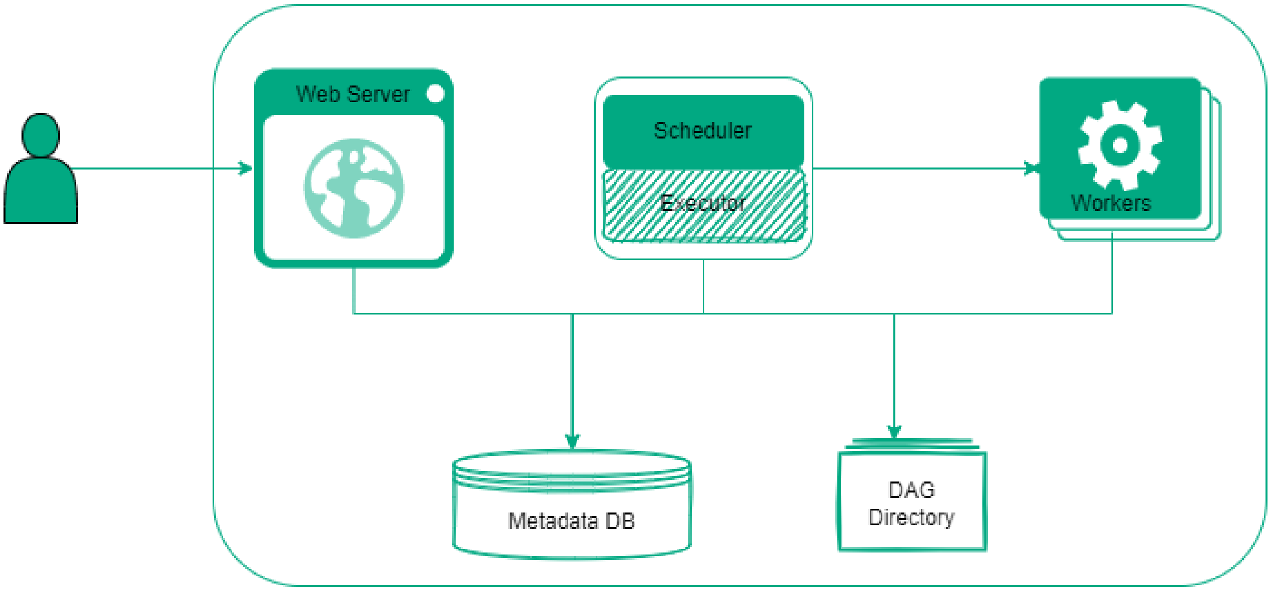
In HPE Ezmeral Unified Analytics Software, Airflow consists of the following parts:

Airflow Operator	Manages and maintains Airflow Base and Airflow Cluster Kubernetes Custom Resources by creating and updating Kubernetes objects.
Airflow Base	Manages the PostgreSQL database that stores Airflow metadata.
Airflow Cluster	Deploys the UI and scheduler components of Airflow.

In HPE Ezmeral Unified Analytics Software, there is only one instance of Airflow per cluster and Airflow DAGs are accessed by all authenticated users.

Airflow Components

Airflow consists of the following components:



Scheduler	Triggers the scheduled workflows and submits the tasks to an executor to run.
Executor	Executes the tasks or delegates the tasks to workers for execution.
Worker	Executes the tasks.
Web Server	Provides a user interface to analyze, schedule, monitor, and visualize the tasks and DAG. The Web Server enables you to manage users, roles, and set configuration options.
DAG Directory	Contains DAG files read by Scheduler, Executor, and Web Server.
Metadata Database	Stores the metadata about DAGs' state, runs, and Airflow configuration options.

Airflow Limitations

Airflow in HPE Ezmeral Unified Analytics Software has the following limitations:

- The CPU and memory resource limits for executors cannot be modified (CPU: 1, memory: 2Gi).
- To use the Spark Operator, you must provide the username by specifying it under the "username" key in the DAG Run Configuration.
- The logs of successfully run DAGs are available until the corresponding pods are deleted.

To learn more about Airflow, see Airflow Concepts.

More information

Financial Time Series Workflow on page 21
Describes how to use HPE Ezmeral Unified Analytics Software to run a Spark application from an Airflow DAG and then run a Jupyter notebook to analyze and visualize data that the Spark application puts into a shared directory in the shared volume that the data scientist's notebook is mounted to.

MNIST Digits Recognition Workflow on page 31
Provides an end-to-end workflow in HPE Ezmeral Unified Analytics Software for an MNIST digits recognition example.

Airflow DAGs Git Repository

Describes how HPE Ezmeral Unified Analytics Software reads DAGs and how to configure a GitHub repository in Airflow.

Airflow DAGs are pulled from the GitHub repository that you specify when you configure Airflow. HPE Ezmeral Unified Analytics Software supports both private and public GitHub repositories. HPE Ezmeral Unified Analytics Software can only read DAGs from a GitHub repository on a specified branch from a specified subdirectory. If the GitHub repository is located behind a proxy, you can configure a proxy for the GitHub repository in Airflow.

In an air-gapped environment where there is no pre-configured proxy to forward outgoing cluster connections to the internet, the installation of Airflow will not function properly. To resolve this issue, the administrator of the HPE Ezmeral Unified Analytics Software must either manually set up an HTTP proxy or configure Airflow with an internal Git repository.

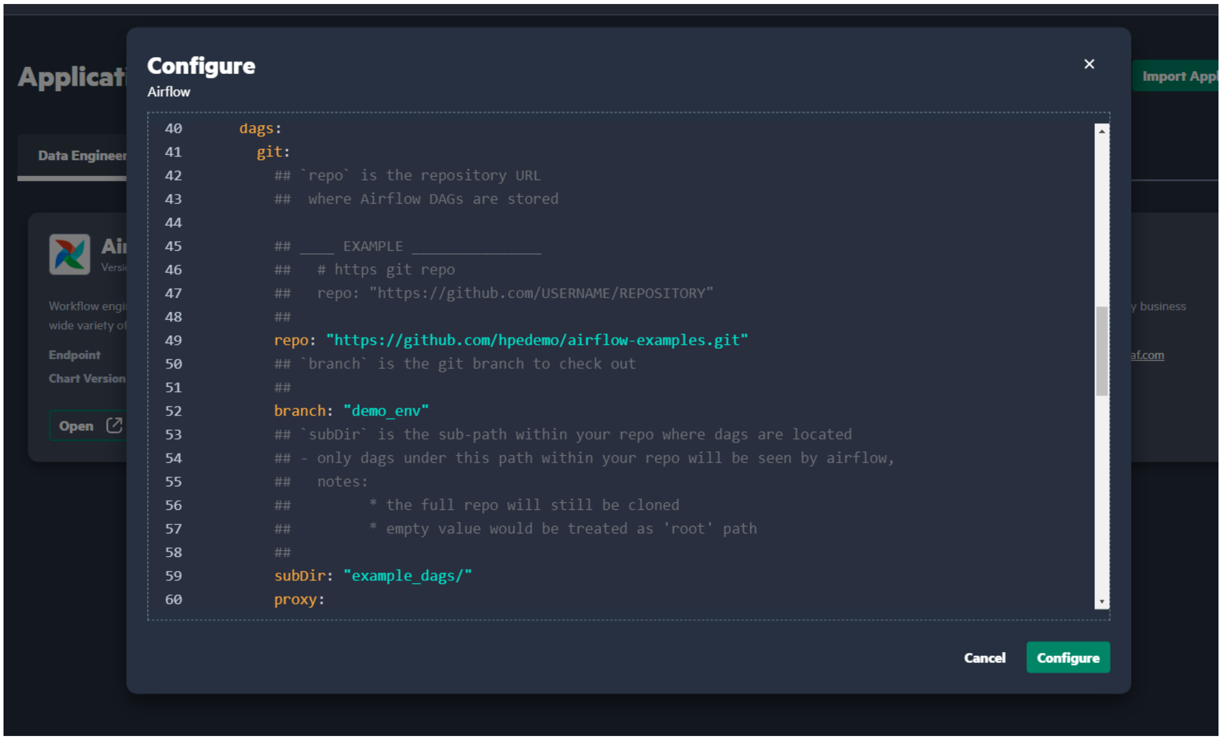
IMPORTANT: Best practice is to use Git submodules if multiple users have DAGs in their own repositories. To manage multiple users within the same GitHub repository, the HPE Ezmeral Unified Analytics Software administrator can create a root GitHub repository and then add all user GitHub repositories as submodules. As owner of the root GitHub repository, the administrator can update the Git submodules after users add/remove/modify files. For example, when a user modifies files, the user can ask the platform administrator to update the latest commit hash of the user's Git submodule in the root repository. For additional information, refer to GitHub - About code owners and Working with submodules.

Configuring a Git Repository for Airflow

To configure Airflow with the GitHub repository where DAGs are stored:

1. Sign in to HPE Ezmeral Unified Analytics Software as Administrator.
2. In the left navigation bar, click **Tools & Frameworks**.
3. Select the **Data Engineering** tab.
4. On the **Airflow** tile, click the **three-dots** menu and then select **Configure**. The YAML file editor opens.
5. In the editor, find the `git:` section.
6. Configure the following parameters in the `git:` section:

repo:	The repository URL for private or public Git repository which stores the DAGs. If you are using an air-gapped system without a proxy, specify your internal Git repository here.
branch:	The name of the branch within the repository to use.
subDir:	The path to the directory where the DAGs are located.



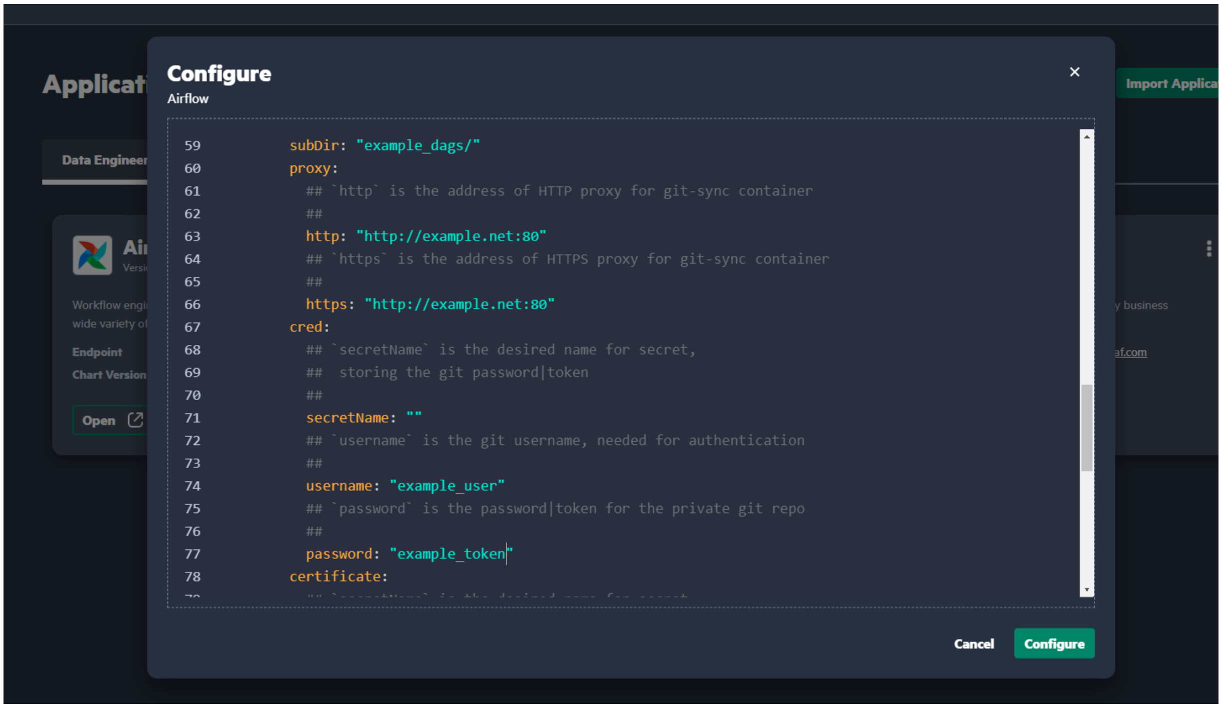
```
  dags:
    git:
      ## `repo` is the repository URL
      ##  where Airflow DAGs are stored

      ## ____ EXAMPLE _______________
      ##   # https git repo
      ##   repo: "https://github.com/USERNAME/REPOSITORY"
      ##
      repo: "https://github.com/hpedemo/airflow-examples.git"
      ## `branch` is the git branch to check out
      ##
      branch: "demo_env"
      ## `subDir` is the sub-path within your repo where dags are located
      ## - only dags under this path within your repo will be seen by airflow,
      ##   notes:
      ##         * the full repo will still be cloned
      ##         * empty value would be treated as 'root' path
      ##
      subDir: "example_dags/"
      proxy:
```

If you are using an air-gapped system, and Git cannot be accessed without a proxy, configure the following fields:

http	The address of HTTP proxy.
https	The address of HTTPS proxy.

If the git repository is private, configure the following fields:

username	The username of the user who has access to the private git repository.
password	The token or password of the user who has access to the private git repository. Alternatively, if you have created a secret in the `airflow-hpe` namespace under `key: 'password'` that contains the password or token information, you can specify the name of that secret in the `secretName` field under the `cred` section instead of using the `password` field directly.



```
        subDir: "example_dags/"
        proxy:
          ## `http` is the address of HTTP proxy for git-sync container
          ##
          http: "http://example.net:80"
          ## `https` is the address of HTTPS proxy for git-sync container
          ##
          https: "http://example.net:80"
        cred:
          ## `secretName` is the desired name for secret,
          ##  storing the git password|token
          ##
          secretName: ""
          ## `username` is the git username, needed for authentication
          ##
          username: "example_user"
          ## `password` is the password|token for the private git repo
          ##
          password: "example_token"
        certificate:
```

7. Click **Configure** and wait until Airflow is configured.

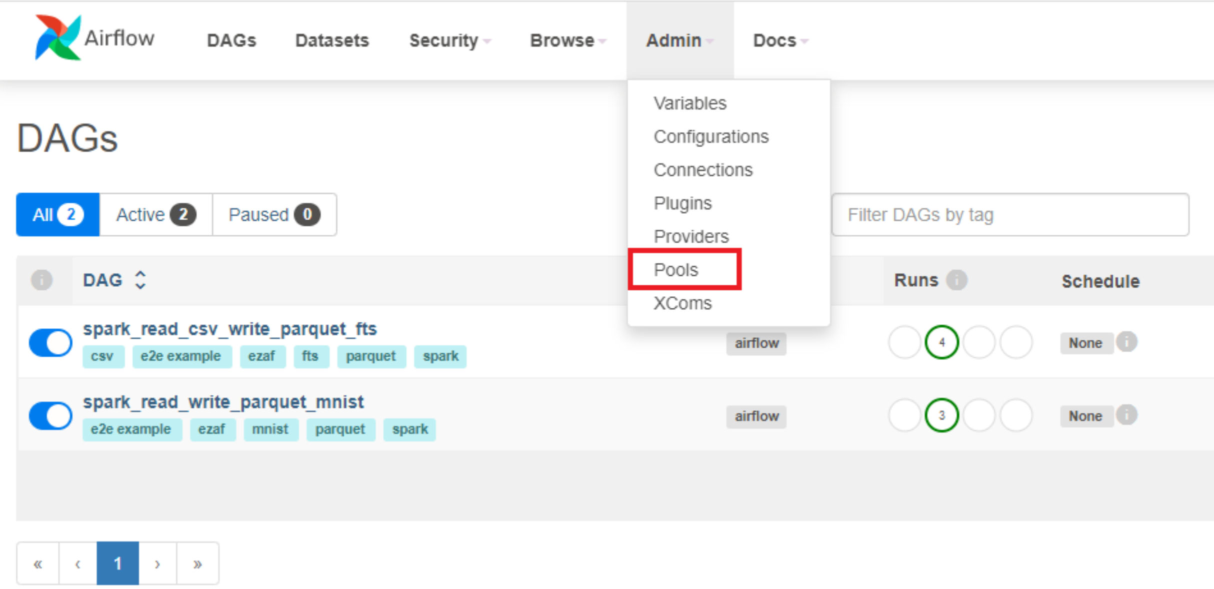
Configuring Airflow

Describes how to configure Airflow in HPE Ezmeral Unified Analytics Software.

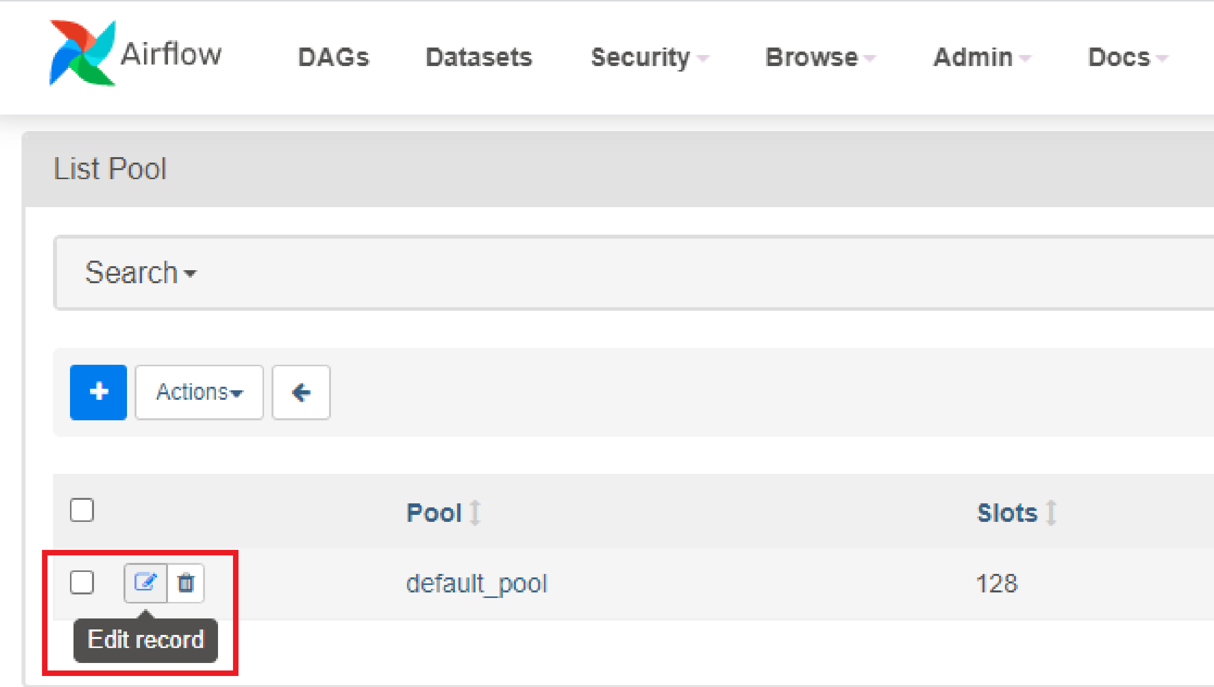
Modifying the Maximum Number of Simultaneous Jobs

To modify the maximum number of tasks from DAGs that can be run simultaneously, perform:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Airflow** tile under the **Data Engineering** tab and click **Open**.
3. Click **Admin** and select **Pools**.



4. Click **Edit Record** and update the value of **Slots**.



Pool	Slots
default_pool	128

5. Click **Save**.

Submitting Spark Applications by Using DAGs

Describes how to submit the Spark applications by using DAGs in Airflow.

Prerequisites

1. Prepare the DAG for your Spark application.
2. Add your DAG to the Git repository.

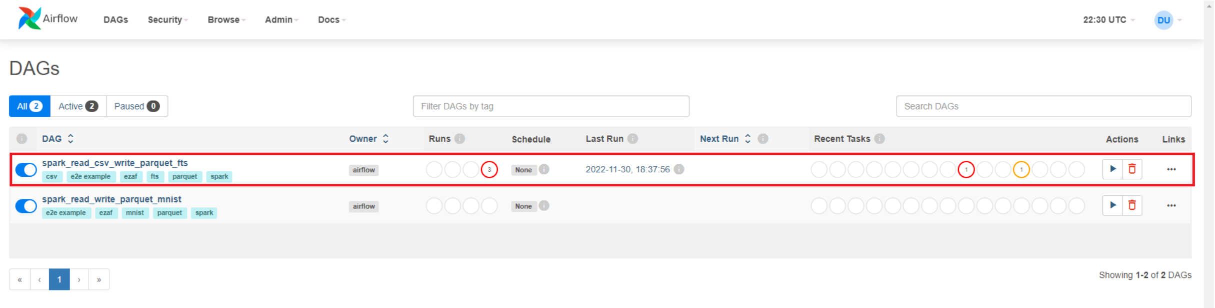
   NOTE: If you do not have the repository to store Airflow DAGs, request an administrator to configure the Git repository now. For details, see Airflow DAGs Git Repository on page 355.

3. After your DAG is available in the Git repository, sign in to HPE Ezmeral Unified Analytics Software as a member.

About this task

To run the DAG to submit the Spark applications in Airflow, follow these steps:

1. Navigate to the Airflow screen using either of the following methods:
   - Click **Data Engineering > Airflow Pipelines**.
   - Click **Tools & Frameworks**, select the **Data Engineering** tab, and click **Open** in the **Airflow** tile.
2. In **Airflow**, verify that you are on the **DAGs** screen.
3. Click `<your-spark-application>` DAG. For example:



DAGs

All 2 | Active 2 | Paused 0

Filter DAGs by tag

Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
spark_read_csv_write_parquet_fts csv e2e example ezaf fts parquet spark	airflow	3	None	2022-11-30, 18:37:56		1 1	▶ 🗑	⋯
spark_read_write_parquet_mnist e2e example ezaf mnist parquet spark	airflow		None				▶ 🗑	⋯

« ‹ 1 › »

Showing 1-2 of 2 DAGs

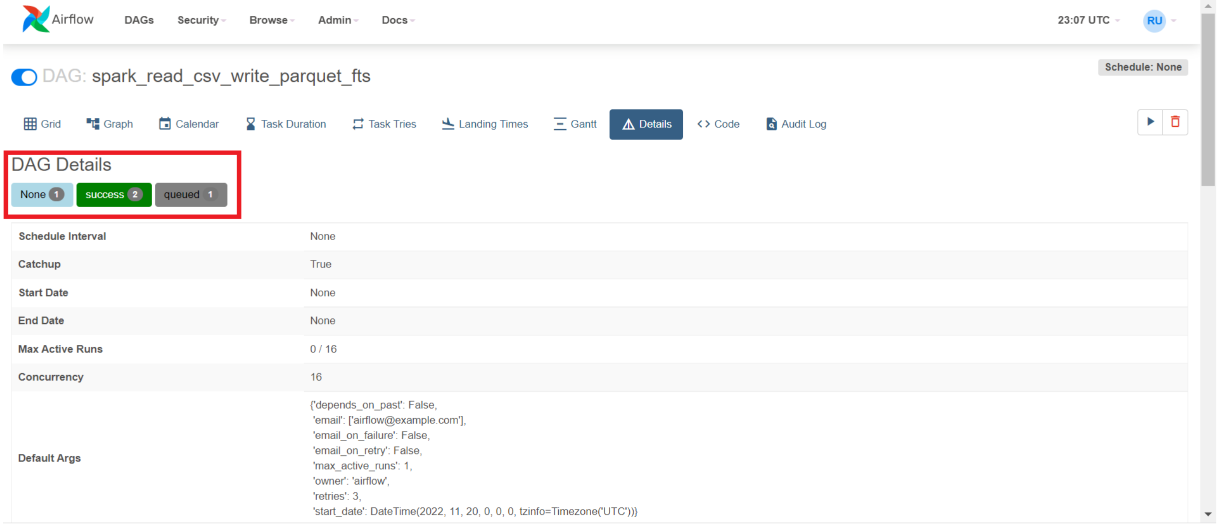
4. Click **Code** to view the DAG code.

5. Click **Graph** to view the graphical representation of the DAG.

6. Click**Run**(play button) and select**Trigger DAG w/ config**to specify the custom configuration.

7. To run a DAG after making configuration changes, click **Trigger.**

8. To view details for the DAG, click **Details**. Under **DAG Details**, you can see green, red, and/or yellow buttons with the number of times the DAG ran successfully or failed.



DAG: spark_read_csv_write_parquet_fts

Schedule: None

DAG Details

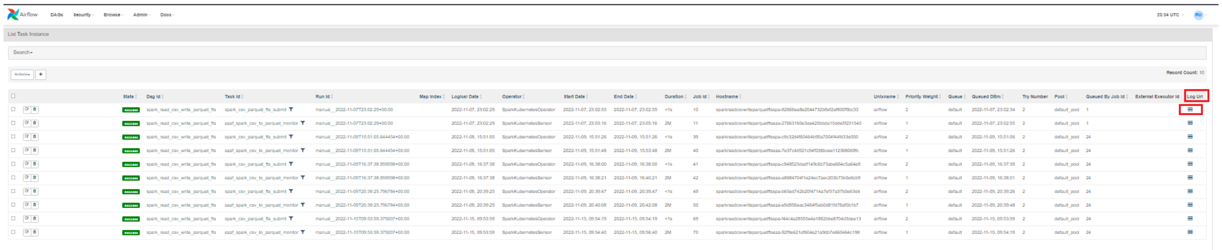
None 1 | success 2 | queued 1

Schedule Interval	None
Catchup	True
Start Date	None
End Date	None
Max Active Runs	0 / 16
Concurrency	16
Default Args	{'depends_on_past': False, 'email': ['airflow@example.com'], 'email_on_failure': False, 'email_on_retry': False, 'max_active_runs': 1, 'owner': 'airflow', 'retries': 3, 'start_date': DateTime(2022, 11, 20, 0, 0, 0, tzinfo=Timezone('UTC'))}

9. Click the **Success** button.

10. To find your job, sort by **End Date** to see the latest jobs that have run, and then scroll to the right and click the log icon under Log URL for that run. Note that jobs run with the configuration:

```
Conf "username":"your_username"
```



List Task Instance

Record Count: 10

State	Dag Id	Task Id	Run Id	Map Index	Logical Date	Operator	Start Date	End Date	Duration	Job Id	Hostname	Unixname	Priority Weight	Queue	Queued Dttm	Try Number	Pool	Queued By Job Id	External Executor Id	Log Url
success	spark_read_csv_write_parquet_fts	spark_csv_parquet_fts_submit	manual__2022-11-07T23:02:29+00:00		2022-11-07, 23:02:29	SparkKubernetesOperator	2022-11-07, 23:02:55	2022-11-07, 23:02:55	<1s	10	sparkreadcsvwriteparquetftsspa-82666ea8e2044732b9ef2af800f9bc33	airflow	2	default	2022-11-07, 23:02:34	2	default_pool	1		≡
success	spark_read_csv_write_parquet_fts	ezaf_spark_csv_to_parquet_monitor	manual__2022-11-07T23:02:29+00:00		2022-11-07, 23:02:29	SparkKubernetesSensor	2022-11-07, 23:03:16	2022-11-07, 23:05:16	2M	11	sparkreadcsvwriteparquetftseza-27663160e3ae420bbda15dde5f231340	airflow	1	default	2022-11-07, 23:02:55	2	default_pool	1		≡
success	spark_read_csv_write_parquet_fts	spark_csv_parquet_fts_submit	manual__2022-11-09T15:51:05.644454+00:00		2022-11-09, 15:51:05	SparkKubernetesOperator	2022-11-09, 15:51:26	2022-11-09, 15:51:26	<1s	39	sparkreadcsvwriteparquetftsspa-c9c32d4f60464b95a7004f44fd33e550	airflow	2	default	2022-11-09, 15:51:06	2	default_pool	24		≡
success	spark_read_csv_write_parquet_fts	ezaf_spark_csv_to_parquet_monitor	manual__2022-11-09T15:51:05.644454+00:00		2022-11-09, 15:51:05	SparkKubernetesSensor	2022-11-09, 15:51:48	2022-11-09, 15:53:48	2M	40	sparkreadcsvwriteparquetftseza-7e37c4d921c94f038bcee112368069fc	airflow	1	default	2022-11-09, 15:51:26	2	default_pool	24		≡
success	spark_read_csv_write_parquet_fts	spark_csv_parquet_fts_submit	manual__2022-11-09T16:37:38.959598+00:00		2022-11-09, 16:37:38	SparkKubernetesOperator	2022-11-09, 16:38:00	2022-11-09, 16:38:00	<1s	41	sparkreadcsvwriteparquetftsspa-c948523dadf149c6b73abe664c5a64e9	airflow	2	default	2022-11-09, 16:37:39	2	default_pool	24		≡
success	spark_read_csv_write_parquet_fts	ezaf_spark_csv_to_parquet_monitor	manual__2022-11-09T16:37:38.959598+00:00		2022-11-09, 16:37:38	SparkKubernetesSensor	2022-11-09, 16:38:21	2022-11-09, 16:40:21	2M	42	sparkreadcsvwriteparquetftseza-a8684704f1e24ec7aec203b73b0e6cb9	airflow	1	default	2022-11-09, 16:38:01	2	default_pool	24		≡
success	spark_read_csv_write_parquet_fts	spark_csv_parquet_fts_submit	manual__2022-11-09T20:39:25.796784+00:00		2022-11-09, 20:39:25	SparkKubernetesOperator	2022-11-09, 20:39:47	2022-11-09, 20:39:47	<1s	49	sparkreadcsvwriteparquetftsspa-b65ed742b20f4714a7ef37a37b0e63d4	airflow	2	default	2022-11-09, 20:39:26	2	default_pool	24		≡
success	spark_read_csv_write_parquet_fts	ezaf_spark_csv_to_parquet_monitor	manual__2022-11-09T20:39:25.796784+00:00		2022-11-09, 20:39:25	SparkKubernetesSensor	2022-11-09, 20:40:08	2022-11-09, 20:42:08	2M	50	sparkreadcsvwriteparquetftseza-e9d956eac3484f5ab0d811d78af0b1b7	airflow	1	default	2022-11-09, 20:39:48	2	default_pool	24		≡
success	spark_read_csv_write_parquet_fts	spark_csv_parquet_fts_submit	manual__2022-11-15T09:53:59.379207+00:00		2022-11-15, 09:53:59	SparkKubernetesOperator	2022-11-15, 09:54:19	2022-11-15, 09:54:19	<1s	69	sparkreadcsvwriteparquetftsspa-f44c4a28555e4e1882bbe8704d3dee13	airflow	2	default	2022-11-15, 09:53:59	2	default_pool	24		≡
success	spark_read_csv_write_parquet_fts	ezaf_spark_csv_to_parquet_monitor	manual__2022-11-15T09:53:59.379207+00:00		2022-11-15, 09:53:59	SparkKubernetesSensor	2022-11-15, 09:54:40	2022-11-15, 09:56:40	2M	70	sparkreadcsvwriteparquetftseza-92f9e621af604e21a9db7e660464c198	airflow	1	default	2022-11-15, 09:54:19	2	default_pool	24		≡

When running Spark applications using Airflow, you can see the logs.

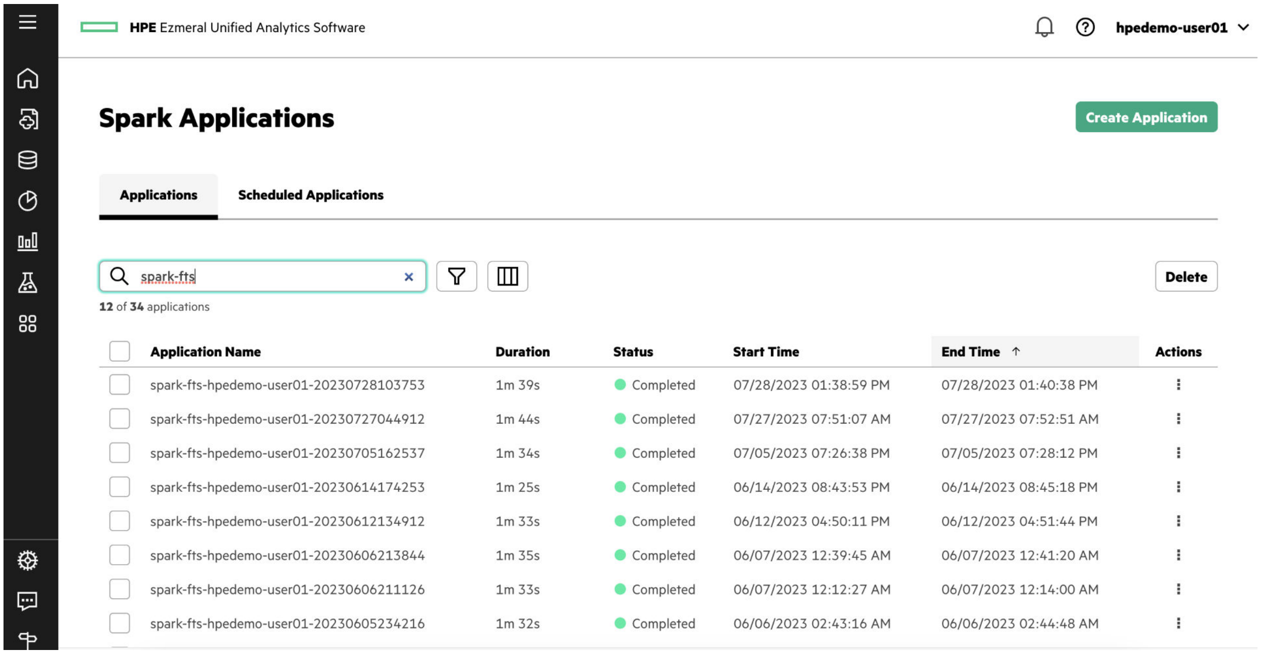
IMPORTANT: The cluster clears the logs that result from the DAG runs. The duration after which the cluster clears the logs depends on the Airflow task, cluster configuration, and policy.

Results

Once you have triggered the DAG, you can view the Spark application in the **Spark Applications** screen.

To view the Spark application, go to **Analytics > Spark Applications**.

Alternatively, you can go to **Tools & Frameworks** and then click on the **Analytics** tab. On the **Analytics** tab, select the **Spark** tile and click **Open**.



Spark Applications

spark-fts

12 of 34 applications

□	Application Name	Duration	Status	Start Time	End Time ↑	Actions
□	spark-fts-hpedemo-user01-20230728103753	1m 39s	● Completed	07/28/2023 01:38:59 PM	07/28/2023 01:40:38 PM	⋮
□	spark-fts-hpedemo-user01-20230727044912	1m 44s	● Completed	07/27/2023 07:51:07 AM	07/27/2023 07:52:51 AM	⋮
□	spark-fts-hpedemo-user01-20230705162537	1m 34s	● Completed	07/05/2023 07:26:38 PM	07/05/2023 07:28:12 PM	⋮
□	spark-fts-hpedemo-user01-20230614174253	1m 25s	● Completed	06/14/2023 08:43:53 PM	06/14/2023 08:45:18 PM	⋮
□	spark-fts-hpedemo-user01-20230612134912	1m 33s	● Completed	06/12/2023 04:50:11 PM	06/12/2023 04:51:44 PM	⋮
□	spark-fts-hpedemo-user01-20230606213844	1m 35s	● Completed	06/07/2023 12:39:45 AM	06/07/2023 12:41:20 AM	⋮
□	spark-fts-hpedemo-user01-20230606211126	1m 33s	● Completed	06/07/2023 12:12:27 AM	06/07/2023 12:14:00 AM	⋮
□	spark-fts-hpedemo-user01-20230605234216	1m 32s	● Completed	06/06/2023 02:43:16 AM	06/06/2023 02:44:48 AM	⋮

Example

- Financial Time Series Workflow on page 21
- MNIST Digits Recognition Workflow on page 31

Defining RBACs on DAGs

Describes role-based access controls (RBACs) with respect to Airflow in HPE Ezmeral Unified Analytics Software and how to define RBACs to permit access to DAGs.

Role-based access controls (RBACs) are an authorization system based on policies, user roles, and bindings between the roles and policies that protect resources. With the introduction of RBACs, HPE Ezmeral Unified Analytics Software users (admins and members) can grant users access to their DAGs through access controls that they define in the DAG constructors.

Admin Role

The following list describes DAG access for admins and the admin-related tasks that impact user access to DAGs:

- HPE Ezmeral Unified Analytics Software admins have full access to all Airflow DAGs regardless of the access controls set.

- Admins can assign a member the *admin* role in HPE Ezmeral Unified Analytics Software to give the user full access to DAGs; however, this action must occur before the user signs in to the HPE Ezmeral Unified Analytics Software UI and accesses Airflow. See User Roles.
- If an admin removes a user from HPE Ezmeral Unified Analytics Software, that user's access to Airflow is automatically revoked. Other users can no longer access the DAGs that the removed user shared.

  CAUTION: HPE only supports user role changes made through the HPE Ezmeral Unified Analytics Software UI. Role changes made in HPE Ezmeral Unified Analytics Software are automatically propagated to Airflow. HPE does not support role changes made directly in Airflow because the changes do not propagate back to HPE Ezmeral Unified Analytics Software, which can cause unexpected system behaviors.

  TIP: Best practice is to use Git submodules if multiple users have DAGs in their own repositories. To manage multiple users within the same GitHub repository, the HPE Ezmeral Unified Analytics Software platform administrator can create a root GitHub repository and then add all user GitHub repositories as submodules. As owner of the root GitHub repository, the platform administrator can update the Git submodules after users add, remove, or modify files. For example, when a user modifies files, the user can ask the platform administrator to update the latest commit hash of the user's Git submodule in the root repository. For additional information, refer to GitHub - About code owners and Working with submodules.

Member Role

The following list describes DAG access for members:

- Members can access DAGs:
  - When DAGs do not have any access controls defined.
  - When permitted to do so through access controls (either defined on their username or defined through the All user role).
- Members can define access controls on the DAGs they create.

Supported Access Controls

The following table lists and describes the access controls that admins and users can define in the DAG constructor, as well as the associated access control values to use when configuring the access controls on a user in the DAG constructor.

Access Control Type	Access Control Value	Description
Read	can_read	The specified user can see the source code but cannot launch the DAG.
Edit	can_edit	The specified user can launch the DAG and add some notes.
Delete	can_delete	The specified user can delete the DAG; however, DAGs repopulate in the GitHub repository every few seconds.

Define the access controls on a username through the `access_control` parameter in the DAG constructor, as shown in the following example for `user01`:

```
access_control={
        'role_user01': {
            'can_read',
            'can_edit',
            'can_delete'
        }
    }
```

If you want to grant *all* users access to a DAG, define access controls on `All` instead of a specific username, as shown in the following example:

```
access_control={
        'All': {
            'can_read',
            'can_edit',
            'can_delete'
        }
    }
```

Defining RBACs on Users

You can define access controls on a user (username) that exists or does not yet exist in HPE Ezmeral Unified Analytics Software. Adding a user to HPE Ezmeral Unified Analytics Software after you define roles on the user (username) in the DAG constructor will not cause any issues between the systems. An HPE Ezmeral Unified Analytics Software admin can add or create the user. See Adding and Removing Users on page 233.

IMPORTANT: The DAG must exist in the GitHub repository or a Git submodule that the Airflow instance in HPE Ezmeral Unified Analytics Software points to.

To define access controls on a user in the DAG constructor:

1. Go to the GitHub repository and add the following `access_control` parameters and values to the DAG constructor, as shown in the following example:

```
}
with DAG{
    dag_id='example_kubernetes_operator',
    dafault_args=default_args,
    schedule_interval=None,
    tags=['example'],
    access_control={
        'role_<username>': {
            'can_read',
            'can_edit',
            'can_delete'
        }
    }
} as dag:
```

TIP:

- If you commit a DAG without the `access_control` annotation, all users (admins and members) can view and access the DAG.
- Only include the access role(s) that you want the user to have. For example, if you do not want the user to launch the DAG, do not assign the user the `can_edit` access control.

2. Commit and push the changes to the DAG.

Viewing Access Controls on Users

HPE Ezmeral Unified Analytics Software admins can go to the **Security** page in Airflow to view access controls on users. Members cannot access the **Security** page.

To view access controls on users:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Tools and Frameworks**.
3. On the **Data Engineering** tab, click **Open** in the **Airflow** tile.
4. In **Airflow**, click the **Security** tab and select **List Roles**.

Using whylogs with Airflow

Note: This feature is presented as a developer preview. Developer previews are not tested for production environments, and should be used with caution.

Describes how to use whylogs with Airflow DAGs.

Prerequisites

Sign in to HPE Ezmeral Unified Analytics Software as a member.

About this task

In HPE Ezmeral Unified Analytics Software, whylogs is integrated to work with Airflow DAGs. You can use whylogs with Airflow to profile and monitor the data and detect drifts as data flows through the data pipelines.

To use whylogs with Airflow DAGs, refer to the Airflow DAG example in GitHub. The basic steps are outlined as follows:

1. Import the required libraries and modules from whylogs in your Airflow DAG script. You can use notebooks to create your Airflow DAG. To learn about notebooks, see Creating and Managing Notebook Servers on page 436.
2. Define your Airflow DAG that can profile and monitor the data to detect drifts.
3. Add your DAG to the Git repository.

NOTE: If you do not have the repository to store Airflow DAGs, request an administrator to configure the Git repository now. For details, see Airflow DAGs Git Repository on page 355.

4. Navigate to the **Airflow** screen using either of the following methods:
   - Click **Data Engineering > Airflow Pipelines**.
   - Click **Tools & Frameworks**, select the **Data Engineering** tab, and click **Open** in the **Airflow** tile.
5. In **Airflow**, verify that you are on the **DAGs** screen and your defined DAG is available in the **DAGs** screen.
6. To run your DAG, click the **play** button.
7. Once your DAG run completes, navigate back to the HPE Ezmeral Unified Analytics Software home screen.
8. In the left navigation bar, go to **Data Engineering > Data Sources**.
9. Click **Browse**.
10. Go to the `/shared/<airflow-whylogs>` folder which is a path set in your DAG to store the logs from whylogs. You can see that the data profiles and the drift summary report are stored in the shared volume in the `.html` and `.bin` formats.
11. To download a summary report, select **Download** from the **Actions** menu.

Results

You can analyze the summary report to detect drifts and monitor your data.

Superset

Provides a brief overview of Superset in HPE Ezmeral Unified Analytics Software.

Superset is a cloud-native business intelligence web application that collects and processes large volumes of data that can be used in the data visualizations and dashboards that you create within it. Superset is accessible in HPE Ezmeral Unified Analytics Software by going to **BI Reporting > Dashboards** or **Tools & Frameworks > Data Engineering** in the left navigation panel.

When you connect HPE Ezmeral Unified Analytics Software to various data sources, you can access the data in those data sources from Superset. For example, you can create any type of chart in Superset and specify query conditions on a selected data set to visualize the query results in the chart. Superset works with EzPresto on page 294, the HPE Ezmeral Unified Analytics Software accelerated SQL query engine, to process the query and display results in the chart. You can then add the chart to a dashboard and continue this process to build out a dashboard that visualizes your analytical workloads.

Underlying Superset and EzPresto is a Presto database that unifies the data sources connected to HPE Ezmeral Unified Analytics Software. The unified data source connection enables you to:

- Add the data sets you create in the HPE Ezmeral Unified Analytics Software Data Engineering space to Superset.
- Connect Superset directly to the Presto database for direct access to the unified data sources.

You can also connect Superset to external databases (those that are not part of the unified data source connection in HPE Ezmeral Unified Analytics Software).

Refer to the following tutorials to get started with Superset in HPE Ezmeral Unified Analytics Software:

- BI Reporting (Superset) Basics on page 14
- Retail Store Analysis Dashboard (Superset) on page 43

For additional information about Superset, see Apache Superset and EzPresto on page 294.

Defining RBACs in Superset

Describes role-based access controls (RBACs) with respect to Superset in HPE Ezmeral Unified Analytics Software and how to define RBACs to permit access to Superset dashboards.

Role-based access controls (RBACs) are an authorization system based on policies, user roles, and bindings between the roles and policies that protect resources. With the introduction of RBAC, HPE Ezmeral Unified Analytics Software maps the HPE Ezmeral Unified Analytics Software admin and member roles to Superset Admin and Alpha roles respectively.

The following user role mapping is defined in the Superset HELM chart (YAML file):

TIP: You cannot edit the role mappings in the HELM chart.

User Type	Mapping Parameter
Admin	AUH_ROLE_ADMIN = 'Admin'
Member	AUTH_USER_REGISTRATION_ROLE = "Alpha"

Admin Role (Admin)

The following list describes admin access and the admin-related tasks that impact users in Superset:

- Admins can edit (add or remove) roles in the Superset UI.
- Admins can change a member's role in HPE Ezmeral Unified Analytics Software to *admin*.
- Admins can view all user activity and data, including all dashboards created by all users, as well as all of the data in the dashboards.
- Admins can access the security settings in Superset, such as viewing user profiles, including user roles and access controls.
- Admins can edit a user in Superset and change the user's roles.

Member Role (Alpha)

The following list describes Superset access for members (Alpha):

- Members can create their own database connections in Superset.
- Members can view charts and datasets created by other users, but cannot view dashboards unless explicitly permitted to do so.
- Members can access dashboards they create (as owner) and dashboards that other users have shared with them (added to dashboard owner list).

  NOTE: Access to a dashboard does not grant access to data. The user must have permission on the data itself to view the data in a dashboard. If the user does not have access to certain data, that data does not display in their view of the dashboard.

- Members cannot see the Superset security settings, such as user roles and access permissions.

⚠ **CAUTION:** HPE only supports user role changes made through the HPE Ezmeral Unified Analytics Software UI. Role changes made in HPE Ezmeral Unified Analytics Software are automatically propagated to Superset. HPE does not support role changes made directly in Superset because the changes do not propagate back to HPE Ezmeral Unified Analytics Software, which can cause unexpected system behaviors.

System and Application Notes

Note the following system and application behaviors:

- Users (members and admins) do not appear in the Superset user list until they sign in to Superset. If a user has not signed in to Superset, other users cannot share anything with that user, such as dashboards.
- When a user is removed from the HPE Ezmeral Unified Analytics Software platform, the user's Superset profile remains. Apache Superset recommends deactivating the user instead of removing the user from Superset.
- If a user was removed and then added back to the HPE Ezmeral Unified Analytics Software platform (registered with the same username and email), the user's Superset access is automatically restored to the user's original Superset profile and all related resources.

Supported Access Controls

HPE Ezmeral Unified Analytics Software supports the following access controls in Superset:

- Admin
- Public
- Alpha
- Gamma
- granter
- sql_lab

Sharing Dashboards

To share a dashboard:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, go to **Tools & Frameworks**.
3. On the **Data Engineering** tab, click **Open** in the **Superset** tile.
4. Click the **Dashboards** tab.
5. In the **Actions** column of the dashboard you want to share, select **Edit** (pencil icon).

6. Under **Access**, click into the field and select the roles you want to assign the user. Alternatively, you can also remove roles from the user.

Viewing Role Descriptions

To see the access a role permits:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, go to **Tools & Frameworks**.
3. On the **Data Engineering** tab, click **Open** in the **Superset** tile.
4. Go to **Settings** and select **List Roles**.
5. Click **Show record** (magnifying glass icon) next to a role to view the role description.

Viewing and Editing Access Controls on Users

To view the access controls on a user or edit access controls on a user:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, go to **Tools & Frameworks**.
3. On the **Data Engineering** tab, click **Open** in the **Superset** tile.
4. Go to **Settings** and select **List Users**.
5. (Optional) To edit the user's role(s), click the edit icon next to the username and then add or remove roles using the dropdown menu in the **Role** field.

Configuring Horizontal Pod Autoscaling (HPA)

Describes how to configure HPA for Superset in HPE Ezmeral Unified Analytics Software.

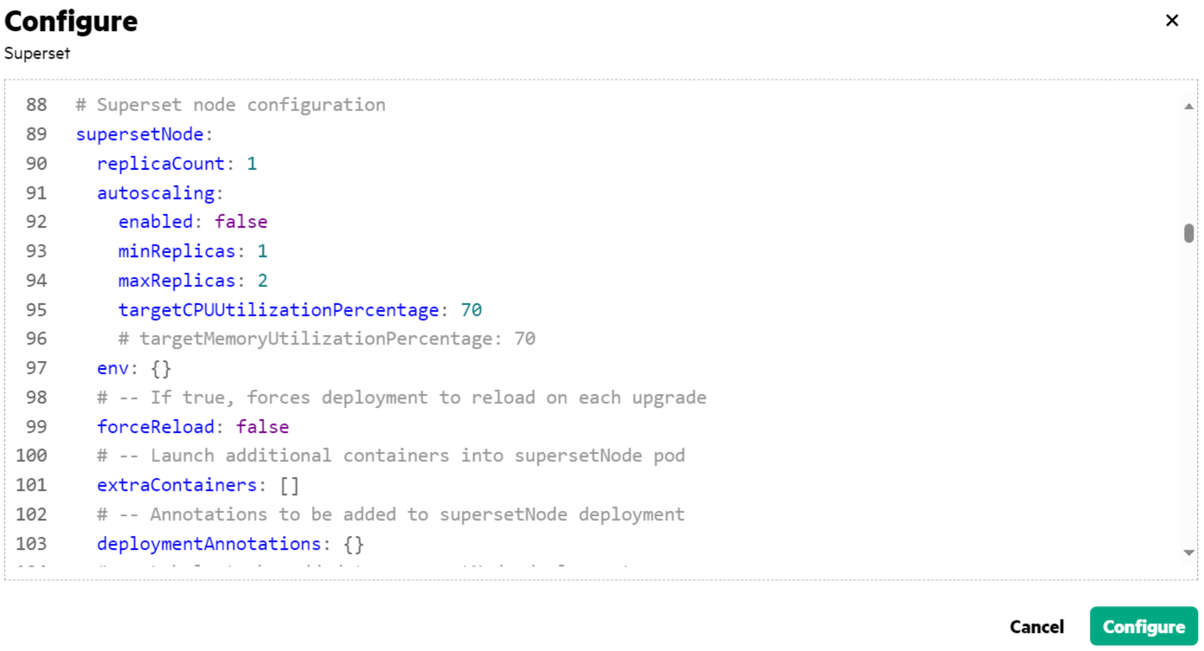
HPE Ezmeral Unified Analytics Software supports autoscaling `supersetNode` and `supersetWorker` using Horizontal Pod Autoscaling (HPA) with CPU and MEM metrics. With HPA configuration, you can perform the following:

- Enable or disable autoscaling.
- Set the minimum and maximum number of replicas.
- Set the target CPU and memory utilization percentage for autoscaling.

By default, autoscaling is disabled for `supersetNode` and enabled for `supersetWorker`.

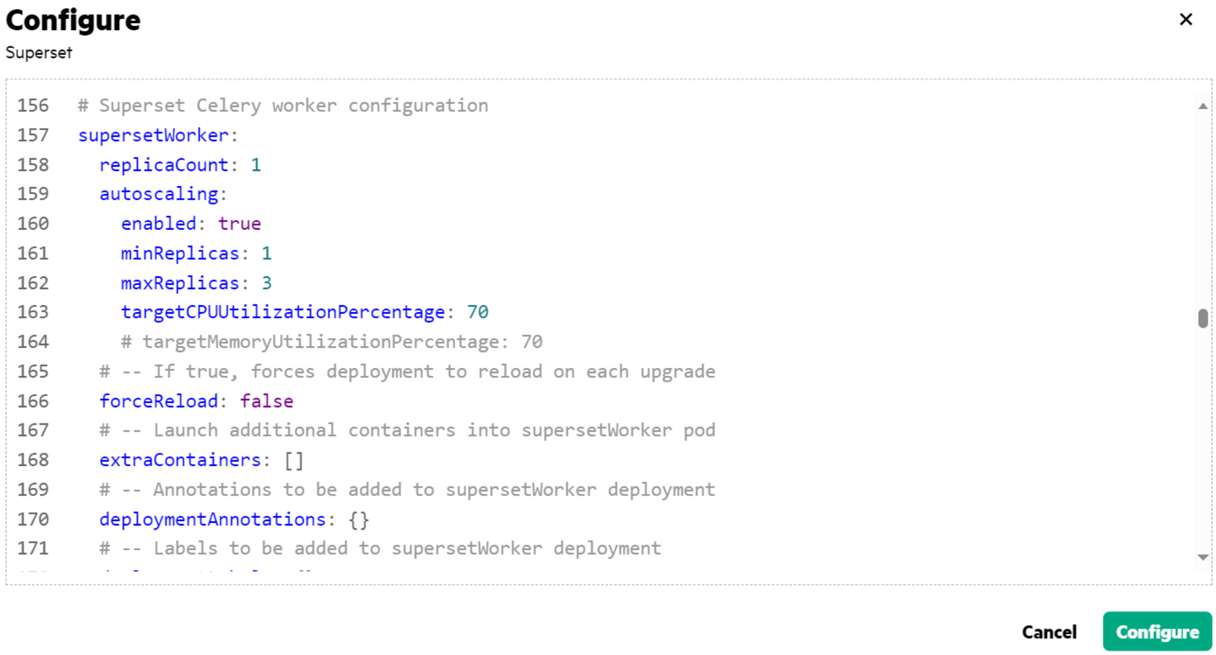
For example,

- The HPA configuration with autoscaling disabled for `supersetNode`:



```yaml
# Superset node configuration
supersetNode:
  replicaCount: 1
  autoscaling:
    enabled: false
    minReplicas: 1
    maxReplicas: 2
    targetCPUUtilizationPercentage: 70
    # targetMemoryUtilizationPercentage: 70
  env: {}
  # -- If true, forces deployment to reload on each upgrade
  forceReload: false
  # -- Launch additional containers into supersetNode pod
  extraContainers: []
  # -- Annotations to be added to supersetNode deployment
  deploymentAnnotations: {}
```

- The HPA configuration with autoscaling enabled for `supersetWorker`:



```yaml
# Superset Celery worker configuration
supersetWorker:
  replicaCount: 1
  autoscaling:
    enabled: true
    minReplicas: 1
    maxReplicas: 3
    targetCPUUtilizationPercentage: 70
    # targetMemoryUtilizationPercentage: 70
  # -- If true, forces deployment to reload on each upgrade
  forceReload: false
  # -- Launch additional containers into supersetWorker pod
  extraContainers: []
  # -- Annotations to be added to supersetWorker deployment
  deploymentAnnotations: {}
  # -- Labels to be added to supersetWorker deployment
```

Data Analytics

Provides a brief overview of data analytics in HPE Ezmeral Unified Analytics Software.

HPE Ezmeral Unified Analytics Software provides a single place where data engineers and data scientists can run analytical workloads through the Apache Spark Operator, interactive sessions in Apache Livy, and schedule jobs using Apache Airflow.

ACID (Atomicity, Consistency, Isolation and Durability) transactions for Spark applications are supported out of box with Delta Lake. Delta Lake has a well-defined open protocol called Delta Transaction Protocol that provides ACID transactions to Apache Spark applications. 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.

HPE Ezmeral Unified Analytics Software simplifies data access and data workflows and pipelines. HPE Ezmeral Unified Analytics Software connects to multiple types of internal and external data sources that you can easily explore with federated SQL queries that you visualize in Superset (dashboards). You can also use Spark to transform raw data sets into consumable formats like data lakehouses.

More information

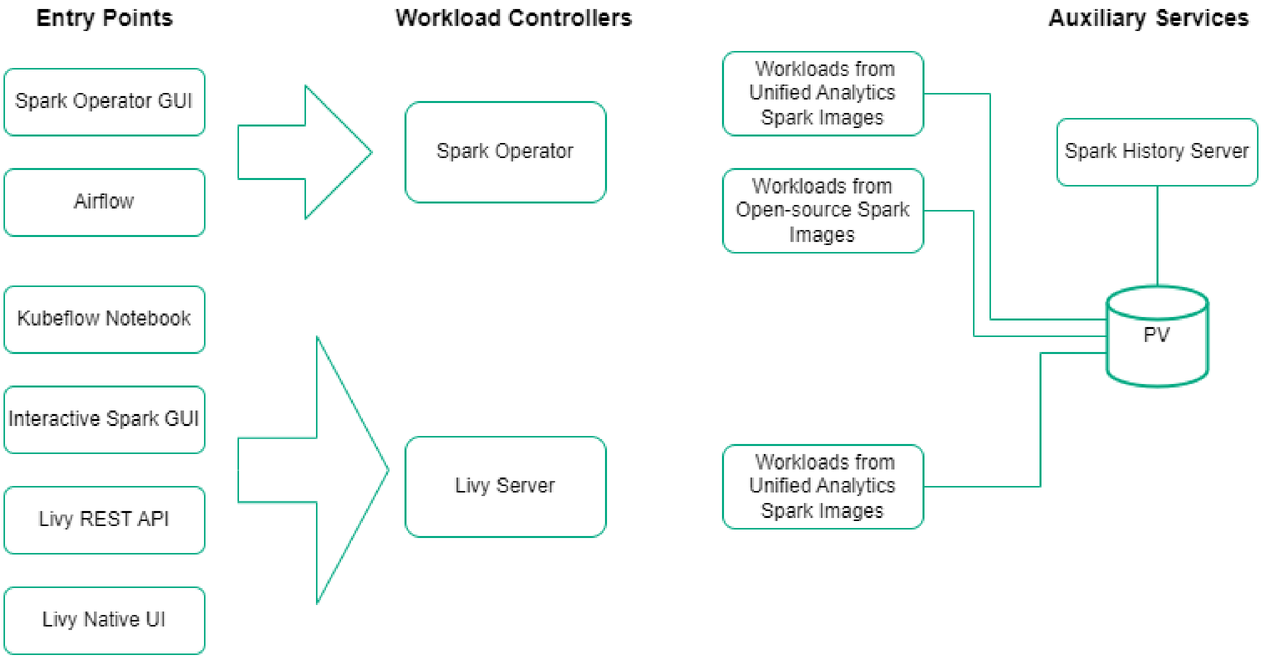
Get Started on page 6
Describes how to get started with HPE Ezmeral Unified Analytics Software.

Spark

Provides a brief overview of Apache Spark in HPE Ezmeral Unified Analytics Software.

Spark is a unified analytics engine with high data processing speed that offers high-level APIs in Java, Scala, Python, and R. Spark provides the in-memory computing and optimized query execution for fast data processing.

In HPE Ezmeral Unified Analytics Software, there are two controllers for running Spark workloads. These controllers are Spark Operator and Livy server.



HPE Ezmeral Unified Analytics Software supports multi-version Spark Operator. You can submit Spark Applications for different versions of Apache Spark using a single Spark Operator.

You can choose to use one of the supported Spark images to submit your Spark application using the Spark Operator workflow. See Using Spark Images on page 370.

To see the list of the Spark images distributed by HPE Ezmeral Unified Analytics Software, see List of Spark Images on page 377.

Livy server uses the Rest API and Spark images (supporting Data Fabric services) provided by HPE Ezmeral Unified Analytics Software to submit the Spark applications. To learn about the supported version of Spark, see Support Matrix on page 201.

NOTE: Livy does not support Spark OSS images or your own open-source Spark images on HPE Ezmeral Unified Analytics Software.

Features and Functionality

HPE Ezmeral Unified Analytics Software provides an enterprise-ready, unified Spark experience that supports an Apache Livy-based interactive sessions..

Spark in HPE Ezmeral Unified Analytics Software supports the following features and functionality:

- ACID transactions for Spark applications with Delta Lake.
- Details for both Spark applications and Livy sessions are stored in Spark History Server. See Spark History Server on page 398.
- Run Spark jobs from HPE Ezmeral Unified Analytics Software using the following components:
  - Spark Operator: The following are entry points for the Spark Operator:
    - Airflow
    - Spark Operator GUI in HPE Ezmeral Unified Analytics Software. See Creating Spark Applications on page 379.
  - Livy Server: The following are entry points for the Livy server:
    - Kubeflow Notebook: You can use Spark Magics to run Livy sessions using Kubeflow notebooks. See Notebook Magic Functions on page 445.
    - Interactive Spark Sessions GUI available in HPE Ezmeral Unified Analytics Software. See Creating Interactive Sessions on page 394.
    - Livy REST API (with basic authentication).
    - Livy native UI (with platform SSO authentication): You can use the Livy native UI to troubleshoot such as checking the state of the session or state of statements. You cannot submit Spark applications using the Livy native UI.
- Spark applications and Livy sessions are preconfigured in such a way that both `user` and `shared` volumes are mounted to driver and executor runtimes and you can use these folders to pass files into Spark runtime when using the HPE Ezmeral Unified Analytics Software GUI. However, `user` and `shared` volumes are not mounted to driver and executor runtimes when using the Livy REST API to create Livy sessions.
- Dynamically set user context to prevent impersonation calls for better security.

Related tasks

Submitting a Spark Wordcount Application on page 60
Provides an end-to-end example for creating and submitting a wordcount Spark Application in HPE Ezmeral Unified Analytics Software.

More information

Financial Time Series Workflow on page 21
Describes how to use HPE Ezmeral Unified Analytics Software to run a Spark application from an Airflow DAG and then run a Jupyter notebook to analyze and visualize data that the Spark application puts into a shared directory in the shared volume that the data scientist’s notebook is mounted to.

MNIST Digits Recognition Workflow on page 31
Provides an end-to-end workflow in HPE Ezmeral Unified Analytics Software for an MNIST digits recognition example.

Using Spark Images

Describes different types of Spark images supported by HPE Ezmeral Unified Analytics Software.

HPE Ezmeral Unified Analytics Software packages two different types of images:

1. HPE-curated Spark images. For details, see Using HPE-Curated Spark Images on page 371.
2. Spark Open-Source Software (OSS) images (Spark OSS images). For details, see Using Spark OSS Images on page 373.

The following table compares the two different types of Spark images packaged with HPE Ezmeral Unified Analytics Software.

Capabilities	HPE-Curated Spark Images	Spark OSS Images
Packaged by HPE	Yes	Yes
Data Fabric (Filesystem, Database, Streams)	Yes	No
Data Fabric Security (data-fabric SASL (`maprsasl`))	Yes	No
Workloads from Spark Operator	Yes	Yes
Workloads from Livy	Yes	No

However, you can also bring your own open-source Spark images compatible with the Kubernetes version supported on HPE Ezmeral Unified Analytics Software. See Using Your Own Open-Source Spark Images on page 374.

Using HPE-Curated Spark Images

Describes how to use HPE-curated Spark images to submit Spark applications.

HPE-Curated Spark images are Apache Spark images that are customized to support Data Fabric filesystem, Data Fabric Streams, or any other Data Fabric sources and sinks that require a Data Fabric client. These Spark images also support Data Fabric-specific security features (data-fabric SASL (`maprsasl`)).

HPE-curated Spark images are the images used by GUI for default experience. See List of Spark Images on page 377.

You can use HPE-curated Spark images with four different workflows as follows:

- Spark Operator workflow using the **Create Spark Application** GUI. See Using the Create Spark Application GUI on page 371.
- Spark Operator workflow using Airflow. See Using Airflow on page 372.
- Livy workflow using the **Spark Interactive Sessions** GUI. See Using the Spark Interactive Sessions GUI on page 372.
- Livy workflow using Jupyter Notebooks. See Using Notebooks on page 372.

Using the Create Spark Application GUI

To use HPE-curated Spark images, choose one of the following options in the GUI:

Using New application	If you choose the **New application** option in the **Application Details** step of the **Create Spark Application** wizard, your Spark application will be configured with HPE-curated Spark image. The List of Spark Images on page 377 page also lists the default HPE-curated Spark images used for GUI experience.

Using Upload YAML	If you choose the **Upload YAML** option, your Spark application will be configured with your chosen Spark image on your YAML file.
	`image: <base-repository>/`
	`<image-name>:<image-tag>`

To learn about how to submit Spark applications by using GUI, see Creating Spark Applications on page 379.

Using the Spark Interactive Sessions GUI

To use HPE-curated Spark images when using the Spark Interactive Sessions, follow these steps:

1. Perform the creating interactive sessions instructions until you reach the **Spark Configurations** box in the **Session Configurations and Dependencies** step. See Creating Interactive Sessions on page 394.

2. In the **Spark Configurations** box, you have two options:

   - If you leave the Key and Value boxes empty, the Spark interactive sessions will be created with the HPE-curated Spark image. The List of Spark Images on page 377 page also lists the default HPE-curated Spark images used for GUI experience.

   - If you set the Key and Value boxes for the Spark image of your choice by adding the following key-value pairs, your Spark interactive session will be created with the Spark image of your choice.

     ```
     Key: spark.kubernetes.container.image
     Value: <spark-image-of-your-choice>
     ```

3. To specify the details for other boxes or options in the **Session Configurations and Dependencies** step and to complete creating interactive sessions, see Creating Interactive Sessions on page 394.

Using Notebooks

To use HPE-curated Spark images when using Spark magic (`%manage_spark`) to create Livy sessions, follow these steps:

1. Run `%manage_spark` to connect to the Livy server and start a new session. See %manage_spark on page 447 for details.

2. Once you run `%manage_spark`, you have two options:

   - Creating sessions with the default Spark configurations. This will use the HPE-curated Spark image to create an interactive session. The List of Spark Images on page 377 page also lists the default HPE-curated Spark images used for GUI experience.

   - Running `%config_spark` and updating the value of `spark.kubernetes.container.image` to the Spark image of your choice. This will use the Spark image of your choice to create an interactive session.

3. To specify the details for the other boxes or options in the **Create Session** step and to complete creating Livy session, see %manage_spark on page 447.

Using Airflow

When you submit the Spark application by using Airflow, your Spark application will be configured with your chosen Spark image in your YAML file. This YAML file is set in the Airflow DAG.

For example:

```
submit = SparkKubernetesOperator(
    task_id='submit',
    namespace="example",
    application_file="example.yaml",
    dag=dag,
    api_group="sparkoperator.hpe.com",
    enable_impersonation_from_ldap_user=True
)
```

To learn about how to submit Spark applications by using Airflow DAG, see Submitting Spark Applications by Using DAGs on page 358.

Using Spark OSS Images

Describes how to use Spark Open-Source Software (OSS) images to submit Spark applications.

Spark OSS are Apache Spark images that do not support Data Fabric filesystem, Data Fabric Streams, or any other Data Fabric sources and sinks that require a Data Fabric client. These Spark images also do not support Data Fabric-specific security features (data-fabric SASL (`maprsasl`)).

You can use Spark OSS images with two different workflows as follows:

- Spark Operator workflow using the **Create Spark Application** GUI. See Using the Create Spark Application GUI on page 373.
- Spark Operator workflow using Airflow. See Using Airflow on page 374.

Using the Create Spark Application GUI

To use Spark OSS images, choose one of the following option in the GUI:

Using Upload YAML in GUI

1. Select the Spark OSS image from the List of Spark Images on page 377.
2. Configure your Spark YAML file with the Spark OSS image.

   ```
   image: gcr.io/mapr-252711/
   apache-spark:<image-tag>
   ```

3. To set the logged-in user's context, add the following configuration in the `sparkConf` section.

   ```
   spark.hpe.webhook.security.context.
   autoconfigure: "true"
   ```

   To learn more about user context, see Setting the User Context on page 376.
4. Perform the instructions to create a Spark application as described in Creating Spark Applications on page 379 until you reach the **Application Details** step.
5. In the **Application Details** step, choose the **Upload YAML** option.
6. Click **Select File** and, browse and upload the YAML file.

7. To specify the details for other boxes or options in the **Application Details** step and to complete creating the Spark application, see Creating Spark Applications on page 379.

Using New application in GUI

1. Perform the instructions to create a Spark application as described in Creating Spark Applications on page 379 until you reach the **Review** step.
2. To open an editor to change the application configuration using YAML in the GUI, click **Edit YAML**.
3. Select the Spark OSS image from the List of Spark Images on page 377.
4. Replace the default Spark image in YAML with the Spark OSS image.

```
image: gcr.io/mapr-252711/
apache-spark:<image-tag>
```

5. To set the logged-in user's context, add the following configuration in the `sparkConf` section.

```
spark.hpe.webhook.security.context.
autoconfigure: "true"
```

   To learn more about user context, see Setting the User Context on page 376.

6. To submit the application with the Spark OSS image, click **Create Spark Application** on the bottom right of the **Review** step.

To learn about how to submit Spark applications by using GUI, see Creating Spark Applications on page 379.

Using Airflow

When you submit the Spark application by using Airflow, your Spark application will be configured with your chosen Spark image in your YAML file. This YAML file is set in the Airflow DAG.

For example:

```
submit = SparkKubernetesOperator(
    task_id='submit',
    namespace="example",
    application_file="example.yaml",
    dag=dag,
    api_group="sparkoperator.hpe.com",
    enable_impersonation_from_ldap_user=True
)
```

To learn about how to submit Spark applications by using Airflow DAG, see Submitting Spark Applications by Using DAGs on page 358.

Using Your Own Open-Source Spark Images

Describes how to use your own open-source Spark images to submit Spark applications.

You can use your own open-source Spark images that are compatible with the Kubernetes version supported on HPE Ezmeral Unified Analytics Software. By bringing your own open-source Spark, you can build Spark with any profile of your choice; however, there will be no support for Data Fabric filesystem, Data Fabric Streams, or any other Data Fabric sources and sinks that require a Data Fabric client. Also, open-source Spark images will not support Data Fabric-specific security features (data-fabric SASL (`maprsasl`)).

To use your own open-source Spark images, follow the next steps:

1. Build Spark. See Building Spark.
2. Build Spark images to run in HPE Ezmeral Unified Analytics Software. See Building Images.
3. Choose one of the following:
   - Using the Create Spark Application GUI on page 375
   - Using Airflow on page 376

Using the Create Spark Application GUI

To use your own open-source Spark images, choose one of the following option in the GUI:

Using Upload YAML

1. Configure your Spark YAML file with the built Spark image of your choice.

   ```
   image: <base-repository>/
   <image-name>:<image-tag>
   ```

2. To set the logged-in user’s context, add the following configuration in the `sparkConf` section.

   ```
   spark.hpe.webhook.security.context.
   autoconfigure: "true"
   ```

   To learn more about user context, see Setting the User Context on page 376.

3. Perform the instructions to create a Spark application as described in Creating Spark Applications on page 379 until you reach the **Application Details** step.
4. In the **Application Details** step, choose the **Upload YAML** option.
5. Click **Select File** and, browse and upload the YAML file.
6. To specify the details for other boxes or options in the **Application Details** step and to complete creating the Spark application, see Creating Spark Applications on page 379.

Using New application

1. Perform the instructions to create a Spark application as described in Creating Spark Applications on page 379 until you reach the **Review** step.

2. To open an editor to change the application configuration using YAML in the GUI, click **Edit YAML**.
3. Replace the default Spark image in YAML with your built open-source Spark image.

   ```
   image: <base-repository>/
   <image-name>:<image-tag>
   ```

4. To set the logged-in user's context, add the following configuration in the `sparkConf` section.

   ```
   spark.hpe.webhook.security.context.
   autoconfigure: "true"
   ```

   To learn more about user context, see Setting the User Context on page 376.
5. To submit the application with your own Spark image, click **Create Spark Application** on the bottom right of the **Review** step.

Using Airflow

When you submit the Spark application by using Airflow, your Spark application will be configured with your chosen Spark image in your YAML file. This YAML file is set in the Airflow DAG.

For example:

```
submit = SparkKubernetesOperator(
    task_id='submit',
    namespace="example",
    application_file="example.yaml",
    dag=dag,
    api_group="sparkoperator.hpe.com",
    enable_impersonation_from_ldap_user=True
)
```

To learn about how to submit Spark applications by using Airflow DAG, see Submitting Spark Applications by Using DAGs on page 358.

Setting the User Context

Describes how to set the user context when using the Spark OSS images.

User context is not configured automatically for the Spark OSS images.

If you do not set the user context, the user identity that is set in the image is used to access the data sources like persistent volumes. This can cause your Spark application to fail due to a lack of proper permissions.

To set the logged-in user's context when using the Spark OSS images, add the following configuration in the `sparkConf` section of your Spark application YAML file.

```
spark.hpe.webhook.security.context.autoconfigure: "true"
```

Once you add this configuration to the YAML file, HPE Ezmeral Unified Analytics Software automatically defines and sets the user context for the logged-in user, and your Spark application runs successfully.

NOTE: Do not add this configuration to your Spark application YAML file if you are using the default Spark images that support Data Fabric services.

To learn more about pod security context, see Configure a Security Context for a Pod or Container.

List of Spark Images

Lists the Spark images distributed by HPE Ezmeral Unified Analytics Software. These images enables you to run the Spark applications in an air-gapped environment.

The images follow the following format:

```
<base-repository>/<image-name>:<image-tag>
```

HPE Ezmeral Unified Analytics Software uses two different types of image tags as follows:

Timestamped image tags

These image tags are static, and they are not updated when new changes are pushed to the image.

For example:

```
gcr.io/mapr-252711/
spark-gpu-3.4.1:202309070600R
```

Non-timestamped image tags

These image tags are dynamic, and they are updated when new changes are pushed to the image.

For example:

```
gcr.io/mapr-252711/
spark-gpu-3.4.0:v3.4.0
```

Images for HPE Ezmeral Unified Analytics Software 1.5.2

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-1.3.8:1.3.8.7-hpe`
	`gcr.io/mapr-252711/`
	`autoticketgen-2.1.4:202407171102`

Livy Server Images	`gcr.io/mapr-252711/` `livy-0.8.0:202406270245R`
Spark History Server Images	`gcr.io/mapr-252711/` `spark-hs-3.5.1:202406270245R`
(Default) HPE-Curated Spark Images When Using GUI (Supporting Data Fabric Services)	`gcr.io/mapr-252711/` `spark-3.5.1:v3.5.1.0.4`
HPE-Curated Spark Workload Images for Spark 3.5.0	`gcr.io/mapr-252711/` `spark-3.5.0:v3.5.0.3.6`

Spark OSS 3.5.1 Images (Not Supporting Data Fabric Services)

```
gcr.io/mapr-252711/
apache-spark:3.5.1-en2
```

Spark OSS 3.5.0 Images (Not Supporting Data Fabric Services)

```
gcr.io/mapr-252711/
apache-spark:3.5.0-en2
```

whylogs Images for Spark 3.5.1

```
gcr.io/mapr-252711/
spark-whylogs-3.5.1:v3.5.1.0.4
```

whylogs Images for Spark 3.5.0

```
gcr.io/mapr-252711/
spark-whylogs-3.5.0:v3.5.0.3.6
```

Images for HPE Ezmeral Unified Analytics Software 1.5.0

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-1.3.8:1.3.8.7-hpe
gcr.io/mapr-252711/
autoticketgen-2.1.4:202407171102
```

Livy Server Images

```
gcr.io/mapr-252711/
livy-0.8.0:202406270245R
```

Spark History Server Images

```
gcr.io/mapr-252711/
spark-hs-3.5.1:202406270245R
```

(Default) HPE-Curated Spark Images When Using GUI (Supporting Data Fabric Services)

```
gcr.io/mapr-252711/
spark-3.5.1:v3.5.1.0.1
gcr.io/mapr-252711/
spark-gpu-3.5.1:v3.5.1.0.1
gcr.io/mapr-252711/
spark-py-3.5.1:v3.5.1.0.1
gcr.io/mapr-252711/
spark-r-3.5.1:v3.5.1.0.1
gcr.io/mapr-252711/
spark-whylogs-3.5.1:v3.5.1.0.1
```

HPE-Curated Spark Workload Images for Spark 3.5.0

```
gcr.io/mapr-252711/
spark-3.5.0:v3.5.0.3.5
gcr.io/mapr-252711/
spark-gpu-3.5.0:v3.5.0.3.5
gcr.io/mapr-252711/
spark-py-3.5.0:v3.5.0.3.5
gcr.io/mapr-252711/
spark-r-3.5.0:v3.5.0.3.5
```

Spark OSS 3.5.1 Images (Not Supporting Data Fabric Services)

```
gcr.io/mapr-252711/apache-spark:3.5.1
gcr.io/mapr-252711/
apache-spark:3.5.1-py
```

	`gcr.io/mapr-252711/apache-spark:3.5.1-r` `gcr.io/mapr-252711/apache-spark:3.5.1-gpu`

Spark OSS 3.5.0 Images (Not Supporting Data Fabric Services)	`gcr.io/mapr-252711/apache-spark:3.5.0`
	`gcr.io/mapr-252711/apache-spark:3.5.0-py`
	`gcr.io/mapr-252711/apache-spark:3.5.0-r`
	`gcr.io/mapr-252711/apache-spark:3.5.0-gpu`

Creating Spark Applications

Describes how to create and submit Spark applications using HPE Ezmeral Unified Analytics Software.

Prerequisites

- Must have a main application file (for example, compiled jar file for Java or Scala).
- Must know the runtime dependencies of your application that are not built-in to the main application file.
- Must know your application arguments.

Create a Spark Application

Complete the following steps to create and submit a Spark application:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select one of the following options:
   - Click the **Analytics** icon and click **Spark Applications** on the left navigation bar of the HPE Ezmeral Unified Analytics Software screen.
   - Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Spark Operator** tile under the **Analytics** tab and click **Open**.
3. Click **Create Application** on the Spark Applications screen. Navigate through each step within the **Create Spark Application** wizard.

   The following table lists the steps in wizard and instructions:

Steps	Instructions
Application Details	Create an application or upload a preconfigured YAML file. • **YAML FILE** - When you select **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. • **Description** - Enter the application description.

Steps	Instructions

Configure Spark Application	Configure the Spark application:
	- **Type** - Select the application type from Java, Scala, Python, or R.
	- **Source** - Select the location of the main application file from **User Directory**, **Shared Directory**, **S3**, and **Other**. HPE Ezmeral Unified Analytics Software preconfigures Spark applications and Livy sessions in such a way that both `<username>` and `shared` volumes are mounted to driver and executor runtimes. For additional details, see the Selecting the Location of the Main Application File on page 382 section below.
	- **File Name** - Manually enter the location and file name of the application for the **S3** and **Other** sources. For example:
	`s3a://apps/my_application.jar`
	For **User Directory** and **Shared Directory**, click **Browse**, and browse and select files.
	NOTE: Ensure the extension of the main application file matches the selected application type. The extension must be `.py` for Python, `.jar` for Java and Scala, and `.r` for R applications.
	- **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.
	NOTE:
	- To refer to data in mounted folders from application source code, use `file://` schema.
	If a Spark application is reading a file from the `shared` or `user` volume and is taking a path to the file as an application argument, the argument will be `file://[mount-path]/path/to/input/file`. For example:
	`User Directory: file:///mounts/<user-name>-volume/`
	`Shared Directory: file:///mounts/shared-volume/`

Dependencies	To add dependencies required to run your applications, select a dependency type from excludePackages, files, jars, packages, pyfiles, or repositories, and enter the value of the dependency. To add more than one dependency, click **Add Dependency**. For example: - Enter the package names as the values for the excludePackages dependency type. - Enter the locations of file, for example, s3://<path-to file>, local://<path-to-file> as the values for files, jars, pyfiles, or repositories.
Driver Configuration	Configure the number of cores, core limits, and memory. The number of cores must be less than or equal to the core limit. See Configuring Memory for Spark Applications on page 393. When boxes in this wizard are left blank, the default values are set. The default values are as follows: - Number of Cores: 1 - Core Limit: unlimited - Memory: 1g

Steps	Instructions
Executor Configuration	Configure the number of executors, number of cores, core limits, and memory. The number of cores must be less than or equal to the core limit. See Configuring Memory for Spark Applications on page 393. When boxes in this wizard are left blank, the default values are set. The default values are as follows: • Number of Executors: 1 • Number of Cores per Executor: 1 • Core Limit per Executor: unlimited • Memory per Executor: 1g
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: **a.** To choose from predefined intervals, select **Predefined Frequency Interval** and click **Update** to open a dialog with predefined intervals. **b.** 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`
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**.

4. To submit the application, click **Create Spark Application** on the bottom right of the **Review** step.

Results:

The Spark application is created and will immediately run, or will wait to run at its scheduled time. You can view it on the **Spark Applications** screen.

Selecting the Location of the Main Application File

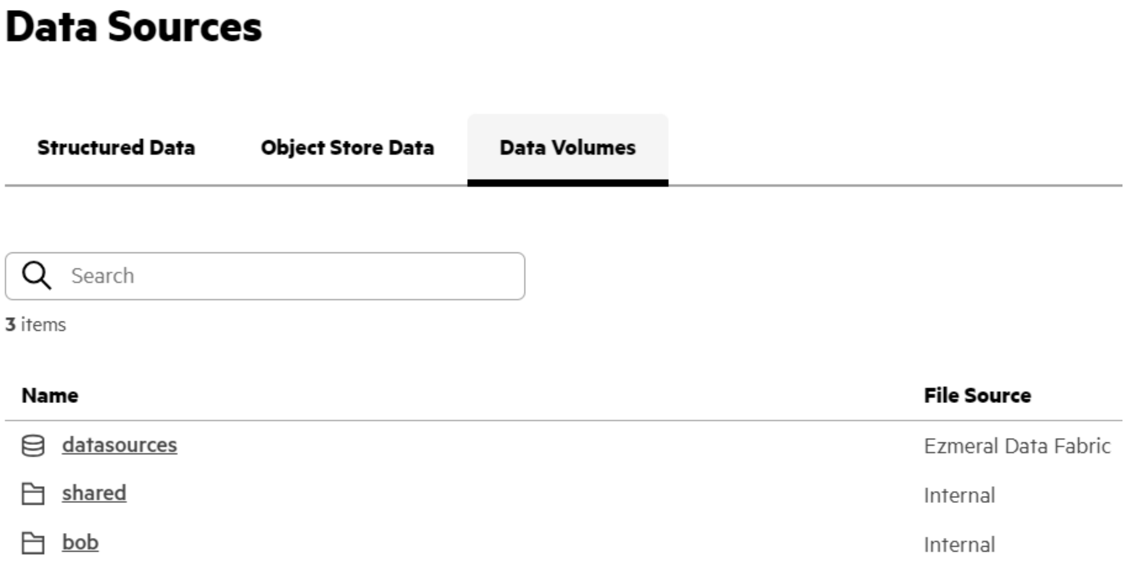
Use one of the following methods to select the location of the main application file:

Uploading Files to the User and Shared Directories

To upload files to the `user` and `shared` directories:

1. Open a different HPE Ezmeral Unified Analytics Software browser.
2. In the left navigation bar, select **Data Engineering Data Sources** and then select the **Data Volumes** tab.
3. On the **Data Volumes** tab, select your user directory or the `shared` directory.

   The following image shows an example of a user (`bob`) directory and a `shared` directory:



Name	File Source
datasources	Ezmeral Data Fabric
shared	Internal
bob	Internal

   If you do not see your user directory or the `shared` directory, contact your administrator.

4. Click **Upload** to upload the Spark application files to the `user/` or `shared/` directory.
5. Return to the browser you were working in with the **Configure Spark Application** wizard.
6. Click **Browse**, and navigate to the location where you uploaded files.
7. Select the Spark application files.

Using S3

When you select S3 as the **Source**, the **S3 Endpoint**, **Secret**, and **File Name** fields appear. The following sections describe what values to enter in these fields:

S3 Endpoint

- Enter an S3 endpoint for direct access to an external S3 data source or enter an S3 endpoint for access through the S3 proxy in Unified Analytics, as described in Getting the Data Source Name and S3 Proxy Endpoint URL.

  For an explanation of direct access versus S3 proxy access, see Configuring a Spark Application to Access External S3 Object Storage on page 383.

Secret

- For HPE-Curated Spark and Spark OSS images, you can enter `access-token` in the **Secret** field. For information about the `access-token` secret, see Auth Tokens on page 238.
- For HPE-Curated Spark images, you can leave the **Secret** field empty if you include `spark.hadoop.fs.s3a.aws.credentials.provider: "org.apache.spark.s3a.EzSparkAWSCredentialProvider"` in the Spark configuration.
- Alternatively, you can generate a secret, as described in Configuring a Spark Application to Directly Access Data in an External S3 Data Source on page 385 and Configuring a Spark Application to Access Data in an External S3 Data Source through the S3 Proxy Layer on page 387.

File Name

Enter the location and name of the Spark application file.

Using Other

Select **Other** as the data source, to reference other locations of the application file.

For example, to refer to main application files and dependency files, or to refer to a file inside the specific Spark image, use the `local://` schema.

```
local:///opt/mapr/spark/spark-3.2.0/examples/jars/
spark-examples_2.12-3.2.0.16-eep-810.jar
```

Related concepts

Configuring a Spark Application to Access External S3 Object Storage on page 383
Describes configuration options for connecting Spark to external S3 object storage.

Related tasks

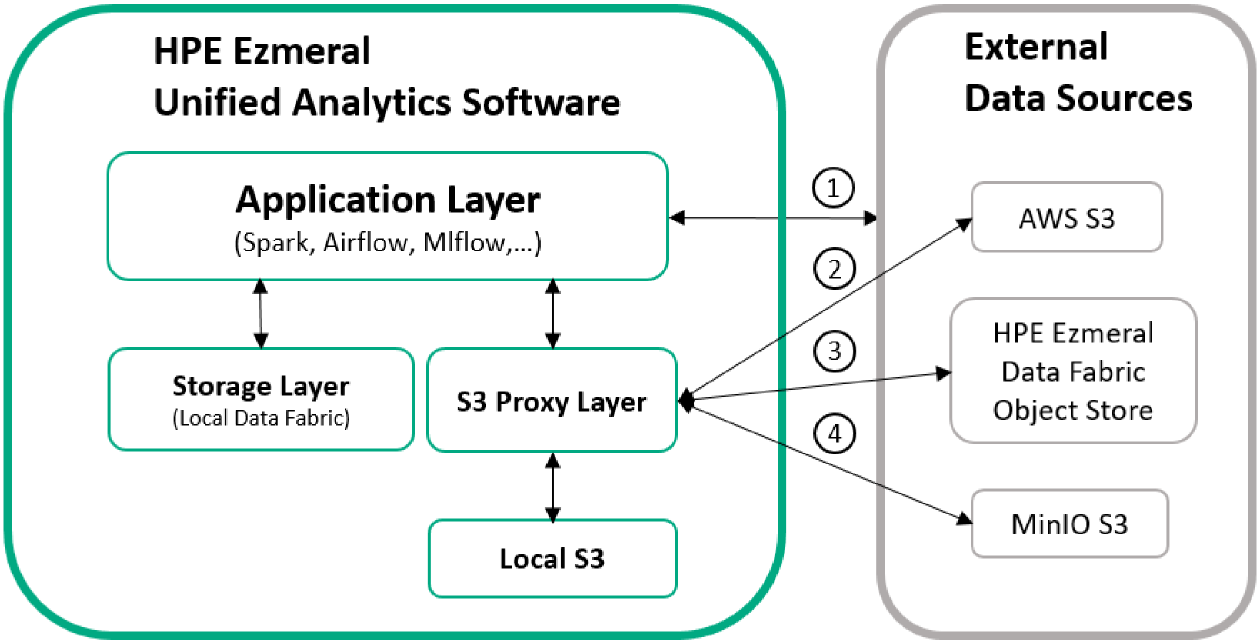
Submitting a Spark Wordcount Application on page 60
Provides an end-to-end example for creating and submitting a wordcount Spark Application in HPE Ezmeral Unified Analytics Software.

Configuring a Spark Application to Access External S3 Object Storage

Describes configuration options for connecting Spark to external S3 object storage.

You can configure a Spark application to connect to an external S3 data source directly or through the S3 proxy layer in HPE Ezmeral Unified Analytics Software.

The following diagram shows how applications in Unified Analytics access external S3 data sources, either through a direct connection from the application to an external S3 data source, as depicted by 1, or through the S3 proxy layer, as depicted by 2, 3, and 4.

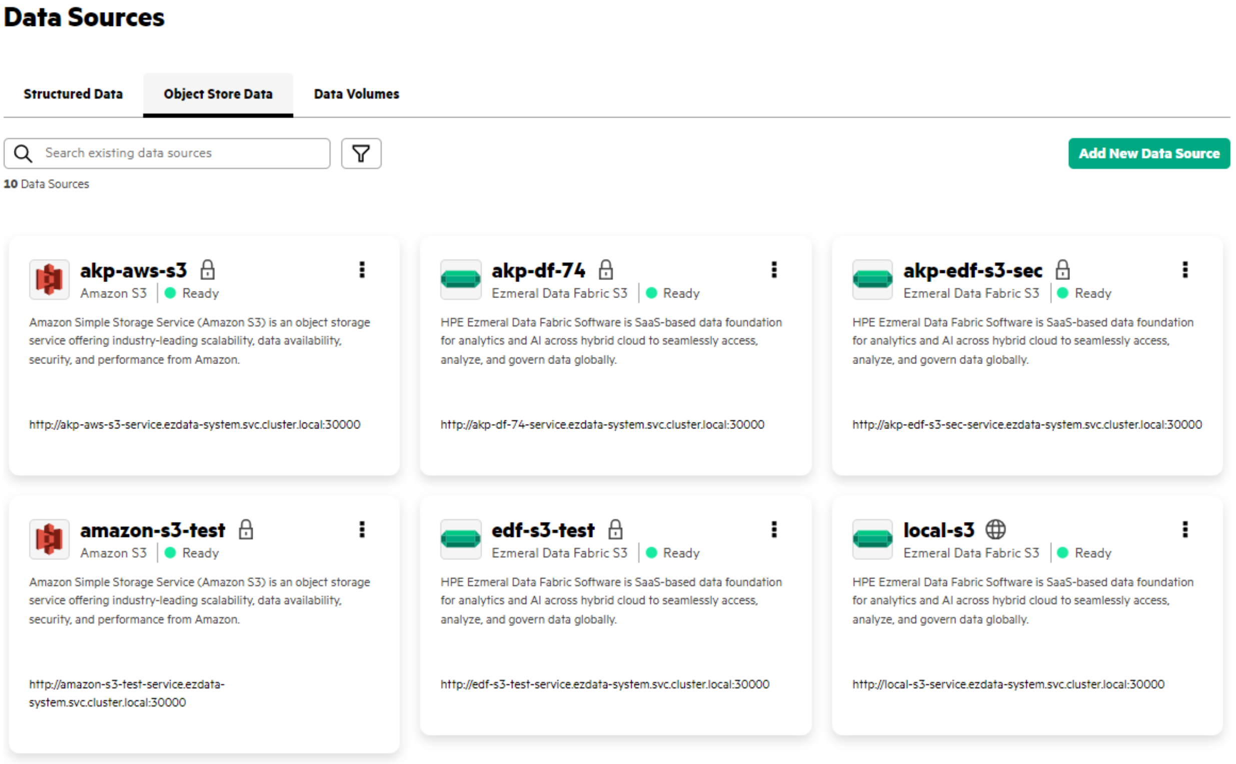


The S3 proxy layer securely connects Unified Analytics to external data sources, such as AWS S3, MinIO S3, and HPE Ezmeral Data Fabric Object Store.

When you configure a Spark application to access an S3 data source through the S3 proxy layer, you do not have to provide the access credentials or ask an administrator for access to the data source. Your Unified Analytics administrator creates the connections to external S3 data sources and provides the required access credentials (access key and secret key) at that time. Your administrator also grants permissions on the data sources. Your access to the data sources is authorized through Unified Analytics.

You can see the external S3 data sources that your administrator configured for you in the Unified Analytics UI by signing in and going to **Data Engineering > Data Sources** and clicking on the **Object Store Data** tab.

The following image shows an example of the **Object Store Data** tab with tiles for each of the connected external S3 data sources.



Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance from Amazon.

http://akp-aws-s3-service.ezdata-system.svc.cluster.local:30000

HPE Ezmeral Data Fabric Software is SaaS-based data foundation for analytics and AI across hybrid cloud to seamlessly access, analyze, and govern data globally.

http://akp-df-74-service.ezdata-system.svc.cluster.local:30000

HPE Ezmeral Data Fabric Software is SaaS-based data foundation for analytics and AI across hybrid cloud to seamlessly access, analyze, and govern data globally.

http://akp-edf-s3-sec-service.ezdata-system.svc.cluster.local:30000

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance from Amazon.

http://amazon-s3-test-service.ezdata-system.svc.cluster.local:30000

HPE Ezmeral Data Fabric Software is SaaS-based data foundation for analytics and AI across hybrid cloud to seamlessly access, analyze, and govern data globally.

http://edf-s3-test-service.ezdata-system.svc.cluster.local:30000

HPE Ezmeral Data Fabric Software is SaaS-based data foundation for analytics and AI across hybrid cloud to seamlessly access, analyze, and govern data globally.

http://local-s3-service.ezdata-system.svc.cluster.local:30000

The following topics describe each of the methods (direct or S3 proxy) for connecting Spark to an external S3 data source.

Configuring a Spark Application to Directly Access Data in an External S3 Data Source
Describes how to configure a Spark application to connect directly to an external S3 data source.

To connect a Spark application directly to an external S3 data source, you must provide the following information in the `sparkConf` section of the Spark application YAML file:

- Access credentials (access key, secret key)
- Secret (to securely pass configuration values)
- Endpoint URL
- Bucket name (name of the bucket in the S3 data source)
- Region (domain)

How you configure the Spark application depends on the type of Spark image used, either HPE-Curated Spark or Spark OSS. Follow the steps in the section that applies to the type of image used.

HPE-Curated Spark

Use these instructions if you are configuring a Spark application using the HPE-Curated Spark image.

Using the `EzSparkAWSCredentialProvider` option in the configuration automatically generates the secret for you.

The following example shows the required configuration options for a Spark application that uses the HPE-Curated Spark image:

```
sparkConf:
    spark.hadoop.fs.s3a.access.key: <S3-ACCESS_KEY>
    spark.hadoop.fs.s3a.secret.key: <S3-SECRET-KEY>
    spark.hadoop.fs.s3a.endpoint: <S3-endpoint>
    spark.hadoop.fs.s3a.connection.ssl.enabled: "true"
    spark.hadoop.fs.s3a.impl: org.apache.hadoop.fs.s3a.S3AFileSystem
    spark.hadoop.fs.s3a.aws.credentials.provider:
"org.apache.spark.s3a.EzSparkAWSCredentialProvider"
    spark.hadoop.fs.s3a.path.style.access: "true"
```

(AWS S3 Only) If you are connecting the Spark application to an AWS S3 data source, you must also include the following options in the `sparkConf` section:

```
spark.driver.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/java/
cacerts
spark.executor.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/java/
cacerts
```

Spark OSS

Use these instructions if you are configuring a Spark application using the Spark OSS image.

Complete the following steps:

1. **Generate a secret.**

   Use either of the following methods to generate a secret:

   Use a Notebook to Generate the Secret

   Use a notebook to create a Kubernetes secret with Base64-encoded values for the 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
   ```

   See Creating and Managing Notebook Servers on page 436.

   Use a Configuration File to Generate the Secret

   Create a `spark-defaults.conf` file to generate the secret. Provide the object store *access key* and *secret key* as values for the `spark.hadoop.fs.s3a.access.key` and `spark.hadoop.fs.s3a.secret.key` properties in the file.

   a. Create a `spark-defaults.conf` file with the following properties:

      ```
      spark.hadoop.fs.s3a.access.key
      EXAMPLE_ACCESS_KEY
      spark.hadoop.fs.s3a.secret.key
      EXAMPLE_SECRET_KEY
      ```

b. Create a secret from the `spark-defaults.conf` file:

```
kubectl create secret generic
<k8s-secret-name> --from-file=spa
rk-defaults.conf
```

2. **Configure the Spark application.**

   The following example demonstrates how to add the required fields to the `sparkConf` section of the Spark application YAML file:

```
sparkConf:
    spark.hadoop.fs.s3a.access.key: <S3-ACCESS_KEY>
    spark.hadoop.fs.s3a.secret.key: <S3-SECRET-KEY>
    spark.hadoop.fs.s3a.connection.ssl.enabled: "true"
    spark.hadoop.fs.s3a.endpoint: <S3-endpoint>
    spark.hadoop.fs.s3a.impl: org.apache.hadoop.fs.s3a.S3AFileSystem
    spark.hadoop.fs.s3a.path.style.access: "true"
```

(AWS S3 Only) If you are connecting the Spark application to an AWS S3 data source, you must also include the following options in the `sparkConf` section:

```
spark.driver.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/java/
cacerts
spark.executor.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/java/
cacerts
```

(Optional) Setting Environment Variables for the Access Key and Secret Key

Regardless of the type of Spark imaged used, you can set environment variables for the access key and secret key.

Set environment variables for the access key and secret key, as shown:

```
spark.kubernetes.driverEnv.AWS_ACCESS_KEY_ID: <ACCESS_KEY>
spark.kubernetes.driverEnv.AWS_SECRET_ACCESS_KEY: <SECRET_KEY>
spark.executorEnv.AWS_ACCESS_KEY_ID: <ACCESS_KEY>
spark.executorEnv.AWS_SECRET_ACCESS_KEY: <SECRET_KEY>
```

IMPORTANT: When you set these environment variables, the user access token (JWT) is not automatically refreshed if the endpoint URL changes. To refresh the token, you must run %update_token on page 451.

Related tasks

Securely Passing Spark Configuration Values on page 404
Describes how to pass the sensitive data to Spark configuration using the Kubernetes Secret.

More information

Hadoop S3 Client

Configuring a Spark Application to Access Data in an External S3 Data Source through the S3 Proxy Layer

Describes how to configure a Spark application to connect to an external S3 data source through the S3 proxy later in HPE Ezmeral Unified Analytics Software.

To connect Spark to an external S3 data source, include the following informaiton in the `sparkConf` section of the Spark application YAML file:

- Data source name
- Endpoint URL
- Secret (to securely pass configuration values)
- Bucket that you want the client to access

You can find the data source name and endpoint URL on the data source tile in the HPE Ezmeral Unified Analytics Software UI.

Once connected, the Spark application can:

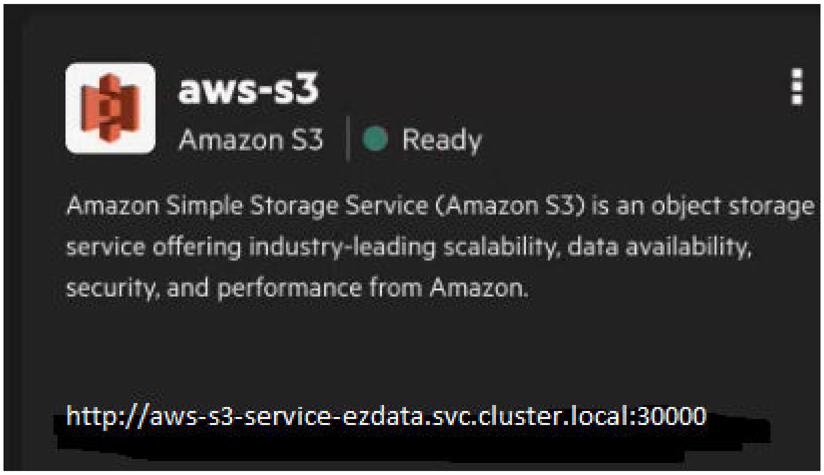
- Read and download files in a bucket
- Upload files from a bucket
- Create buckets

Getting the Data Source Name and S3 Proxy Endpoint URL

To get the data source name and S3 proxy endpoint URL:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation bar, select **Data Engineering > Data Sources**.
3. On the **Data Sources** page, find the tile for the S3 data source that you want the Spark application to connect to.

   The following image shows an example of a tile for an AWS S3 data source with the name (aws-s3) and the enpoint URL (http://aws-s3-service-ezdata.svc.cluster.local:30000):

   aws-s3

   Amazon S3 | Ready

   Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance from Amazon.

   http://aws-s3-service-ezdata.svc.cluster.local:30000

   [redacted]

   NOTE: By default, a local-s3 Ezmeral Data Fabric tile also displays on the screen. This Ezmeral Data Fabric version of S3 is a local S3 version used internally by HPE Ezmeral Unified Analytics Software. Do not connect to this data source.

4. Note the *data source name* and *endpoint URL* and then use them in the Spark configuration.

Configuring Spark

How you configure the Spark application depends on the type of Spark image used, either HPE-Curated Spark or Spark OSS.

- If you used the HPE-Curated Spark image to create the Spark application, see HPE-Curated Spark on page 389.
- If you used the Spark OSS image to create the Spark application, see Spark OSS.

HPE-Curated Spark

Use these instructions if you are configuring a Spark application using the HPE-Curated Spark image.

Using the `EzSparkAWSCredentialProvider` option in the configuration automatically generates the secret for you.

The following example shows the required configuration options for a Spark application that uses the HPE-Curated Spark image:

```
sparkConf:
    spark.hadoop.fs.s3a.endpoint: <S3-endpoint>
    spark.hadoop.fs.s3a.connection.ssl.enabled: "true"
    spark.hadoop.fs.s3a.impl: org.apache.hadoop.fs.s3a.S3AFileSystem
    spark.hadoop.fs.s3a.aws.credentials.provider: 
"org.apache.spark.s3a.EzSparkAWSCredentialProvider"
    spark.hadoop.fs.s3a.path.style.access: "true"
```

(AWS S3 Only) If you are connecting the Spark application to an AWS S3 data source, you must also include the following options in the `sparkConf` section:

```
spark.driver.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/java/
cacerts
spark.executor.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/java/
cacerts
```

Spark OSS

Use these instructions if you are configuring a Spark application using the Spark OSS image.

Complete the following steps:

1. **Generate a secret.**

   Use either of the following methods to generate a secret:

   Apply a YAML

   Use a notebook to create a Kubernetes secret with Base64-encoded values for the 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
```

See Creating and Managing Notebook Servers on page 436

Run a Script in a Notebook

Run the following script in a notebook to generate the secret. You can also access a sample script from your notebook server in the `shared/ezua-tutorials/Data-Analytics/` directory.

```
def deploy_s3_secret(namespace,
spark_secret):
    try:
        #Run kubectl apply command
using subprocess
        subprocess.run(['kubectl',
'delete', 'secret', spark_secret,
'-n', namespace], check=False)
        subprocess.run(['kubectl',
'create', 'secret', 'generic',
spark_secret, '-n', namespace ,
'--from-file=spark-defaults.conf'],
check=True)
        print("Secret creation
successful!")
        except
subprocess.CalledProcessError as e:
        print(f"Secret creation
failed. Error: {e}")

s3_access_data =
"spark.hadoop.fs.s3a.access.key
EXAMPLE_ACCESS_KEY"
s3_secret_data =
"spark.hadoop.fs.s3a.secret.key
EXAMPLE_SECRET_KEY"

s3_data =
s3_access_data.replace('EXAMPLE_ACCE
SS_KEY', os.environ['AUTH_TOKEN'])
s3_data += "\n" +
s3_secret_data.replace("EXAMPLE_SECR
ET_KEY", "s3")

namespace = os.environ['USER']
spark_secret = "spark-s3-secret"

#Save data to a file
spark-defaults.conf
with open('spark-defaults.conf',
'w') as file:
    file.write(s3_data)

# Call the function to deploy the
Kubernetes secret
deploy_s3_secret(namespace,
spark_secret)
```

2. **Configure the Spark application.**

   The following example demonstrates how to add the required fields to the `sparkConf` section of the Spark application YAML file:

```
sparkConf:
    spark.hadoop.fs.s3a.connection.ssl.enabled: "true"
    spark.hadoop.fs.s3a.endpoint: <S3-endpoint>
    spark.hadoop.fs.s3a.impl: org.apache.hadoop.fs.s3a.S3AFileSystem
    spark.hadoop.fs.s3a.path.style.access: "true"
```

   (AWS S3 Only) If you are connecting the Spark application to an AWS S3 data source, you must also include the following options in the `sparkConf` section:

```
spark.driver.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/java/
cacerts
spark.executor.extraJavaOptions: -Djavax.net.ssl.trustStore=/etc/pki/
java/cacerts
```

Managing Spark Applications

Describes how to view and manage Spark applications using HPE Ezmeral Unified Analytics Software.

About this task

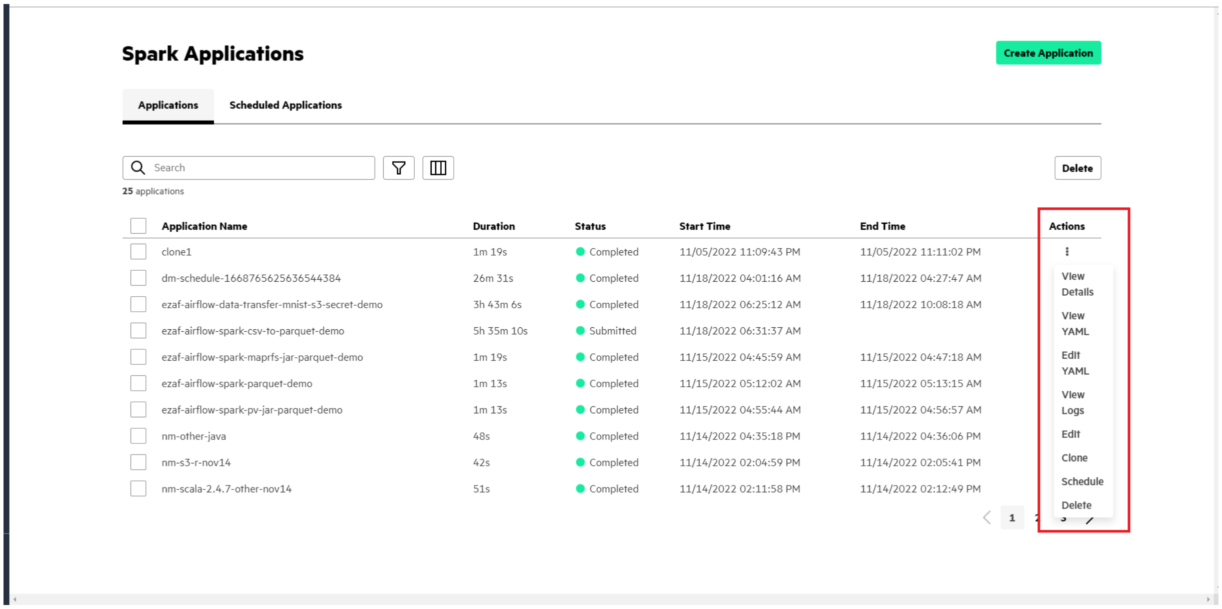
View and manage the status of all the Spark applications and scheduled Spark applications.

Procedure

1. To view and manage Spark applications, you can choose one of the following options:

   - Click the **Analytics** icon and click **Spark Applications** on the left navigation bar of the HPE Ezmeral Unified Analytics Software screen.
   - Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Spark** tile under the **Analytics** tab and click **Open**.

2. To view actions that you can perform on the **Applications** and **Scheduled Applications** tab, click the **menu** icon in the **Actions** column.



25 applications

☐	Application Name	Duration	Status	Start Time	End Time	Actions
☐	clone1	1m 19s	● Completed	11/05/2022 11:09:43 PM	11/05/2022 11:11:02 PM	⋮
☐	dm-schedule-1668765625636544384	26m 31s	● Completed	11/18/2022 04:01:16 AM	11/18/2022 04:27:47 AM	
☐	ezaf-airflow-data-transfer-mnist-s3-secret-demo	3h 43m 6s	● Completed	11/18/2022 06:25:12 AM	11/18/2022 10:08:18 AM	
☐	ezaf-airflow-spark-csv-to-parquet-demo	5h 35m 10s	● Submitted	11/18/2022 06:31:37 AM		
☐	ezaf-airflow-spark-maprfs-jar-parquet-demo	1m 19s	● Completed	11/15/2022 04:45:59 AM	11/15/2022 04:47:18 AM	
☐	ezaf-airflow-spark-parquet-demo	1m 13s	● Completed	11/15/2022 05:12:02 AM	11/15/2022 05:13:15 AM	
☐	ezaf-airflow-spark-pv-jar-parquet-demo	1m 13s	● Completed	11/15/2022 04:55:44 AM	11/15/2022 04:56:57 AM	
☐	nm-other-java	48s	● Completed	11/14/2022 04:35:18 PM	11/14/2022 04:36:06 PM	
☐	nm-s3-r-nov14	42s	● Completed	11/14/2022 02:04:59 PM	11/14/2022 02:05:41 PM	
☐	nm-scala-2.4.7-other-nov14	51s	● Completed	11/14/2022 02:11:58 PM	11/14/2022 02:12:49 PM	

View Details:

To view the details of an application, 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.

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**. 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 update all the application parameters except name, and type using **Edit**. Use **Clone** to update the parameters and create an application.

You can update the schedule of the scheduled Spark application by using **Edit**.

To open an editor to change the application configuration using YAML, click **Edit YAML** in the **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 and configure the scheduling details in the **Schedule Application** step, it will create a new scheduled Spark application.
>
> Submitting an application 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 379.

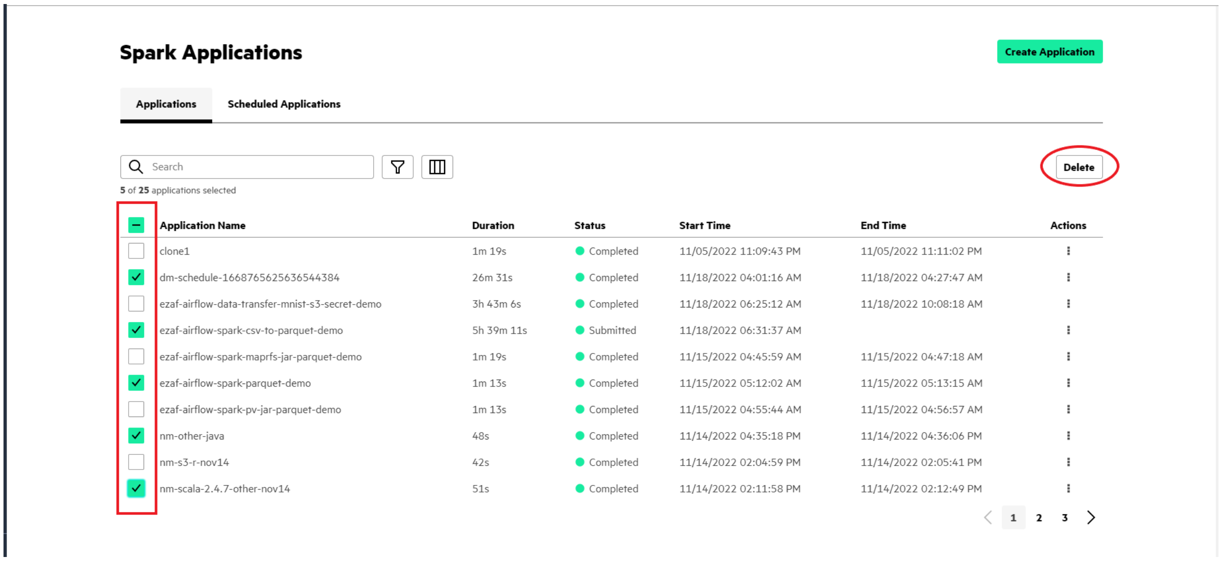
Suspend:

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 the **Actions** menu in **Scheduled Applications** tab.

Delete: To delete the Spark application, select **Delete**.

3. Delete multiple Spark applications at once:



5 of 25 applications selected

☐	Application Name	Duration	Status	Start Time	End Time	Actions
☐	clone1	1m 19s	● Completed	11/05/2022 11:09:43 PM	11/05/2022 11:11:02 PM	⋮
☑	dm-schedule-1668765625636544384	26m 31s	● Completed	11/18/2022 04:01:16 AM	11/18/2022 04:27:47 AM	⋮
☐	ezaf-airflow-data-transfer-mnist-s3-secret-demo	3h 43m 6s	● Completed	11/18/2022 06:25:12 AM	11/18/2022 10:08:18 AM	⋮
☑	ezaf-airflow-spark-csv-to-parquet-demo	5h 39m 11s	● Submitted	11/18/2022 06:31:37 AM		⋮
☐	ezaf-airflow-spark-maprfs-jar-parquet-demo	1m 19s	● Completed	11/15/2022 04:45:59 AM	11/15/2022 04:47:18 AM	⋮
☑	ezaf-airflow-spark-parquet-demo	1m 13s	● Completed	11/15/2022 05:12:02 AM	11/15/2022 05:13:15 AM	⋮
☐	ezaf-airflow-spark-pv-jar-parquet-demo	1m 13s	● Completed	11/15/2022 04:55:44 AM	11/15/2022 04:56:57 AM	⋮
☑	nm-other-java	48s	● Completed	11/14/2022 04:35:18 PM	11/14/2022 04:36:06 PM	⋮
☐	nm-s3-r-nov14	42s	● Completed	11/14/2022 02:04:59 PM	11/14/2022 02:05:41 PM	⋮
☑	nm-scala-2.4.7-other-nov14	51s	● Completed	11/14/2022 02:11:58 PM	11/14/2022 02:12:49 PM	⋮

a) To select multiple applications, click the check box besides **Application Name** in the table.

b) Click **Delete** in the top right pane of the table.

4. To display the Spark applications according to the status, click the **Filter** icon.

5. To select the columns to display on your applications table, click the **Columns** icon.

Configuring Memory for Spark Applications

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 Unified Analytics Software. See Creating Spark Applications on page 379.

You can configure the driver and executor memory options for the Spark applications by manually setting the following properties in the Spark application YAML file. See Spark application YAML.

- `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, which is 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 in HPE Ezmeral Unified Analytics Software.

Set the following configurations options in the Spark application YAML file by clicking **Edit YAML** in **Review** step or **Edit YAML** from the **Actions** menu on **Spark Applications** screen. See Managing Spark Applications on page 391.

```
spark.driver.memoryOverhead
```

```
spark.executor.memoryOverhead
```

To learn more about driver or executor memory, memory overhead, and other properties, see Apache Spark 3.x.x application properties.

Creating Interactive Sessions

Describes how to create interactive sessions in HPE Ezmeral Unified Analytics Software.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software. See Get Started on page 6.

About this task

Create an interactive session in HPE Ezmeral Unified Analytics Software.

Procedure

1. To start creating interactive sessions, you can choose one of the following options:
   - Click the **Analytics** icon and click **Spark Interactive Sessions** on the left navigation bar of the HPE Ezmeral Unified Analytics Software screen.
   - Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Livy** tile under the **Analytics** tab and click **Open**.
2. Click **Create Interactive Session** in the Spark Interactive Sessions screen. Navigate through each step within the **Create Interactive Session** wizard:

   a) **Session Configurations and Dependencies**: Configure session details, Spark configurations, and dependencies. Set the following boxes:

Name:	Enter the session name.

   Spark Configurations: Set Spark configurations by providing key-value pairs See Spark Configurations for available configurations.

   To add additional Spark configurations required to run your session, click **Add Configuration**.

   Dependencies

Type:	Select one of the following dependency types: • files • jars • pyfiles • archives
Value:	Enter the file location for the dependency. For example, if the type is files, jars, or pyfiles, you could enter s3:// or local:// as the value.

To add additional dependencies required to run your session, click **Add Dependency**.

b) **Driver and Executor Configuration:** Configure the number of cores, memory, number of executors, number of cores per executor, and memory per executor.

c) **Review:** Review the session details. Click the **pencil icon** in each section to navigate to the specific step to change the session configuration.

3. To create the interactive session, click **Create Interactive Session** on the bottom right of the **Review** step.

Results

A new interactive session is created and you can view it in the **Spark Interactive Sessions** screen.

Submitting Statements

Describes how to submit statements in HPE Ezmeral Unified Analytics Software.

Prerequisites

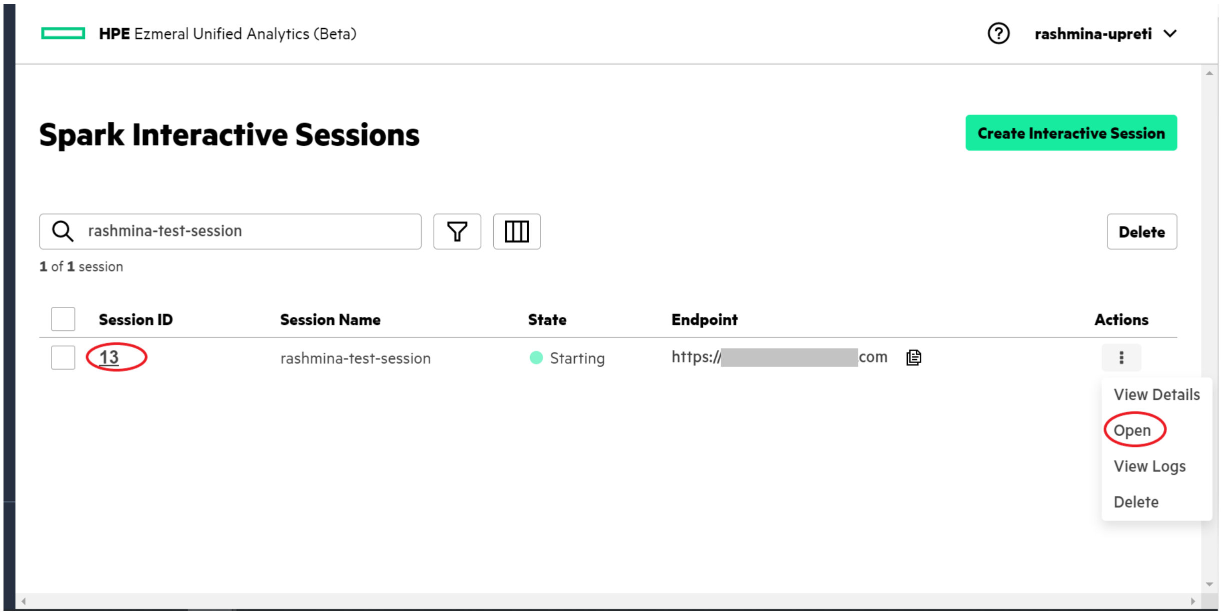
- Create an interactive session. See Creating Interactive Sessions on page 394.

About this task

Run statements in Python, R, or Scala.

Procedure

1. To submit statements, you can choose one of the following options:
   - Click **Session ID** of your Spark interactive ession.
   - Click the **menu** icon in the **Actions** column and click **Open**.



HPE Ezmeral Unified Analytics (Beta)
rashmina-upreti
Create Interactive Session
Delete
View Details
Open
View Logs
Delete

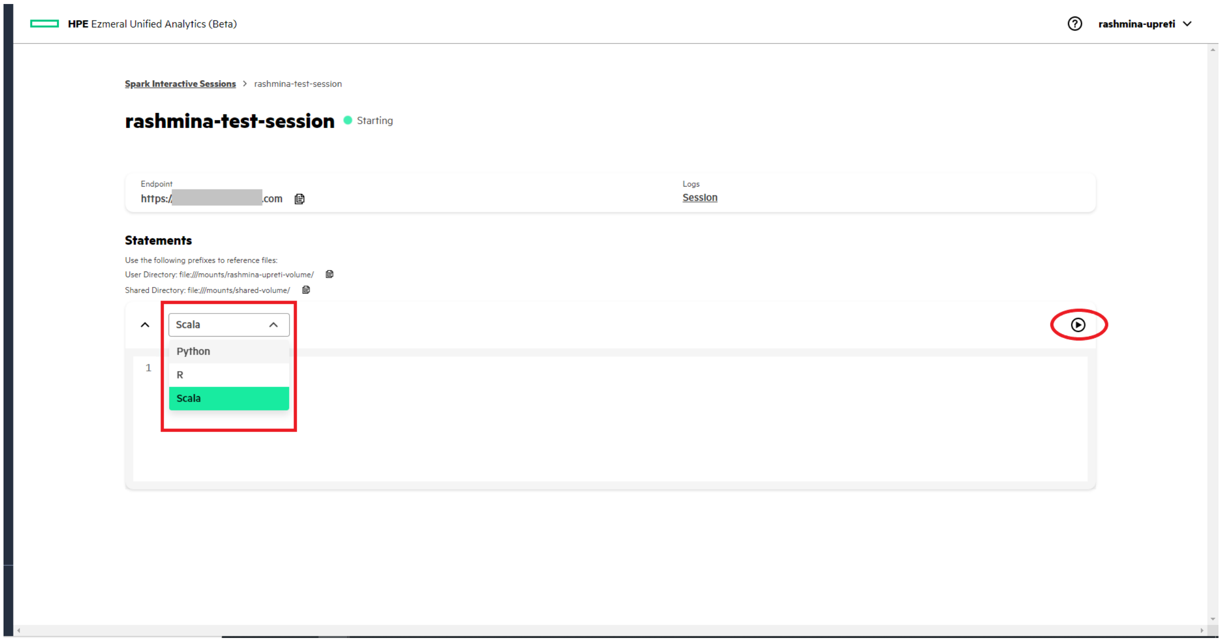
Spark Interactive Sessions

rashmina-test-session

1 of 1 session

Session ID	Session Name	State	Endpoint	Actions
13	rashmina-test-session	● Starting	https://[redacted]com	⋮

2. Select either Python, R, or Scala as the statements' programming language.



HPE Ezmeral Unified Analytics (Beta)
rashmina-upreti
Spark Interactive Sessions > rashmina-test-session

rashmina-test-session ● Starting

Endpoint: https:/[redacted].com

Logs: Session

Statements

Use the following prefixes to reference files:

User Directory: file:///mounts/rashmina-upreti-volume/

Shared Directory: file:///mounts/shared-volume/

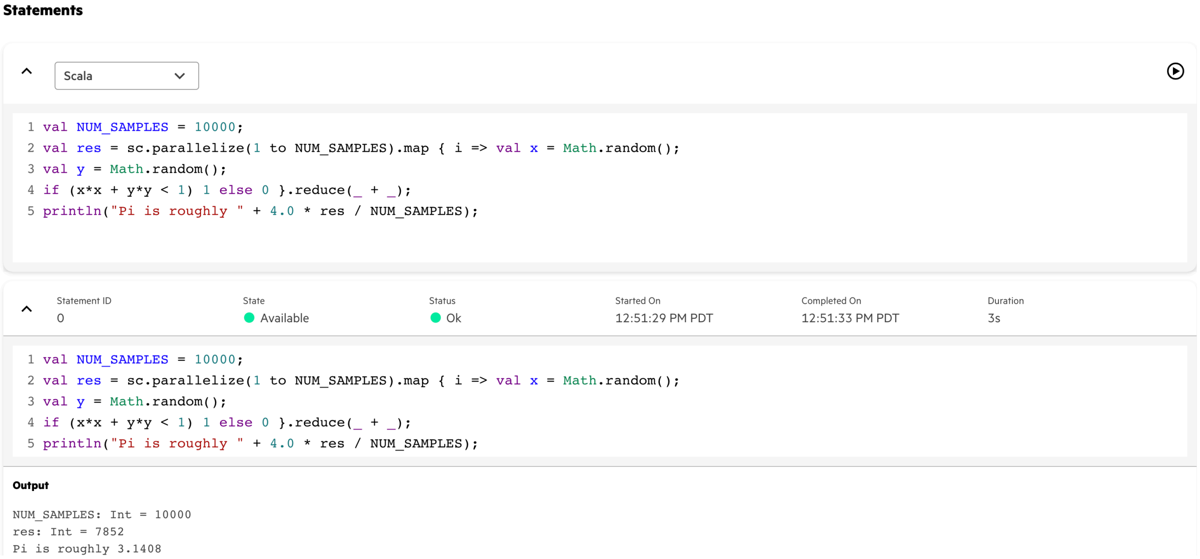
3. Enter statements in Python, R, or Scala.

   For example: Select Scala as programming language and calculate the value of Pi by running the following statement.

```
val NUM_SAMPLES = 10000;
val res = sc.parallelize(1 to NUM_SAMPLES).map { i => val x =
Math.random();
val y = Math.random();
if (x*x + y*y < 1) 1 else 0 }.reduce(_ + _);
println("Pi is roughly " + 4.0 * res / NUM_SAMPLES);
```

4. Click the **Run** icon on the top right of the **Statements** pane.

   For example: Running the previous statement returns the following statement result:



Statements

```
val NUM_SAMPLES = 10000;
val res = sc.parallelize(1 to NUM_SAMPLES).map { i => val x = Math.random();
val y = Math.random();
if (x*x + y*y < 1) 1 else 0 }.reduce(_ + _);
println("Pi is roughly " + 4.0 * res / NUM_SAMPLES);
```

Statement ID	State	Status	Started On	Completed On	Duration
0	● Available	● Ok	12:51:29 PM PDT	12:51:33 PM PDT	3s

```
val NUM_SAMPLES = 10000;
val res = sc.parallelize(1 to NUM_SAMPLES).map { i => val x = Math.random();
val y = Math.random();
if (x*x + y*y < 1) 1 else 0 }.reduce(_ + _);
println("Pi is roughly " + 4.0 * res / NUM_SAMPLES);
```

Output

```
NUM_SAMPLES: Int = 10000
res: Int = 7852
Pi is roughly 3.1408
```

NOTE: Each Spark interactive session expires in 60 minutes.

Managing Interactive Sessions

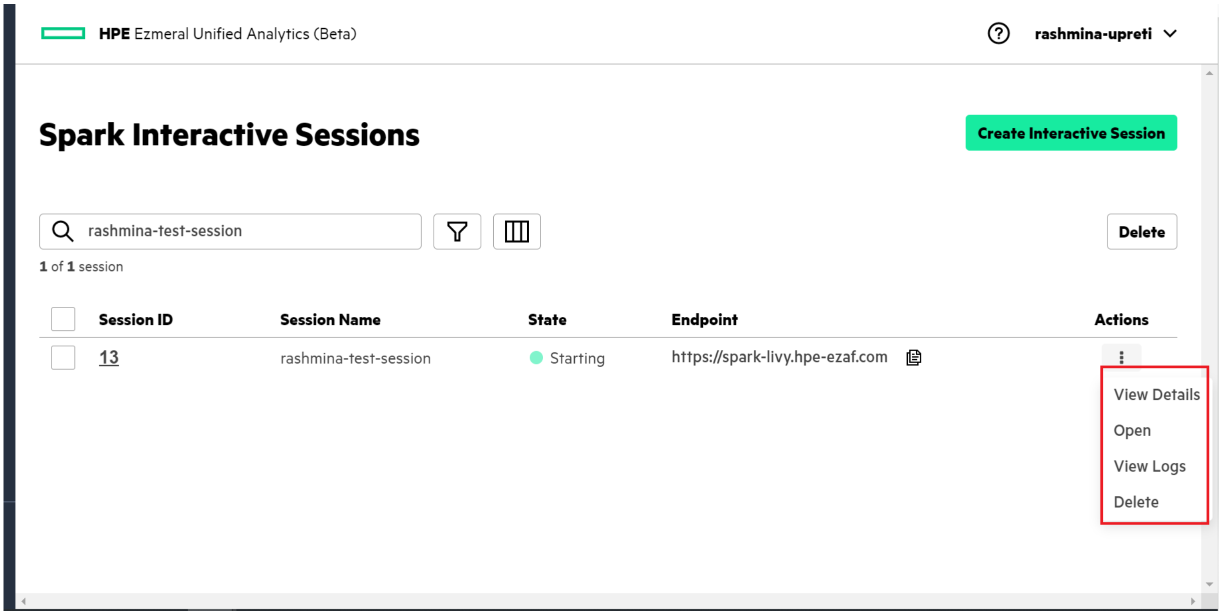
Describes how to view and manage Spark interactive sessions in HPE Ezmeral Unified Analytics Software.

About this task

View and manage the status of all the Spark interactive sessions.

Procedure

1. To view and manage Spark interactive sessions, you can choose one of the following options:
   - Click the **Analytics** icon and click **Spark Interactive Sessions** on the left navigation bar of the HPE Ezmeral Unified Analytics Software screen.
   - Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Livy** tile under the **Analytics** tab and click **Open**.
2. To view actions that you can perform on the **Spark Interactive Sessions** screen, click the **menu** icon in the **Actions** column.



Spark Interactive Sessions

1 of 1 session

□	Session ID	Session Name	State	Endpoint	Actions
□	13	rashmina-test-session	● Starting	https://spark-livy.hpe-ezaf.com	⋮

View Details: To view the details of an application, and events and logs of the pods, select **View Details**.

Open: To submit statements, select **Open**. See Submitting Statements on page 395.

View Logs: To view the session logs provided by Livy server, select **View Logs**.

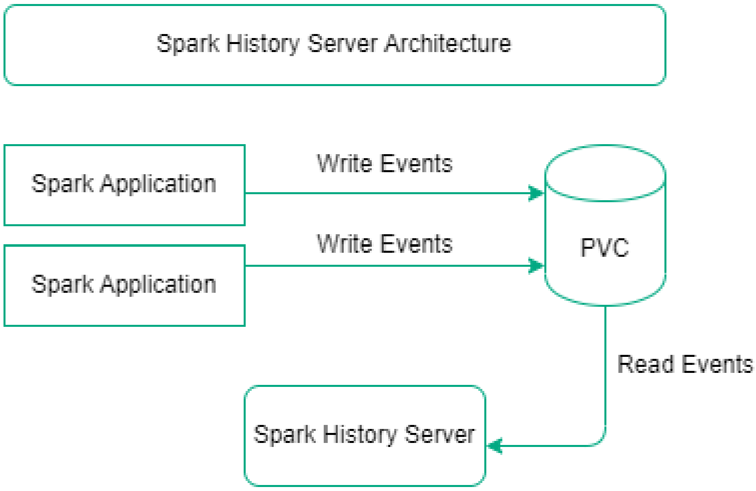
Delete: To delete the Spark interactive session, select **Delete**.

3. Delete multiple sessions at once:
   a) To select multiple sessions, click the check box besides **Session ID** in the table.
   b) Click **Delete** on the top right pane of the table.
4. To display the Spark interactive sessions according to the status, click the **Filter** icon.
5. To select the columns to display on your applications table, click the **Columns** icon.

Spark History Server

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.



To access Spark History Server in HPE Ezmeral Unified Analytics Software, click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Spark History Server** tile under the **Analytics** tab and click **Open**.

Spark History Server gathers metrics and enables you to get information about your Spark applications.

By default, all the Spark applications are integrated with Spark History Server. You can disable the integration of the Spark applications with Spark History Server by reconfiguring the Spark applications.

Spark History Server pulls the details of the Spark applications from the event logs directory. A persistent volume is mounted to all the Spark applications. The event logs from the Spark runtime are written to the event log directory on that persistent volume. Spark History Server reads the event logs and displays them on the UI.

SSO is enabled for the Spark History Server. When you sign in to the Spark History Server, you can see the list of all applications; however, you can only read the details of your own applications. Only an administrator can read the details of other users' applications. You can configure each Spark application with its own separate access control lists (ACLs), see Authentication and Authorization for details.

Using Spark SQL API

Describes how to use Spark SQL API in HPE Ezmeral Unified Analytics Software.

In HPE Ezmeral Unified Analytics Software, you can use the Spark SQL API in two different ways:

External Metastore

NOTE: There will be some limitations to integration with external metastore.

To integrate Spark with external metastore, follow these steps:

1. Set the metastore URI with the `spark.hive.metastore.uris` config option. This URI should be public and accessible from your Spark applications.
2. Set the value of `spark.sql.warehouse.dir` property to the same value as that of external metastore. For example: if you want to query a managed table then the path to that managed table must match in both metastore and Spark runtime.

3. Verify that the metastore host can accept external connections so that Spark can connect to the metastore. Configure the gateway rules for securing the metastore as the metastore doesn't have authentication and authorization.

4. Verify that your Spark applications are querying the data from locations accessible within the Spark runtime.

Temporary Views

The temporary view is a feature in the Spark DataFrame API. You can read data and create a temporary view for the data by using the temporary view feature. These views are not global and cannot be shared between any two Spark applications. You can use a temporary view in the following two scenarios:

1. If the schema is available for your data, use `DataFrame:create[OrReplace]TempView`. Some file formats already include schema, for example, parquet files or CSV files with the header. You can read the file, create a DataFrame and then call the `create[OrReplace]TempView` function and give it the view name and finally, you can query data using Spark SQL API.

2. If the schema is not available for your data, you can set it while creating or converting the DataFrame, then create the temporary view. By default, Spark sets aliases for the column names like underscore 1, underscore 2, and so on, however, you can set your own column names.

Enabling GPU Support for Spark

Describes NVIDIA `spark-rapids` accelerator support for Spark, and how to enable and allocate the GPU resources on Spark.

In HPE Ezmeral Unified Analytics Software, you can use RAPIDS Accelerator for Apache Spark by NVIDIA to accelerate the processing for Spark by using the GPUs.

The GPU image (spark-gpu-<spark-version>), for example,`spark-gpu-3.5.0` , has a built-in open-source RAPIDS plugin in HPE Ezmeral Unified Analytics Software.

To see the list of Spark GPU images, see List of Spark Images on page 377.

NOTE:

- Do not allocate GPUs for a driver pod. GPUs are used by executor pods only.
- With MIG configuration, only one GPU can be assigned per application. For details, see GPU Support on page 137.

Spark Configurations for GPU

Spark Configurations	Key	Value
GPU Images See List of Spark Images on page 377	`spark.kubernetes.container.image`	`gcr.io/mapr-252711/spark-gpu-<spark-version>:<image-tag>`
Enable RAPIDS plugin	`spark.plugins`	`com.nvidia.spark.SQLPlugin`
	`spark.rapids.sql.enabled`	`true`
	`spark.rapids.force.caller.classloader`	`false`

Spark Configurations	Key	Value
Allocate GPU resources	`spark.task.resource.gpu.amount`	1
	`spark.executor.resource.gpu.amount`	1
	`spark.executor.resource.gpu.vendor`	`nvidia.com`
Set GPU discovery script path	`spark.executor.resource.gpu.discoveryScript`	`/opt/mapr/spark/spark-<spark-version>/examples/src/main/scripts/getGpusResources.sh`
Set RAPIDS shim layer for the run[1]	`spark.rapids.shims-provider-override`	`com.nvidia.spark.rapids.shims.<spark-identifier>.SparkShimServiceProvider`

[1] The Spark version distributed by HPE is compatible with its corresponding open-source version. The RAPIDS jar includes the shim layer provider classes called `com.nvidia.spark.rapids.shims.[spark-identifier].SparkShimServiceProvider`. You can replace the `[spark-identifier]` based on the Spark distributed by HPE such as:

- For spark-3.5.0, the identifier is spark350.

More information

GPU Support on page 137
Provides information about support for NVIDIA GPU, MIG partitioning, preparing hosts for GPU-enabled environment, adding hosts and enabling GPU in HPE Ezmeral Unified Analytics Software.

Enabling GPU Support for Spark Operator
Describes how to enable and allocate GPU resources on Spark Operator.

Enabling GPU Support for Spark Operator

To enable GPU processing and allocate GPU resources on Spark Operator, follow these steps:

1. Set the image option within the `spec` property of the Spark application yaml file to `gcr.io/mapr-252711/spark-gpu-<spark-version>:<image-tag>`. To see the list of Spark GPU images, see List of Spark Images on page 377.

2. Add the following configuration options to `sparkConf` section within the `spec` property.

   - To enable the RAPIDS plugin and allocate the GPU resources, add:

     ```
     # Enabling RAPIDs plugin
     spark.plugins: "com.nvidia.spark.SQLPlugin"
     spark.rapids.sql.enabled: "true"
     spark.rapids.force.caller.classloader: "false"

     # GPU allocation and discovery settings
     spark.task.resource.gpu.amount: "1"
     spark.executor.resource.gpu.amount: "1"
     spark.executor.resource.gpu.vendor: "nvidia.com"
     ```

   - To set the path to the GPU discovery script, add:

     ```
     spark.executor.resource.gpu.discoveryScript: "/opt/mapr/spark/
     spark-<spark-version>/examples/src/main/scripts/getGpusResources.sh"
     ```

- To set the RAPIDS shim layer used for the run, add:

  ```
  spark.rapids.shims-provider-override:
  "com.nvidia.spark.rapids.shims.<spark-identifier>.SparkShimServiceProvi
  der"
  ```

  The Spark version distributed by Hewlett Packard Enterprise is compatible with its corresponding open-source version. The RAPIDS jar includes the shim layer provider classes called `com.nvidia.spark.rapids.shims.[spark-identifier].SparkShimServiceProvider`. You can replace the `[spark-identifier]` based on the Spark distributed by Hewlett Packard Enterprise such as:

  - For spark-3.5.0, the identifier is spark350.
  - For example, for spark-gpu-3.5.0, set the RAPIDS shim layer as follows:

    ```
    spark.rapids.shims-provider-override:
    "com.nvidia.spark.rapids.shims.spark350.SparkShimServiceProvider"
    ```

Verifying Spark Applications are Running on GPU

To verify the Spark applications are running on GPU, you can use the explain Spark method.

Run the following PySpark application:

```
from pyspark.sql import SQLContext
from pyspark import SparkConf
from pyspark import SparkContext

conf = SparkConf()
sc = SparkContext.getOrCreate()
sqlContext = SQLContext(sc)

df = sqlContext.createDataFrame([1,2,3], "int").toDF("value")
df.createOrReplaceTempView("df")

sqlContext.sql("SELECT * FROM df WHERE value<>1").explain()
sqlContext.sql("SELECT * FROM df WHERE value<>1").show()

sc.stop()
```

If you get the following output where the explain method prints the GPU-related stages, you can verify that your Spark application is running on GPU.

```
== Physical Plan ==
GpuColumnarToRow false
+- GpuFilter NOT (value#2 = 1), true
   +- GpuRowToColumnar targetsize(2147483647)
      +- *(1) SerializeFromObject [input[0, int, false] AS value#2]
         +- Scan[obj#1]
```

However, if you get the following output, your Spark application is not running on GPU but instead on CPU. You must ensure that Spark applications are configured properly to work on GPU.

```
== Physical Plan ==
*(1) Filter NOT (value#2 = 1)
+- *(1) SerializeFromObject [input[0, int, false] AS value#2]
   +- Scan[obj#1]
```

Spark Operator YAML Example Using GPU for Spark 3.5.0

Example:

```
apiVersion: "sparkoperator.hpe.com/v1beta2"
kind: SparkApplication
metadata:
  name:
spark-eep-gpu-350
  namespace: spark
spec:
  sparkConf:
    # Enabling RAPIDs plugin
    spark.plugins: "com.nvidia.spark.SQLPlugin"
    spark.rapids.sql.enabled: "true"
    spark.rapids.force.caller.classloader: "false"

    # GPU allocation and discovery settings
    spark.task.resource.gpu.amount: "1"
    spark.executor.resource.gpu.amount: "1"
    spark.executor.resource.gpu.vendor: "nvidia.com"
    spark.executor.resource.gpu.discoveryScript: "/opt/mapr/spark/
spark-3.5.0/examples/src/main/scripts/getGpusResources.sh"
    spark.rapids.shims-provider-override:
"com.nvidia.spark.rapids.shims.spark350.SparkShimServiceProvider"

  type: Python
  sparkVersion:3.5.0
  mode: cluster
  image: gcr.io/mapr-252711/spark-gpu-3.5.0:v3.5.0
  imagePullPolicy: Always
  mainApplicationFile: .../path/to/application.py
  restartPolicy:
    type: Never
  imagePullSecrets:
    - imagepull
  driver:
    cores: 1
    coreLimit: "1000m"
    memory: "1024m"
    labels:
      version: 3.5.0
  executor:
    cores: 1
    coreLimit: "1000m"
    instances: 1
    memory: "2G"
    labels:
      version: 3.5.0
```

Enabling GPU Support for Livy Sessions

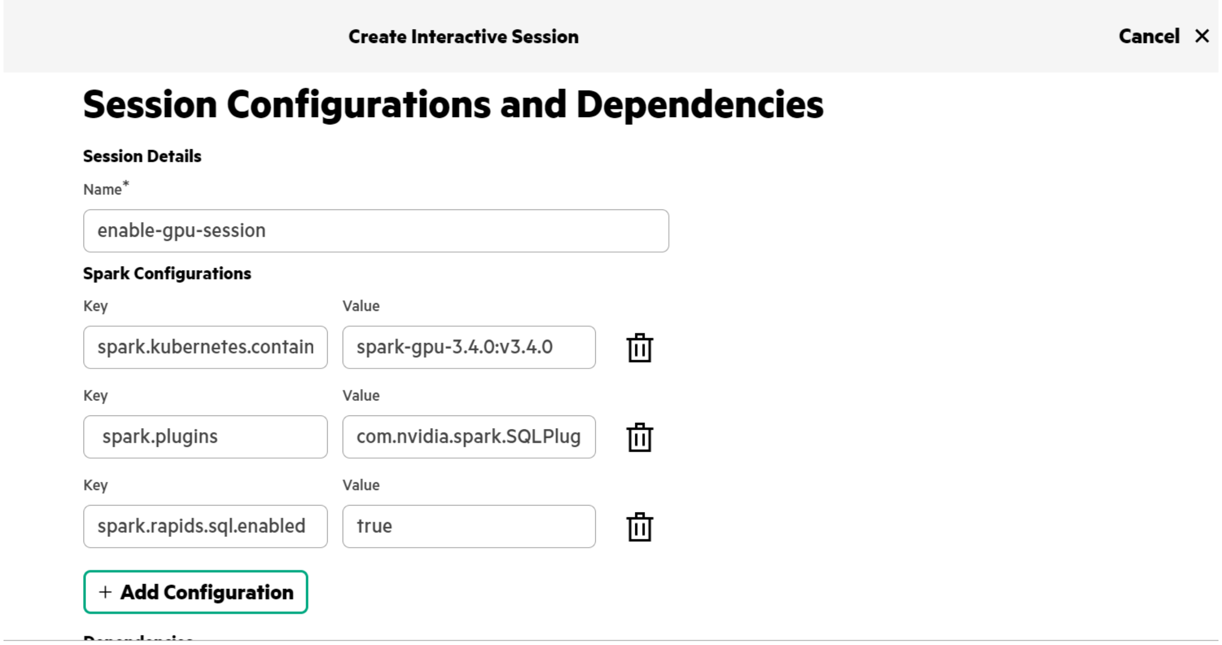
Describes how to enable and allocate GPU resources on Livy Server.

Enabling GPU Support for Livy Sessions Created Using Spark Interactive Sessions

To enable GPU processing and allocate GPU resources when using Spark interactive sessions, follow these steps:

1. Perform the creating interactive sessions instructions until you reach the **Spark Configurations** box in the **Session Configurations and Dependencies** step. See Creating Interactive Sessions on page 394.

2. Set the Spark Configurations for GPU on page 399 by providing key-value pairs. To add each Spark configurations required to run your session, click **Add Configuration**.



Session Configurations and Dependencies

Session Details

Name*: enable-gpu-session

Spark Configurations

Key	Value
spark.kubernetes.contain	spark-gpu-3.4.0:v3.4.0
spark.plugins	com.nvidia.spark.SQLPlug
spark.rapids.sql.enabled	true

3. To specify the details for other boxes or options in the **Session Configurations and Dependencies** step and to complete creating interactive sessions, see Creating Interactive Sessions on page 394.

Enabling GPU Support for Livy Sessions Created Using Notebooks

To enable GPU processing and allocate GPU resources when using Spark magic (`%manage_spark`) to create Livy sessions, follow these steps:

1. Run `%manage_spark` to connect to the Livy server and start a new session. See %manage_spark on page 447 for details.

2. Run `%config_spark` to add the Spark configurations.

3. Click the **+Add Spark Configuration Key-Value Pair** button.

4. Enter the key and value for Spark Configurations for GPU on page 399 in their respective boxes.

5. After you have finished adding the key-value pairs, click **Submit**. This will save the new Spark configuration changes to enable the GPU support for Livy sessions.

6. To specify the details for the other boxes or options in the **Create Session** step and to complete creating Livy session, see %manage_spark on page 447.

Verifying Livy Sessions are Running on GPU

To verify Livy sessions are running on GPU, you can use the explain Spark method.

Run the following PySpark application for Livy Sessions Created Using Spark Interactive Sessions:

```
sqlContext = SQLContext(sc)

df = sqlContext.createDataFrame([1,2,3], "int").toDF("value")
df.createOrReplaceTempView("df")
```

```
sqlContext.sql("SELECT * FROM df WHERE value<>1").explain()
sqlContext.sql("SELECT * FROM df WHERE value<>1").show()
```

Run the following PySpark application for Livy Sessions Created Using Notebooks:

```
from pyspark.sql import SQLContext

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

sqlContext = SQLContext(sc)

df = sqlContext.createDataFrame([1,2,3], "int").toDF("value")
df.createOrReplaceTempView("df")

sqlContext.sql("SELECT * FROM df WHERE value<>1").explain()
sqlContext.sql("SELECT * FROM df WHERE value<>1").show()
```

If you get the following output where the explain method prints the GPU-related stages, you can verify that your Livy session is running on GPU.

```
== Physical Plan ==
GpuColumnarToRow false
+- GpuFilter NOT (value#2 = 1), true
   +- GpuRowToColumnar targetsize(2147483647)
      +- *(1) SerializeFromObject [input[0, int, false] AS value#2]
         +- Scan[obj#1]
```

However, if you get the following output, your Livy session is not running on GPU but instead on CPU. You must ensure that Livy sessions are configured properly to work on GPU.

```
== Physical Plan ==
*(1) Filter NOT (value#2 = 1)
+- *(1) SerializeFromObject [input[0, int, false] AS value#2]
   +- Scan[obj#1]
```

Securely Passing Spark Configuration Values

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. You can use notebook to create secrets.

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>"
...
```

Running Spark Applications in Namespaces

Describes how namespaces work with regard to Spark applications in HPE Ezmeral Unified Analytics Software.

Information in this topic relates to Spark applications that use the HPE-curated Spark images or Spark OSS images with the security context set in the Spark application YAML, as described in Setting Security Context for Spark OSS Images on page 406.

HPE Ezmeral Unified Analytics Software users (admins and members) can submit Spark applications through the following clients and interfaces:

- HPE Ezmeral Unified Analytics Software UI
- APIs/CLI (kubectl)
- Notebooks
- Airflow DAGs

By default, when a user submits a Spark application, the Spark application runs in the user's designated namespace, isolating the user's work and resource use from other users in the HPE Ezmeral Unified Analytics Software cluster. For example, if `user01` is signed into HPE Ezmeral Unified Analytics Software and submits a Spark application, the Spark application automatically runs in the `user01` namespace. Only `user01` can access the Spark application and Spark application details in the Spark History Server UI.

Alternatively, a user can run their Spark applications in the `spark` namespace. When a user changes the namespace to `spark` in the Spark application YAML, the Spark application runs in the `spark` namespace and all users (admins and members) can access the Spark application through the HPE Ezmeral Unified Analytics Software UI. However, only the user that submitted the Spark application can access the application details in the Spark History Server UI.

NOTE: Currently, the HPE Ezmeral Unified Analytics Software UI does not support running Spark applications in the `spark` namespace. You can only run Spark applications in the `spark` namespace through kubectl, notebooks, and Airflow DAGs.

The following table describes how HPE Ezmeral Unified Analytics Software responds when you submit Spark applications through the supported clients and interfaces:

Client/Interface	Description
HPE Ezmeral Unified Analytics Software UI	• Spark applications run in the user's designated namespace. • Does not support running Spark applications in the `spark` namespace. • If a user changes the namespace in their Spark application, the system automatically reverts the namespace back to the namespace of the user submitting the Spark application. For example, if `user01` submits the Spark application as `user02,` the system automatically reverts the namespace back to `user01` and runs the application in the `user01` namespace.
API/CLI (kubectl)	• Spark applications run in the user's designated namespace. • Users can change the namespace to `spark`; Spark applications run in the `spark` namespace and become accessible to all users. • If a user changes the namespace in their Spark application, for example `user01` changes the namespace to `user02,` the system accepts the Spark application, but returns an *access denied* error.
Notebook	• Spark applications run in the user's designated namespace. • If a user changes the namespace in their Spark application, for example `user01` changes the namespace to `user02,` the system returns an *access denied* error.
Airflow DAG	• A Spark application launched through an Airflow DAG automatically runs in the namespace of the user that deployed the DAG. For example, if `user01` deploys a DAG with a Spark application in the workflow, the Spark application runs in the `user01` namespace. • Manually triggered DAGs launch in the namespace of the trigger event owner. • Scheduled DAGs launch in the namespace of the last user to un-pause the DAG.

Spark History Server

In an HPE Ezmeral Unified Analytics Software cluster, one Spark History Server runs in the `spark` namespace. Users can go to the Spark History Server UI to view a list of all Spark applications that have run. However, users can only view the details of Spark applications that they submit, regardless of the namespace they use (their own namespace or the `spark` namespace).

If a user submits a Spark application in the `spark` namespace, only that user can view the application details in the Spark History Server UI. For example, if `user01` submits a spark application in the `spark` namespace, `user02` cannot access the Spark application details in the Spark History Server UI. Only `user01` can view the Spark application details.

The system returns an unauthorized message when users try to view application details for Spark applications that were submitted by other users.

Setting Security Context for Spark OSS Images

The Spark OSS images do not contain the security context required to run Spark applications against volumes in HPE Ezmeral Unified Analytics Software. HPE Ezmeral Unified Analytics Software denies user access to the volume if it cannot authenticate the user, which results in Spark application failures.

To add security context to your Spark application, add the following configuration setting in the Spark application YAML:

```
sparkConf:
    spark.hpe.webhook.security.context.autoconfigure: "true"
```

This security context flag sets the pod security context and enables HPE Ezmeral Unified Analytics Software to recognize you as a valid HPE Ezmeral Unified Analytics Software user when you run your Spark applications.

When you add the security context flag to the Spark application YAML and run the Spark application, the application automatically runs in your user-designated namespace. If you change the namespace to `spark`, the Spark application runs in the `spark` namespace.

WARNING: Do not set the security context in HPE-Curated Spark images. Setting the security context in HPE-Curated Spark images causes Spark applications to fail.

For additional information, see User Isolation on page 224 and Setting the User Context on page 376.

Using whylogs with Spark

Note: This feature is presented as a developer preview. Developer previews are not tested for production environments, and should be used with caution.

Describes how to use whylogs with Spark.

Prerequisites
Sign in to HPE Ezmeral Unified Analytics Software as a member.

About this task

In HPE Ezmeral Unified Analytics Software, whylogs is integrated to work with Livy sessions submitted through Kubeflow notebooks using the `%manage_spark` magic function. You can use whylogs with Spark to profile, visualize, and monitor data to detect drifts.

To use whylogs with Spark, refer to the Data Validation example and WhyLogs Profiling example in GitHub. The basic steps are outlined as follows:

1. Create a notebook or import your notebook into HPE Ezmeral Unified Analytics Software. See Creating and Managing Notebook Servers on page 436.
2. Enter the `%manage_spark` command in your notebook and configure your Spark session through different tabs. You must select the authentication as Single Sign-On and the runtime language as Python. To learn about creating sessions by using `%manage_spark`, see %manage_spark on page 447.
3. Enter the `%config_spark` magic in your notebook and update the value of `spark.kubernetes.container.image` property to `gcr.io/mapr-252711/spark<version>:<image-tag>`. Click **Submit** when done. To learn about using `%config_spark`, see %config_spark on page 448.
4. Verify that your created session is in the **Idle** state. You can verify by clicking the **Manage Sessions** tab or by navigating to the **Spark Interactive Sessions** screen. See Managing Interactive Sessions on page 397.
5. Once the session is in the **Idle** state, you can set the environment variables and import the required libraries and modules from whylogs.

6. Create data frames to profile the data or validate the data with whylogs and run the notebook.
7. Once you finish running your notebook, navigate back to the HPE Ezmeral Unified Analytics Software home screen.
8. In the left navigation bar, go to **Data Engineering > Data Sources**.
9. Click **Browse**.
10. Go to the `/shared/<spark-whylogs>` folder which is a path set in your notebook to store the logs from whylogs. You can see that the data profiles and the drift summary report are stored in the shared volume in the `.html` and `.bin` formats.
11. To download a summary report, select **Download** from the **Actions** menu.

Results

You can analyze the summary report to detect drifts and monitor your data.

Data Science

Provides a brief overview of data science in HPE Ezmeral Unified Analytics Software.

Data scientists can use programming languages such as Python, R, Java, and SQL to build, train, and deploy machine learning models in HPE Ezmeral Unified Analytics Software using open-source tools that optimize the performance of predictive machine learning models.

Data scientists can use the tools provided in HPE Ezmeral Unified Analytics Software to:

- Perform exploratory data analysis in Notebooks.
- Build features or labels from the data.
- Create and train models in Notebooks or Pipelines and training frameworks like TensorFlow, Ray, or PyTorch.
- Create and run pipelines based on variable conditions for repetitive tasks.
- Run jobs across the distributed clusters or cloud burst (launch) the jobs into a separate cloud environment using APIs from Kubeflow.
- Select your model and hyperparameters for your model to run AutoML jobs by using Katib and MLflow.
- Compile the models into a container and enter the container into the registry to make it available for model serving as a part of KServe.
- Query pipelines for data drift, bias, and robustness.
- Evaluate models and replace the previous models for optimization or retrain and deploy the models for better performance.

Feast

Provides a brief overview of Feast in HPE Ezmeral Unified Analytics Software.

Feast is a feature store that configures data infrastructure for serving machine learning features and operationalizing a machine learning model. Feast tracks and defines feature metadata and enables the reusing and sharing of features across multiple teams. To learn more, see Feast.

You can interact with Feast by using the Kubeflow notebooks.

To access Feast in HPE Ezmeral Unified Analytics Software, click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Feast** tile under the **Data Science** tab and click **Open**.

The mount path for Feast is `/mnt/shared/feast-store`.

To see the files in `feast-store` folder, click **Data Engineering** **Data Sources** on the left navigation bar. Click **Browse** and then go to `shared/feast-store` folder.

Related tasks

Feast Ride Sharing Use Case on page 19
Provides an end-to-end workflow using Feast in HPE Ezmeral Unified Analytics Software to generate training data and perform online model inference for the ride-sharing driver satisfaction model.

Kubeflow

Provides a brief overview of Kubeflow in HPE Ezmeral Unified Analytics Software.

Kubeflow is the platform to develop and deploy machine learning (ML) workflows using Kubeflow components. You can create Kubeflow pipelines, manage Katib experiments, and serve ML models using Kubeflow in a fully managed and secured unified environment provided by HPE Ezmeral Unified Analytics Software.

The external link for KServe InferenceService follows the following pattern:

```
service-name.namespace.domain.com
```

A dot is used between the service name and its namespace, creating two subdomains.

Features and Functionality

Kubeflow in HPE Ezmeral Unified Analytics Software supports the following features and functionality:

- Provides a seamless SSO login experience for authorization and authentication.
- A default notebook is created with tensorflow image by using Kubeflow Notebooks. See Creating and Managing Notebook Servers on page 436.
- A default user volume is created in Kubeflow notebooks where only the current user has access to the data stored in the `user` folder.
- Kubeflow notebooks contains the `shared` directory with all the notebook examples that can be accessed by all the authorized users.

Kubeflow Components

The following are Kubeflow components:

- Central Dashboard
- Kubeflow Notebooks
- Kubeflow Pipelines
- Katib

- Training Operators

To learn more, see Kubeflow documentation.

More information

Financial Time Series Workflow on page 21
Describes how to use HPE Ezmeral Unified Analytics Software to run a Spark application from an Airflow DAG and then run a Jupyter notebook to analyze and visualize data that the Spark application puts into a shared directory in the shared volume that the data scientist's notebook is mounted to.

MNIST Digits Recognition Workflow on page 31
Provides an end-to-end workflow in HPE Ezmeral Unified Analytics Software for an MNIST digits recognition example.

Candy Sharing Tutorial (Kale) on page 18
Describes how Kale converts Notebook to pipeline by applying notebook annotations.

Kubeflow Sizing

Describes the resource allocation for different Kubeflow components.

Kubeflow sets default resource usage for each workload and component. You can customize the values for resource consumption for Katib experiments, model serving, and Kubeflow pipelines using the YAML file before applying the YAML file to a cluster. You can customize the resource consumption values for the Notebook while creating a Notebook in the Kubeflow UI.

Katib Experiments

Katib experiments create a pod for each trial, and allocates the following resources:

- vCPU: 50m
- Memory: 10Mi

Model Serving

Model serving creates a serving pod for each model, and allocates the following resources:

- vCPU: 100m
- Memory: 128Mi

Kubeflow Pipeline

Kubeflow Pipeline creates a workload pod for each step, and allocates the following resources:

- vCPU: 1
- Memory: 1Gi

Notebook

The default notebook is allocated with the following resources:

- vCPU: 1
- Memory: 2Gi

When you create a new notebook, the following resources will be allocated by default:

- vCPU: 0.5
- Memory: 1Gi

However, you can change the value of these resources during the notebook creation step in the Kubeflow UI.

Enabling GPU Support on Kubeflow Kserve Model Serving

Describes how to enable GPU support on Kubeflow Kserve model serving instance.

Prerequisites

- Sign in to HPE Ezmeral Unified Analytics Software.
- Train and save a model using the PyTorch CUDA or Tensorflow CUDA libraries.

About this task

To enable GPU support for Kubeflow Kserve model serving instance in HPE Ezmeral Unified Analytics Software, follow these steps:

Procedure

1. Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Kubeflow** tile under the **Data Science** tab and click **Open**.
2. Click **Endpoints** on the left side menubar of the **Kubeflow Central Dashboard**.
3. Click the **+ New Endpoint** button or click on your saved model.
4. Create or update the `InferenceService` yaml manifest and set `storageURI` and the corresponding type of `predictor` (tensorflow or pytorch).
5. To enable GPU, set the `resources.limits` section of the yaml as follows:

   For example:

```
 apiVersion: "serving.kserve.io/v1beta1"
kind: "InferenceService"
metadata:
  name: "tensorflow-gpu"
  namespace: "<user-name>"
spec:
  predictor:
    serviceAccountName: <service-account-name>
    tensorflow:
      storageUri: "s3://mlflow/4/4d60878e34a947b080a6015ae297aaca/
artifacts"
      resources:
        limits:
          nvidia.com/gpu: 1
```

NOTE: With MIG configuration, only one GPU can be assigned per application. For details, see GPU Support on page 137.

Results

The GPU is now enabled on Kubeflow Kserve model serving instance.

More information

GPU Support on page 137
Provides information about support for NVIDIA GPU, MIG partitioning, preparing hosts for GPU-enabled environment, adding hosts and enabling GPU in HPE Ezmeral Unified Analytics Software.

HPE Machine Learning Development Environment

Provides a brief overview of HPE Machine Learning Development Environment (HPE MLDE) in HPE Ezmeral Unified Analytics Software.

HPE Machine Learning Development Environment is a machine learning platform that offers features such as automated hyperparameter tuning as well as distributed training and scaling of computations across multiple GPUs to ensure faster model training times.

HPE MLDE supports various deep learning frameworks, enabling you to work with tools such as TensorFlow and PyTorch. It enables efficient model training and deployment, optimizing machine learning workflows. To learn more, see HPE MLDE.

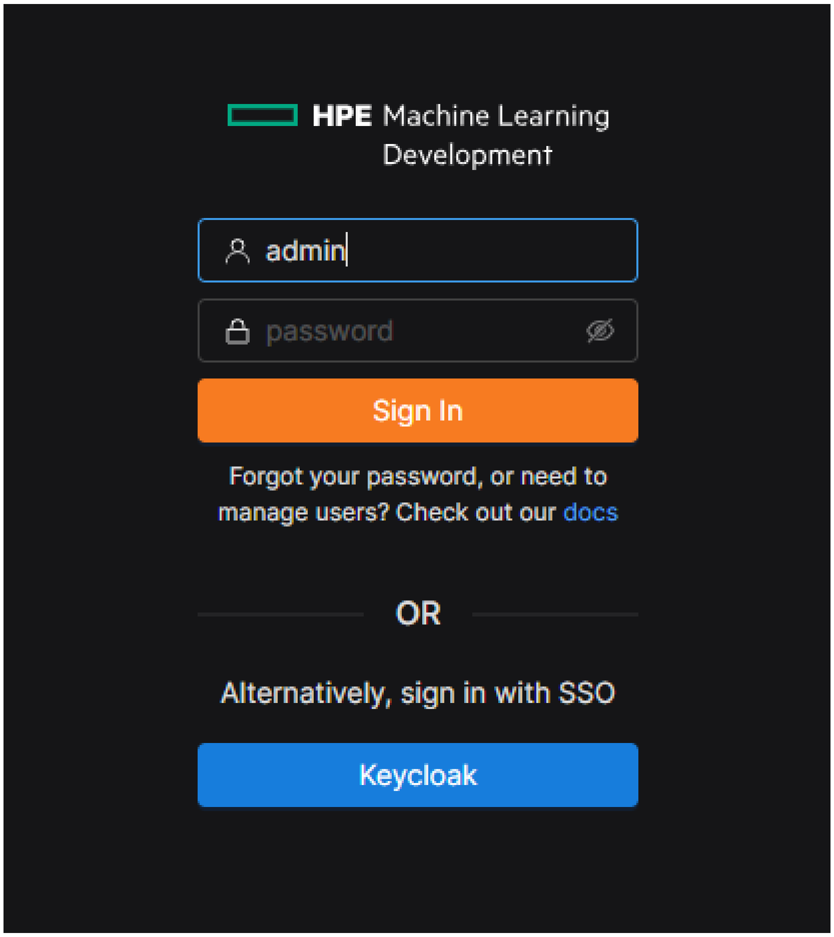
To access HPE MLDE in HPE Ezmeral Unified Analytics Software, click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **HPE MLDE** tile under the **Data Science** tab and click **Open**.

HPE MLDE User Authentication

Describes the methods of user authentication in HPE Machine Learning Development Environment.

In HPE MLDE, there are two methods of authentication:

- Authentication with SSO via Keycloak (Recommended) on page 413
- Authentication with Built-in User Profiles on page 413



admin

Forgot your password, or need to manage users? Check out our docs

Alternatively, sign in with SSO

Authentication with SSO via Keycloak (Recommended)

HPE recommends signing in to HPE MLDE with SSO by using the Keycloak button. You can sign in by using your Unified Analytics account via SSO.

If you sign in with your Unified Analytics Administrator account, you will have the ClusterAdmin role in HPE MLDE.

If you sign in with your Unified Analytics Member account, you will have the WorkspaceCreator role in HPE MLDE.

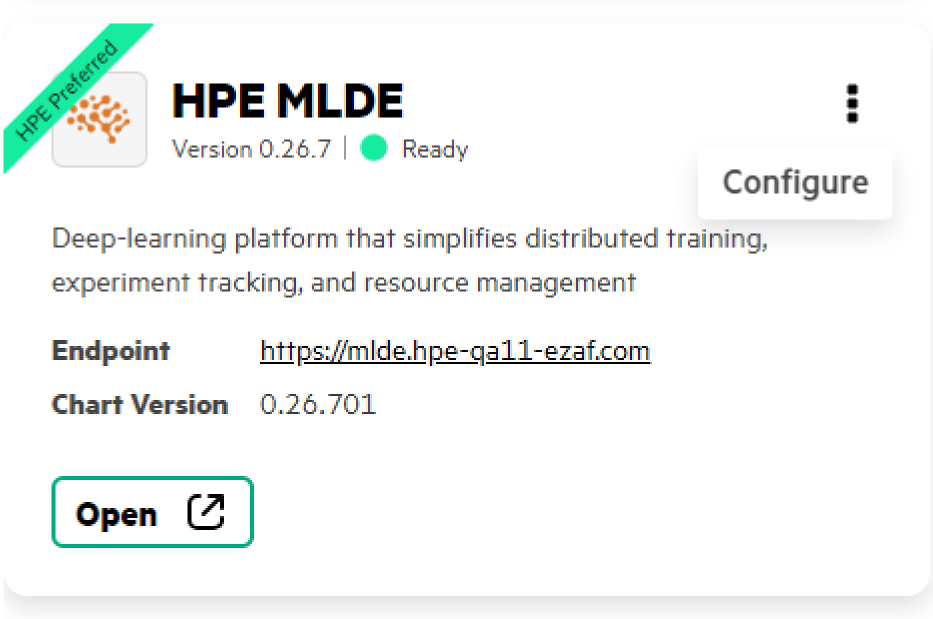
NOTE: If your username is `admin` or `determined`, you cannot sign in to HPE MLDE with SSO by using the Keycloak button. In this scenario, you must sign in with built-in user profiles by manually entering your username and password. To learn more, see Authentication with Built-in User Profiles on page 413.

Authentication with Built-in User Profiles

You can sign in to HPE MLDE by using one of the two built-in user profiles called `admin` and `determined`.

To locate the password for built-in profiles, follow these steps:

1. Sign in to HPE Ezmeral Unified Analytics Software as an Administrator.
2. Click the **Tools & Frameworks** icon on the left navigation bar.
3. Navigate to the **HPE MLDE** tile under the **Data Science** tab.
4. On the **HPE MLDE** tile, click the **three-dots** button.



HPE MLDE

Version 0.26.7 | Ready

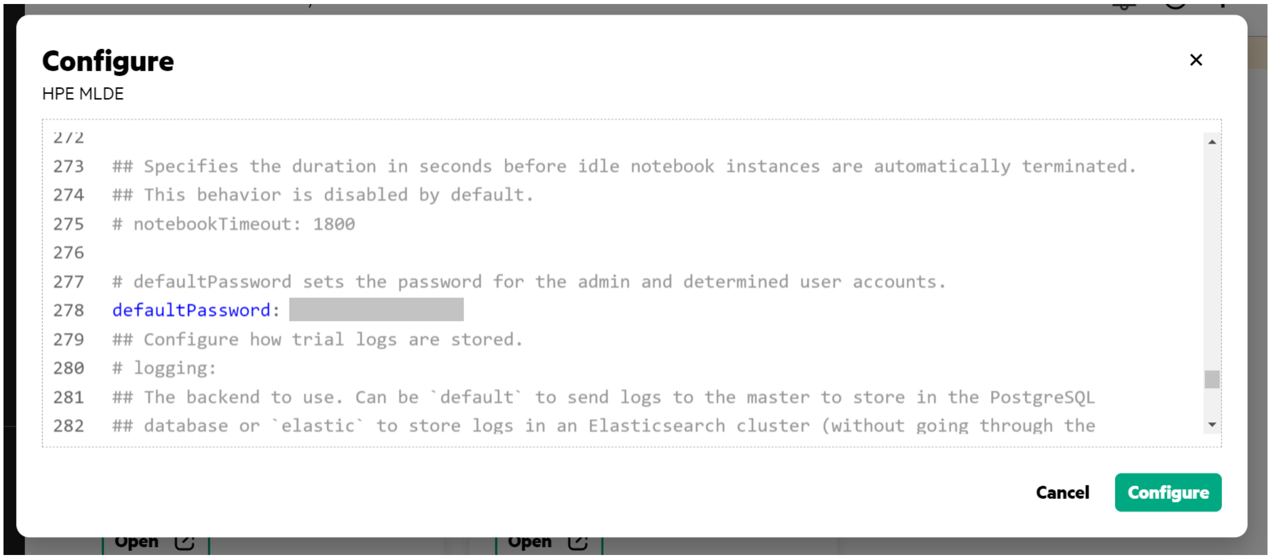
Deep-learning platform that simplifies distributed training, experiment tracking, and resource management

Endpoint https://mlde.hpe-qa11-ezaf.com

Chart Version 0.26.701

5. Select **Configure** to open the editor.

The password is available as a value for `defaultPassword`.



```

## Specifies the duration in seconds before idle notebook instances are automatically terminated.
## This behavior is disabled by default.
# notebookTimeout: 1800

# defaultPassword sets the password for the admin and determined user accounts.
defaultPassword: [redacted]
## Configure how trial logs are stored.
# logging:
## The backend to use. Can be `default` to send logs to the master to store in the PostgreSQL
## database or `elastic` to store logs in an Elasticsearch cluster (without going through the
```

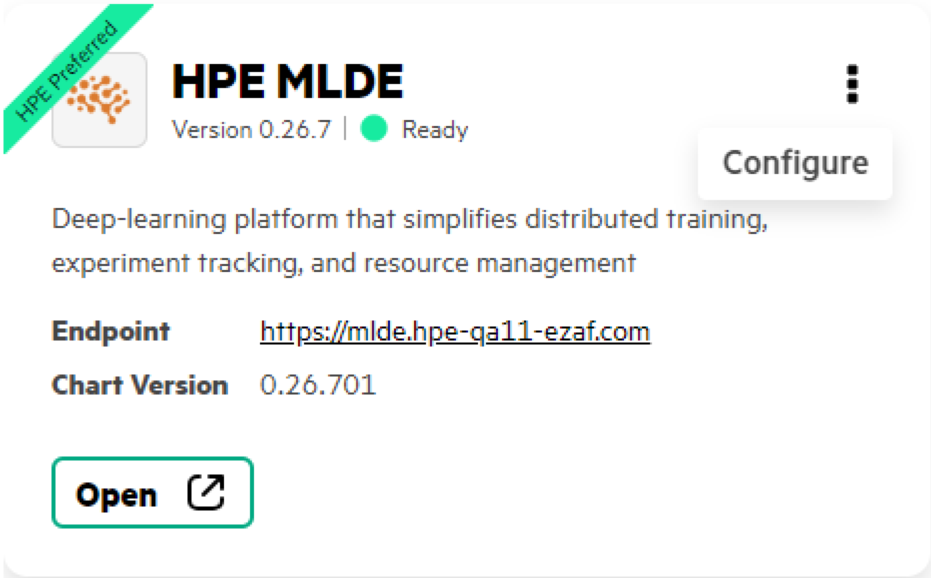
Enabling HPE MLDE in an Air-Gapped Environment

Describes how to enable HPE MLDE in an air-gapped (disconnected) environment.

In HPE Ezmeral Unified Analytics Software, HPE MLDE is disabled (deployment is scaled to zero) by default in an air-gapped environment.

To enable HPE MLDE in an air-gapped environment, follow these steps:

1. Sign in to HPE Ezmeral Unified Analytics Software as an Administrator.
2. Contact HPE support to get the HPE MLDE master enterprise image.
3. Upload the HPE MLDE master enterprise image to the airgap registry. See Using the Air Gap Utility on page 95.
4. Click the **Tools & Frameworks** icon on the left navigation bar.
5. Navigate to the **HPE MLDE** tile under the **Data Science** tab.
6. On the **HPE MLDE** tile, click the **three-dots** button.



HPE MLDE

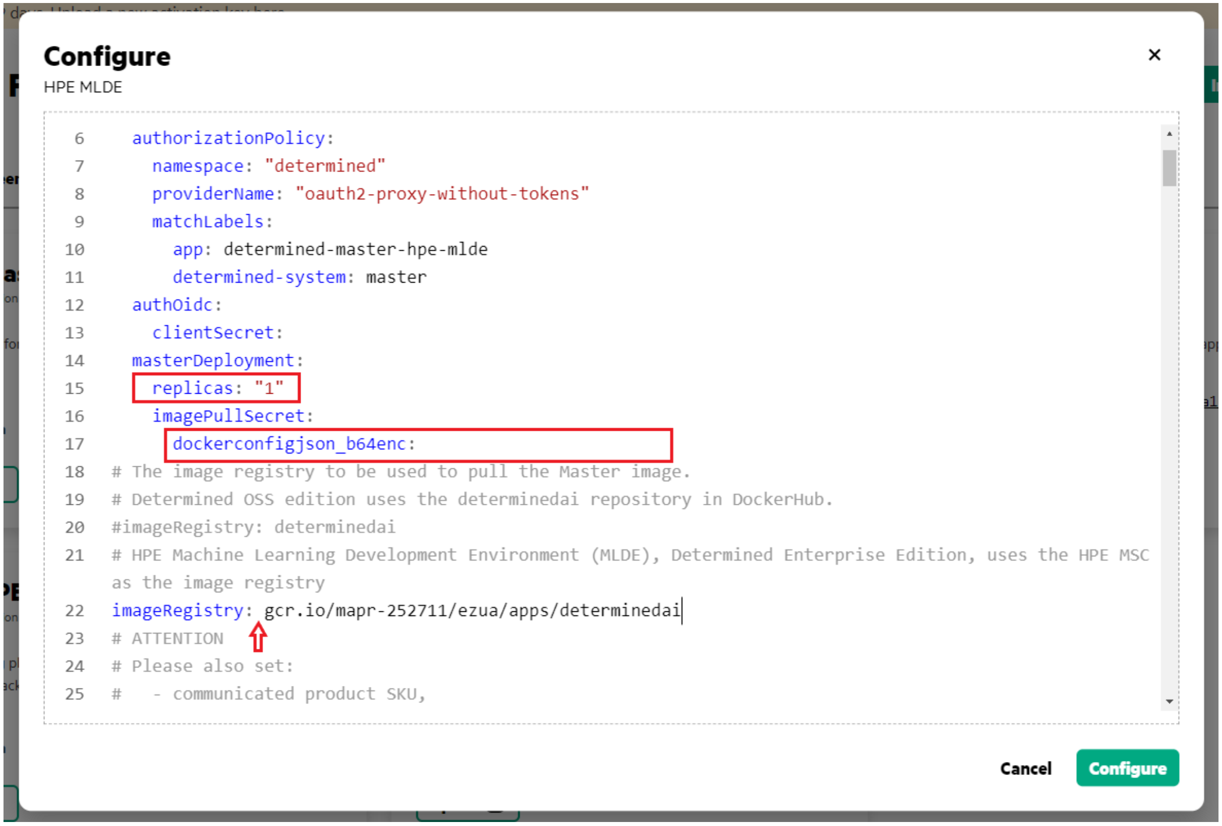
Version 0.26.7 | ● Ready

Deep-learning platform that simplifies distributed training, experiment tracking, and resource management

Endpoint https://mlde.hpe-qa11-ezaf.com

Chart Version 0.26.701

7. Select **Configure** to open the editor.



```
  authorizationPolicy:
    namespace: "determined"
    providerName: "oauth2-proxy-without-tokens"
    matchLabels:
      app: determined-master-hpe-mlde
      determined-system: master
  authOidc:
    clientSecret:
  masterDeployment:
    replicas: "1"
    imagePullSecret:
      dockerconfigjson_b64enc:
# The image registry to be used to pull the Master image.
# Determined OSS edition uses the determinedai repository in DockerHub.
#imageRegistry: determinedai
# HPE Machine Learning Development Environment (MLDE), Determined Enterprise Edition, uses the HPE MSC as the image registry
imageRegistry: gcr.io/mapr-252711/ezua/apps/determinedai
# ATTENTION
# Please also set:
#   - communicated product SKU,
```

8. Add the airgap registry address as the prefix to the `imageRegistry` option.

   For example:

```
imageRegistry: <customer-registry.com/some-folder-name>/gcr.io/
mapr-252711/ezua/apps/determinedai
```

9. Set `replicas` to `1` under the `ezua.masterDeployment.replicas` option.

10. If your airgap registry differs from the one configured for Unified Analytics and needs authentication, you can provide credentials in the base64 encoded format. Set the `dockerconfigjson_b64enc` option under the `ezua.masterDeployment.imagePullSecret` property.

Configuring HPE MLDE for Added GPU Nodes

Describes how to configure HPE MLDE for added GPU nodes in a cluster after cluster expansion.

If you add GPU nodes to the cluster after installing HPE MLDE, you must perform the following steps to ensure HPE MLDE works on these nodes.

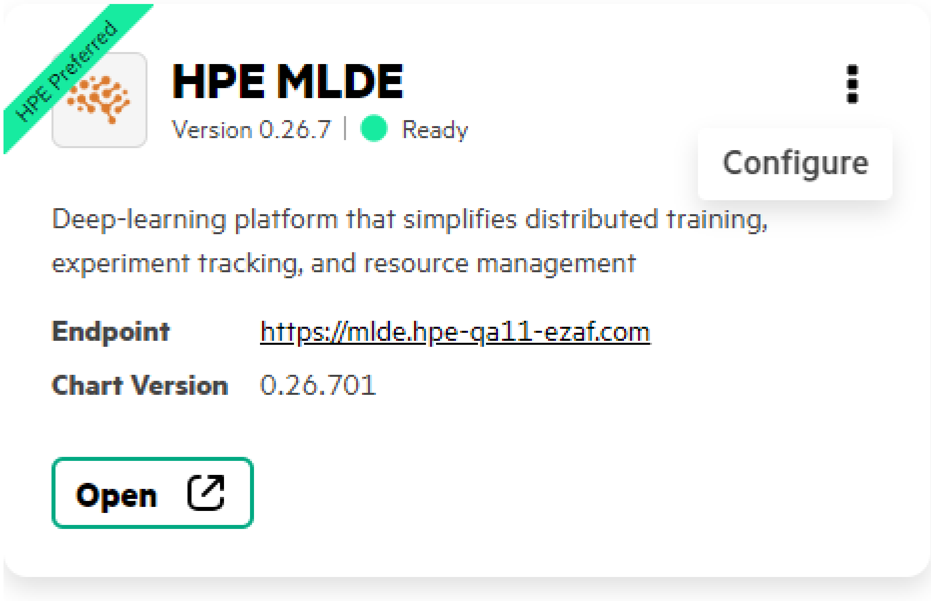
1. Sign in to HPE Ezmeral Unified Analytics Software as an Administrator.

2. To determine the maximum number of GPUs per node in the cluster, run the following command in the shell with a configured kubeconfig.

```
kubectl get nodes -l nvidia.com/gpu.count -o json | jq '.items |
            map(select(.status.capacity."nvidia.com/gpu")
| .status.capacity."nvidia.com/gpu" | tonumber) | max // 0'
```

3. Click the **Tools & Frameworks** icon on the left navigation bar.

4. Navigate to the **HPE MLDE** tile under the **Data Science** tab.
5. On the **HPE MLDE** tile, click the **three-dots** button.



HPE MLDE

Version 0.26.7 | ● Ready

Deep-learning platform that simplifies distributed training, experiment tracking, and resource management

Endpoint https://mlde.hpe-qa11-ezaf.com

Chart Version 0.26.701

6. Select **Configure** to open the editor.
7. Set `maxSlotsPerPod` to the maximum number of GPUs per node in the cluster.

Related concepts

Expanding the Cluster on page 113
Describes how to add additional user-provided hosts to the management cluster to increase resource capacity and how to expand the cluster to include the additional user-provided hosts.

MLflow

Provides a brief overview of MLflow in HPE Ezmeral Unified Analytics Software.

MLflow is an open-source platform that manages the end-to-end machine learning lifecycle, including experimentation, reproducibility, deployment, and a central model registry. You can train your ML model and run ML experiments in a fully managed and secured unified environment provided by HPE Ezmeral Unified Analytics Software. To learn more, see open-source MLflow documentation.

The model management framework with MLflow integration in HPE Ezmeral Unified Analytics Software is offered with the following capabilities.

Notebook Integration	Build and Train ML models using MLFlow APIs with an underlying tracking server.
Experiment Tracking	Track experiments and compare the output parameters for various runs.
MLflow Models	Enables users to log all parameters, save artifacts, load models, and deploy models.
Model Artifacts	Log params and save model artifacts to HPE Ezmeral Data Fabric Object Store.
MLflow Registry	A centralized model store, set of APIs, and UI, to collaboratively manage the full lifecycle of an MLflow Model.

Exploring MLflow in HPE Ezmeral Unified Analytics Software

HPE Ezmeral Unified Analytics Software includes sample files and data that you can access through the notebook server instance.

To access the sample files in your notebook server instance:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. In the left navigation pane, click **Notebooks**.
3. Connect to your notebook server instance.
4. To access the sample files, navigate to the `mlflow` folder in the `/<username>` directory.

   TIP: If the `/user` directory does not contain the sample files, copy the sample files from the `/shared/mlflow` folder to the `/username` directory. The `/shared` directory is accessible to all users. Editing or running examples from the `/shared` directory is not advised. The `/username` directory is specific to you and cannot be accessed by other users.

Defining RBACs on MLflow Experiments

Describes role-based access controls (RBACs) with respect to MLflow in HPE Ezmeral Unified Analytics Software and how to define RBACs to permit access to experiments in MLflow.

Role-based access controls (RBACs) are an authorization system based on policies, user roles, and bindings between the roles and policies that protect resources. With the introduction of RBACs, HPE Ezmeral Unified Analytics Software users (admins and members) can define access controls on their experiments through the MLflow API or SDK.

User access to MLflow is granted when a user makes a request to the MLflow server. A user is automatically authenticated and granted access to MLflow based on their user role in HPE Ezmeral Unified Analytics Software, as either an admin or a member.

Admins can add users through the HPE Ezmeral Unified Analytics Software UI, as described in Adding and Removing Users on page 233 and User Roles on page 226.

Admin Role

The following list describes *admin* access and the *admin-related* tasks that impact users in MLflow:

- Admins can view and edit all experiments in MLflow regardless of the access controls set. For example, if the `NO_PERMISSIONS` access control is defined in an experiment, admins can still access the experiment.
- Admins can change a user's role in HPE Ezmeral Unified Analytics Software to *admin*. When a user has the *admin* role in HPE Ezmeral Unified Analytics Software, that user can access all existing experiments in MLflow. If the *admin* role is removed from the user (reverted back to *member*), the user cannot see any experiments created by other users.

  NOTE: By default, the MLflow default admin user is disabled to prevent any security issues, such as the plain text password being stored in open-source code.

Member Role

The following list describes MLflow access for *members*:

- By default, members have full control over the experiments they create. When a member creates an experiment, the experiment has the `MANAGE` permission set. The `MANAGE` permission enables the experiment owner to grant other users access to their experiment through access controls.
- Members cannot access experiments created by other users unless explicitly permitted to do so by the experiment owner through access controls set in the experiment.

- If an HPE Ezmeral Unified Analytics Software admin changes a member's role to *admin* in the HPE Ezmeral Unified Analytics Software UI, the user is granted full access to all experiments in MLflow.
- After deleting and re-adding a member user in the **Administration->Identity & Access Management** screen, previously granted MLflow experiment and model permissions remain intact for members. For example, if you previously created an MLflow experiment and granted the bob user the READ privilege, then deleted and re-added the bob user, the READ privilege for the MLflow experiment will persist for the bob user.

  HPE Ezmeral Unified Analytics Software does not delete user experiment or model permission objects associated with the user during a hard delete. Unified Analytics retains the associated permissions despite the user's deletion. For details, see MLflow Server Auth Initialization Code and MLflow Auth Service Client Documentation.

  To ensure that all user permissions are correctly removed when deleting a user, you must explicitly delete all related permissions as follows:

  - Use `delete_experiment_permission` to remove the user's access to any experiments. See delete experiment permissions.
  - Use `delete_registered_model_permission` to remove the user's access to any registered models. See delete registered model permissions.

  By explicitly deleting these permissions, you can ensure that re-adding the user does not unintentionally restore their previous access privileges.

CAUTION: HPE only supports user role changes made through the HPE Ezmeral Unified Analytics Software UI. Role changes made in HPE Ezmeral Unified Analytics Software are automatically propagated to MLflow. HPE does not support role changes made directly in MLflow because the changes do not propagate back to HPE Ezmeral Unified Analytics Software, which can cause unexpected system behaviors.

Supported Access Controls

HPE Ezmeral Unified Analytics Software supports the following access controls on experiments:

Access Control Type	Access Control Value	Description
None	NO_PERMISSIONS	Only the experiment creator and admins can access the experiment. Returns an "access denied" message when unauthorized users try to access the experiment.
Manage	MANAGE	Default permission set on an experiment at the time of creation. Only the experiment creator and admins can access the experiment. You cannot set this access control on any existing experiments.
Read	READ	The experiment creator has full access to the experiment. Specified users can only view the experiment in MLflow.
Modify	EDIT	Experiment creator has full access to the experiment. Specified users modify the experiment in MLflow.
Delete	DELETE	Only admin users can use DELETE to remove permissions on an experiment.

Defining Access Controls on Users

To permit access to experiments, use the MLflow API or SDK in your MLflow experiments to define access controls on users.

MLflow provides an `AuthServiceClient` that implements CRUD functionality for `experiment_permission` and `model_permission` objects.

Use the following code examples as a guide to define access controls on users.

Required code to set access controls on an experiment

```
from mlflow.server.auth.client import
AuthServiceClient
user = "<username>"
permission = "<access_control>"
exp_id =
mlflow.get_experiment_by_name(experime
nt_name).experiment_id
client = AuthServiceClient("http://
mlflow.mlflow.svc.cluster.local:5000")
```

Create permission

```
permission = "READ"
exp_permission =
client.create_experiment_permission(ex
p_id, user, permission)
```

Modify permission

```
permission = "EDIT"
exp_permission =
client.update_experiment_permission(ex
p_id, user, permission)

permission = "NO_PERMISSIONS"
exp_permission =
client.update_experiment_permission(ex
p_id, user, permission)
```

Delete permission

```
exp_permission =
client.delete_experiment_permission(ex
p_id, user, permission)
client.get_user( 'admin' )
```

Using whylogs with MLflow

Note: This feature is presented as a developer preview. Developer previews are not tested for production environments, and should be used with caution.

Describes how to use whylogs with MLflow.

Prerequisites

Sign in to HPE Ezmeral Unified Analytics Software as a member.

About this task

In HPE Ezmeral Unified Analytics Software, whylogs is integrated with MLflow to log and analyze the data quality. You can use whylogs to analyze the data quality throughout the machine learning lifecycle.

To use whylogs with MLflow, refer to the MLflow logging example in the GitHub. The basic steps are outlined as follows:

1. Create a notebook or import the notebook into HPE Ezmeral Unified Analytics Software. See Creating and Managing Notebook Servers on page 436.
2. Import the required libraries and modules from whylogs.
3. Train a model and create data frames to profile the data, and then run the notebook.
4. Once you finish running your notebook, navigate back to the HPE Ezmeral Unified Analytics Software home screen.
5. Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **MLflow** tile under the **Data Science** tab and click **Open**.
6. View the whylogs output in the `whylogs` directory within that run’s artifacts in the MLflow UI.

Results

You can analyze the data quality metrics and ensure the data quality by using whylogs ouput.

Ray

Provides a brief overview of Ray in HPE Ezmeral Unified Analytics Software.

Ray is a unified framework for scaling AI/ML and Python applications, handling distributed workloads, and parallelizing serial applications. As a distributed computing framework, Ray simplifies scalability and fault tolerance. Ray offers flexible programming for parallel tasks and actors, making it suitable for data processing, reinforcement learning, and simulation.

To learn about API changes for Ray 2.0, see Ray 2.0 Migration Guide.

Ray Core	Ray Core provides core primitives to build and scale distributed applications. The core primitives are: • Tasks • Actors • Objects
Ray Libraries	HPE Ezmeral Unified Analytics Software supports the following Ray libraries: • Ray Serve. See Rent Forecasting Model (Ray Serve) on page 39. • Ray Tune. See Running Independent Tune Trials (Ray Tune) on page 55
Purpose	• Simplify development by providing high-level abstractions and automatic management of complex distributed systems. • Accelerate the development process by reducing the complexity of building distributed systems.
Use Cases	• Data Processing: Efficiently handle large-scale data processing tasks.

- Reinforcement Learning: Scale RL experiments across multiple machines for faster learning.
- High-Performance Computing: Parallelize complex computations for faster execution in HPC scenarios.
- Event-driven and Real-time Systems: Process events or data streams in parallel for timely processing.

Features and Functionality

Ray in HPE Ezmeral Unified Analytics Software supports the following features and functionality:

Ray Cluster Reconciliation

HPE Ezmeral Unified Analytics Software provides an automatic Ray cluster reconciliation feature using Helm hooks.

When you upgrade Ray in HPE Ezmeral Unified Analytics Software, all Ray workloads, including head nodes, workgroup nodes, small group nodes, and computational resources such as CRDs, config maps, services, and others are managed autonomously.

The Ray cluster reconciliation feature improves the user experience for AI application development.

Notebook Integration

A pre-existing image is created in Kubeflow notebooks with Ray library. See Creating and Managing Notebook Servers on page 436.

To submit jobs using Ray, you can connect to Ray cluster. See Connecting to Ray Cluster on page 422.

Ray Dashboard

Ray dashboard in HPE Ezmeral Unified Analytics Software allows you to:

- Understand Ray memory utilization and debug memory errors.
- See per-actor resource usage, executed tasks, logs, and more.
- View cluster metrics.
- Kill actors and profile your Ray jobs.
- See errors and exceptions at a glance.
- View logs across many machines in a single pane.
- See Ray Tune jobs and trial information.

To access Ray dashboard, click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Ray** tile under the **Data Science** tab and click **Open**.

To enable Metrics view in Ray dashboard, see Enabling Metrics in the Ray Dashboard on page 423.

Security

To configure Ray to use TLS authentication for client-server communication, see TLS Authentication.

To learn more about Ray, see Ray documentation.

Connecting to Ray Cluster

Describes how to connect to Ray clusters to submit jobs.

NOTE: The Ray Client has multithreading and connection issues which impact its reliability and submitting Ray job using Ray Client is an outdated method. Hewlett Packard Enterprise recommends using `JobSubmissionClient` to submit Ray jobs. For details, see Using JobSubmissionClient to Submit Ray Jobs on page 423.

To submit jobs using Ray, you can connect to Ray cluster in two different ways:

Connecting to Ray in HPE Ezmeral Unified Analytics Software

To connect to Ray in HPE Ezmeral Unified Analytics Software, run:

```
ray.init(address="ray://
kuberay-head-svc.kuberay:10001")
```

Connecting to Ray from outside of HPE Ezmeral Unified Analytics Software

To connect to Ray cluster from outside of HPE Ezmeral Unified Analytics Software, perform the following steps:

1. To change service type to NodePort, run:

   ```
   kubectl -n kuberay edit service
   kuberay-head-svc
   ```

   Output:

   ```
   spec:
   ...
     type: NodePort
   ...
   ```

2. To get the cluster master IP, run:

   ```
   kubectl cluster-info
   ```

3. To get the client port, run:

   ```
   kubectl -n kuberay describe
   service kuberay-head-svc
   ```

   Output:

   ```
   ...
   Port:                     client
   10001/TCP
   TargetPort:               10001/TCP
   NodePort:                 client
   31536/TCP
   Endpoints:
   10.244.1.85:10001
   ```

4. Connect through `<K8 Master IP>:<Client Port>`.

   ```
   ray.init(address="ray://<K8 Master
   IP>:31536")
   ```

Using `JobSubmissionClient` to Submit Ray Jobs

Describes how to connect to Ray cluster and submit Ray jobs using `JobSubmissionClient`.

The Ray Client has multithreading and connection issues which impact its reliability and submitting Ray job using Ray Client is an outdated method. Hewlett Packard Enterprise recommends using `JobSubmissionClient` to submit Ray jobs.

To submit Ray jobs using `JobSubmissionClient`, you must specify entry point resources as follows:

- For CPU, set `entrypoint_num_cpus` to 1 or <M>
- For GPU, set `entrypoint_num_gpus` to 1 or <M>

NOTE: The failure to specify entry point resources before submitting any jobs in the Ray cluster results in unexpected behavior.

To learn how to submit Ray jobs using `JobSubmissionClient`, see Independent Tune Trials.

Example:

The following code block shows the sample code for connecting to the Ray cluster and submitting Ray Jobs using `JobSubmissionClient`:

```
import ray
from ray.job_submission import JobSubmissionClient
import time

# Ray cluster information
ray_head_ip = "kuberay-head-svc.kuberay.svc.cluster.local"
ray_head_port = 8265
ray_address = f"http://{ray_head_ip}:{ray_head_port}"

# Submit Ray job using JobSubmissionClient
client = JobSubmissionClient(ray_address)
job_id = client.submit_job(
    entrypoint="python demo.py",
    runtime_env={
        "working_dir": "./",
        # "excludes": ['']
    },
    entrypoint_num_cpus = 3
)

print(client.__dict__)
print(f"Ray job submitted with job_id: {job_id}")
```

Enabling Metrics in the Ray Dashboard

Describes how to enable metrics in the Ray dashboard.

Prerequisites

- Ensure that Ray's head pod has enough resources to run the Grafana server.

- By default, the Ray head node is adequately provisioned for Grafana. However, resource needs vary based on the intensity of Ray job submissions. Although the head node does not directly run jobs, large file submissions can strain object memory.

  Therefore, the adequacy of resources depends on the specific case. You must monitor performance and adjust resources to find the optimal balance for your specific use case.

  To find the minimum resource requirements for Grafana, see Grafana minimum system resources.
- To configure the Ray resources, see Configuring Resources in the UI on page 427.

About this task

To enable Metrics view in dashboard, you must install Grafana, configure the data source as centralized Prometheus, and start the Grafana server with the specific configuration file in Ray's head pod.

Procedure

1. By default, Ray's metrics are scraped by centralized Prometheus, so specify Prometheus' service URL as the data source in `/tmp/ray/session_latest/metrics/grafana/provisioning/datasources/default.yml` file.

   For example:

   ```
   apiVersion: 1

   datasources:
     - name: Prometheus
       url: http://
   af-prometheus-kube-prometh-prometheus.prometheus.svc.cluster.local:9090
       type: prometheus
       isDefault: true
       access: proxy
   ```

2. Install Grafana in Ray's head pod and navigate to Grafana's home directory.

   To install Grafana in Ray's head pod, follow these steps:

   a) Access the shell on the head node.

   ```
   kubectl -n kuberay exec -it <head_pod_name> -- bash
   ```

   b) Download Grafana.

   ```
   wget https://dl.grafana.com/oss/release/
   grafana-9.3.6.linux-amd64.tar.gz
   ```

   c) Go to the Grafana home directory.

   ```
   tar -zxvf grafana-9.3.6.linux-amd64.tar.gz
   cd grafana-9.3.6
   ```

   d) Start the Grafana server with the Ray configuration file.

   ```
   ./bin/grafana-server --config /tmp/ray/session_latest/metrics/grafana/
   grafana.ini web
   ```

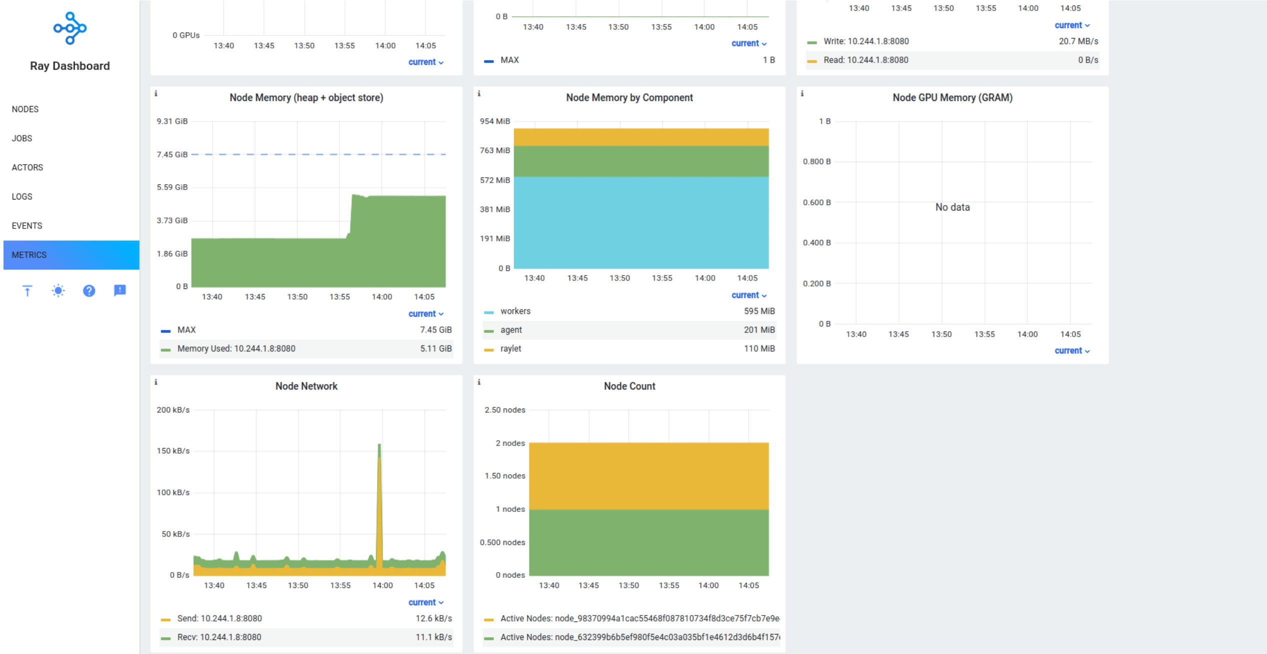
3. Forward Grafana's default port.

```
kubectl -n kuberay port-forward --address 0.0.0.0 <head_pod_name>
3000:3000
```

4. Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Ray** tile under the **Data Science** tab and click **Open**.

Results

Metrics view is enabled in the Ray dashboard.



To learn more, see Ray Metrics.

Resource Configuration and Management

Describes resource configuration and management for Ray.

Resource Configuration

In HPE Ezmeral Unified Analytics Software, a Ray cluster is deployed using KubeRay Operator.

Currently, the Ray cluster consists of a single head node and a single operator node. Auto-scaling is enabled by default for worker nodes, the Ray cluster automatically scales up and down based on resource demand.

When there is no workload, the Ray cluster has a head, and an operator node as follows:

```
> kubectl -n kuberay get pod
NAME                                              READY    STATUS     RESTARTS
AGE
kuberay-operator-6c75647d8b-7mpqp                 1/1      Running    0
22h
ray-cluster-kuberay-head-gw8lc                    2/2      Running    0
22h
```

When a submitted job demands more resources than the cluster current resources, then the auto scaler will create two more pods.

Auto-scaling is enabled by default configuration so that the Ray cluster creates two more worker pods when needed. If a pod stays idle for 60 seconds, then the auto scaler destroys it.

Upper resource limits for pods type are as follows:

- Head pod: 2 CPU and 8 GB memory.
- Worker pod: 3 CPU and 8 GB memory.

Resource Management

While running a heavy workload, you might get an Out of Memory exception. To avoid the out-of-memory exception, there are two best practices:

Memory Aware Scheduling

By default, Ray does not consider the potential memory usage of a task or an actor when scheduling as it cannot estimate beforehand how much memory is required by the task or actor. However, if you know how much memory a task or an actor might require, you can specify it in the resource requirements of `ray.remote` decorator to enable memory-aware scheduling.

For example:

```
# reserve 500MiB of available memory
to place this task
@ray.remote(memory=500 * 1024 * 1024)
def some_function(x):
  pass
```

```
# reserve 2.5GiB of available memory
to place this actor

@ray.remote(memory=2500 * 1024 *
1024)
class SomeActor(object):
   def __init__(self, a, b):
    pass
```

Scheduling Strategies

There are two scheduling strategies in Ray:

Default

Ray uses `DEFAULT` as the default strategy. Currently, Ray assigns tasks or actors on nodes until the resource utilization is beyond a certain threshold and spreads them afterward.

For example:

```
@ray.remote
def func():
  return 1
```

Spread

Ray uses `SPREAD` strategy to spread tasks or actors among available nodes.

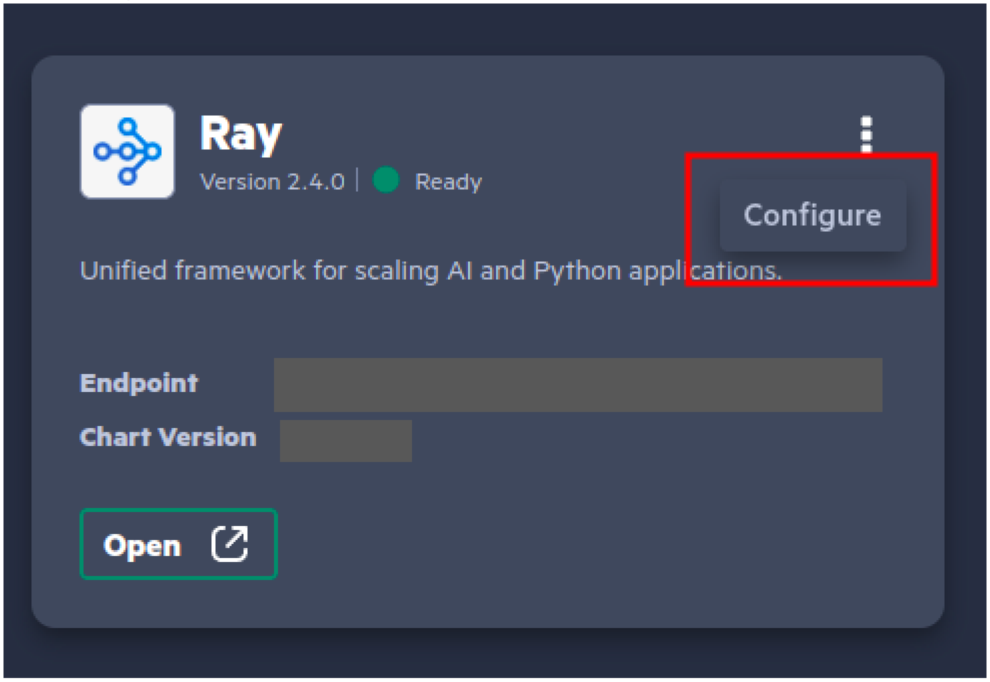
For example:

```
@ray.remote(sched
uling_strategy="S
PREAD")
def
spread_func():
  return 2
```

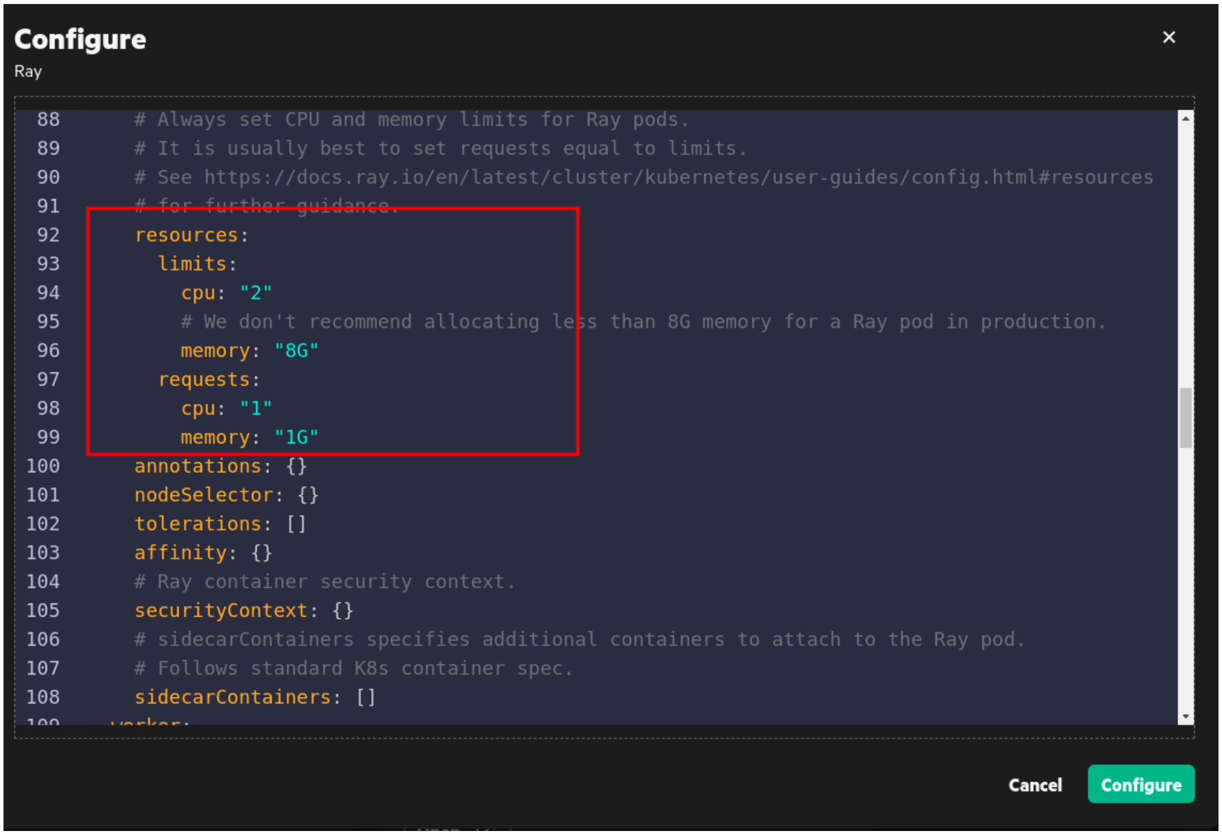
To learn more see Scheduling Strategies.

Configuring Resources in the UI

1. Sign in to HPE Ezmeral Unified Analytics Software as an Administrator.
2. Click the **Tools & Frameworks** icon on the left navigation bar.
3. Navigate to the **Ray** tile under the **Data Science** tab.
4. On the **Ray** tile, click the **three-dots** button.



5. Select **Configure** to open the editor.
6. In the editor, modify the `resources` section to adjust resources.



```
    # Always set CPU and memory limits for Ray pods.
    # It is usually best to set requests equal to limits.
    # See https://docs.ray.io/en/latest/cluster/kubernetes/user-guides/config.html#resources
    # for further guidance.
    resources:
      limits:
        cpu: "2"
        # We don't recommend allocating less than 8G memory for a Ray pod in production.
        memory: "8G"
      requests:
        cpu: "1"
        memory: "1G"
    annotations: {}
    nodeSelector: {}
    tolerations: []
    affinity: {}
    # Ray container security context.
    securityContext: {}
    # sidecarContainers specifies additional containers to attach to the Ray pod.
    # Follows standard K8s container spec.
    sidecarContainers: []
  worker:
```

GPU Support for Ray

Describes how to enable GPU, configure the GPU resources, and disable GPU for Ray.

Sign in as Administrator to HPE Ezmeral Unified Analytics Software to enable GPU to submit GPU-accelerated jobs with Ray.

You can enable GPU support for Ray in two different ways:

- Enabling GPU support during HPE Ezmeral Unified Analytics Software installation.
- Enabling GPU support after HPE Ezmeral Unified Analytics Software installation.

Enabling GPU Support During HPE Ezmeral Unified Analytics Software Installation

To enable the GPU for Ray during the HPE Ezmeral Unified Analytics Software installation, see GPU Support on page 137.

If you enabled GPU during the platform installation, you do not need to separately enable GPU for Ray. The platform installation automatically enables GPU for all applications and frameworks including Ray.

Enabling GPU Support and Configuring Resources After HPE Ezmeral Unified Analytics Software Installation

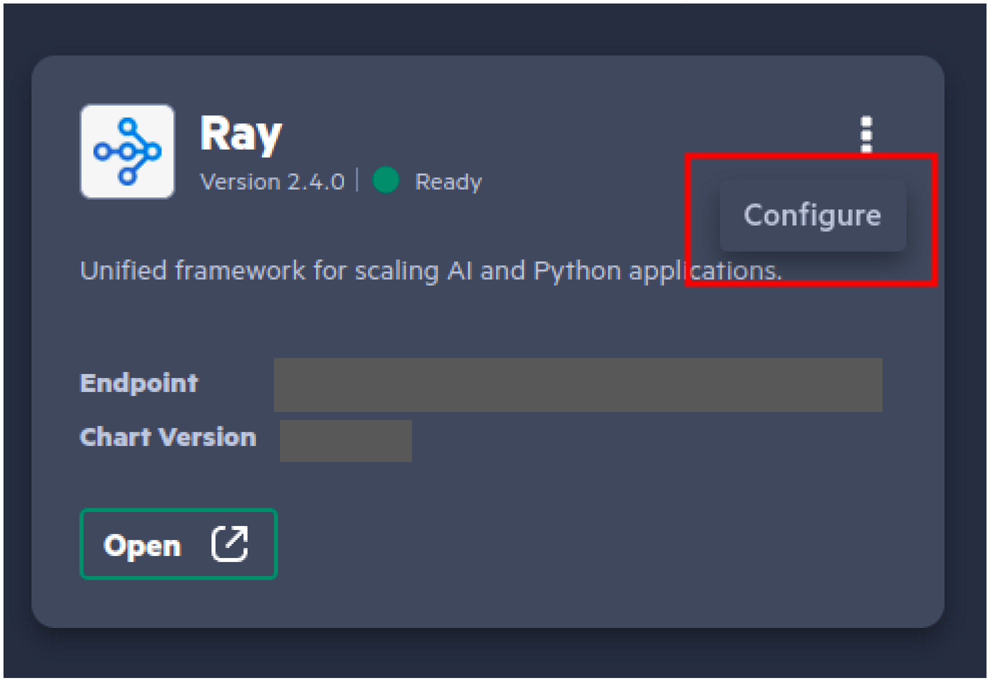
Before enabling the GPU support, when Ray is in an idle state, there are two pods running:

```
> kubectl -n kuberay get pod
NAME                                 READY   STATUS    RESTARTS   AGE
kuberay-head-5c2jj                   2/2     Running   0          10m
kuberay-operator-7b976fdb86-x5k4c    1/1     Running   0          10m
```

The operator pod creates the head pod and monitors the cluster. The head pod is the cluster master and generates additional small worker pods as required.

To enable GPU support for Ray after HPE Ezmeral Unified Analytics Software installation, follow these steps:

1. Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Ray** tile under the **Data Science** tab.
2. Click the **three dots** menu on the Ray tile and click **Configure**.

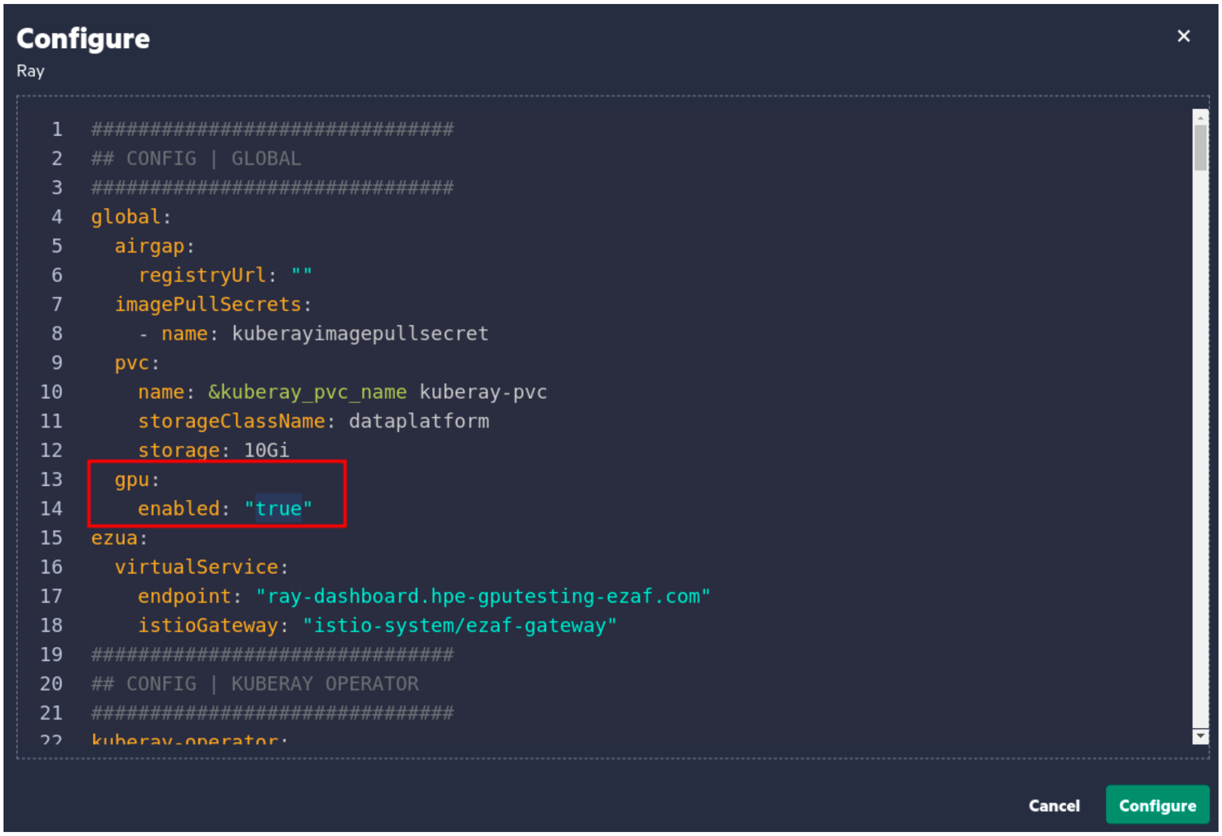


Unified framework for scaling AI and Python applications.

Endpoint [redacted]

Chart Version [redacted]

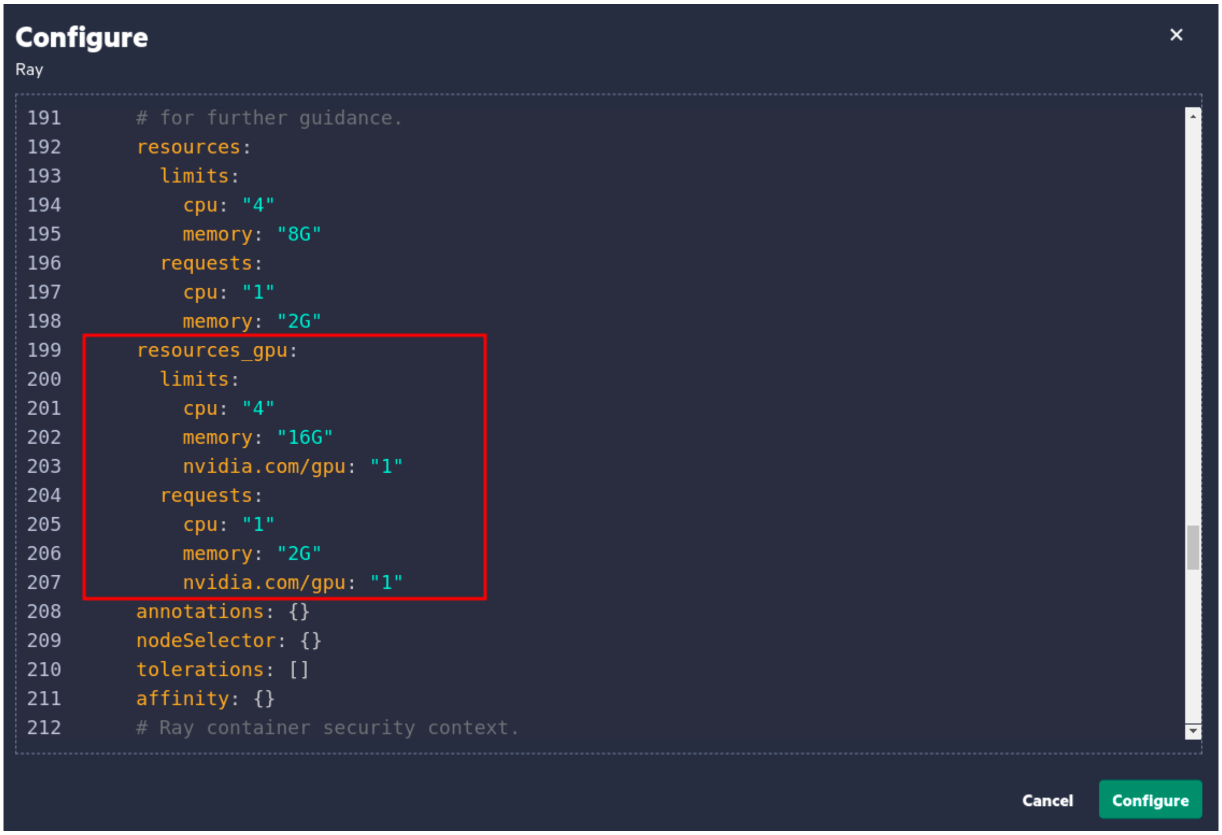
3. Set the value of `gpu.enabled` to `true`.



```
###############################
## CONFIG | GLOBAL
###############################
global:
  airgap:
    registryUrl: ""
  imagePullSecrets:
    - name: kuberayimagepullsecret
  pvc:
    name: &kuberay_pvc_name kuberay-pvc
    storageClassName: dataplatform
    storage: 10Gi
  gpu:
    enabled: "true"
ezua:
  virtualService:
    endpoint: "ray-dashboard.hpe-gputesting-ezaf.com"
    istioGateway: "istio-system/ezaf-gateway"
###############################
## CONFIG | KUBERAY OPERATOR
###############################
kuberay-operator:
```

4. (Optional) Modify the available resources as required by updating the values within the `resources_gpu` section.

   NOTE: With MIG configuration, only one GPU can be assigned per application. To learn more on what happens when you assign more than one GPU to the Ray cluster, see GPU on page 165. For details regarding GPU, see GPU Support on page 137.



Configure

Ray

```
    # for further guidance.
    resources:
      limits:
        cpu: "4"
        memory: "8G"
      requests:
        cpu: "1"
        memory: "2G"
    resources_gpu:
      limits:
        cpu: "4"
        memory: "16G"
        nvidia.com/gpu: "1"
      requests:
        cpu: "1"
        memory: "2G"
        nvidia.com/gpu: "1"
    annotations: {}
    nodeSelector: {}
    tolerations: []
    affinity: {}
    # Ray container security context.
```

5. Click **Configure**.

Results:

The GPU is now enabled on Ray. After enabling GPU, Ray creates more pods as follows:

```
> kubectl -n kuberay get pod
NAME                                    READY   STATUS     RESTARTS    AGE
kuberay-head-5c2jj                      2/2     Running    0           10m
kuberay-operator-7b976fdb86-x5k4c       1/1     Running    0           10m
kuberay-worker-smallgroup-xhhbq         1/1     Running    0           10m
kuberay-worker-workergroup-rdptj        1/1     Running    0           13s #New
pod with GPU resources!
```

You can also see the new pod with GPU resources on the Ray Dashboard.

Unlike the `worker-smallgroup` pod, the `worker-workergroup` pod cannot be scaled using an autoscaler. When GPU-accelerated jobs are submitted, the `worker-workergroup` pod handles the workload. Simultaneously, Ray manages regular jobs by using the `worker-smallgroup` pod.

Submitting GPU-Accelerated Jobs to the Ray Cluster

To submit the GPU-accelerated jobs, specify the following resource requirements:

```
@ray.remote(num_gpus=1)
def use_gpu():
    print("ray.get_gpu_ids(): {}".format(ray.get_gpu_ids()))
    print("CUDA_VISIBLE_DEVICES:
{}".format(os.environ["CUDA_VISIBLE_DEVICES"]))
```

The function `use_gpu` does not use any GPUs directly. Instead, Ray schedules it on a node with at least one GPU and allocates one GPU specifically for its run. However, it is up to the function to utilize the GPU, which is typically done through an external library such as TensorFlow.

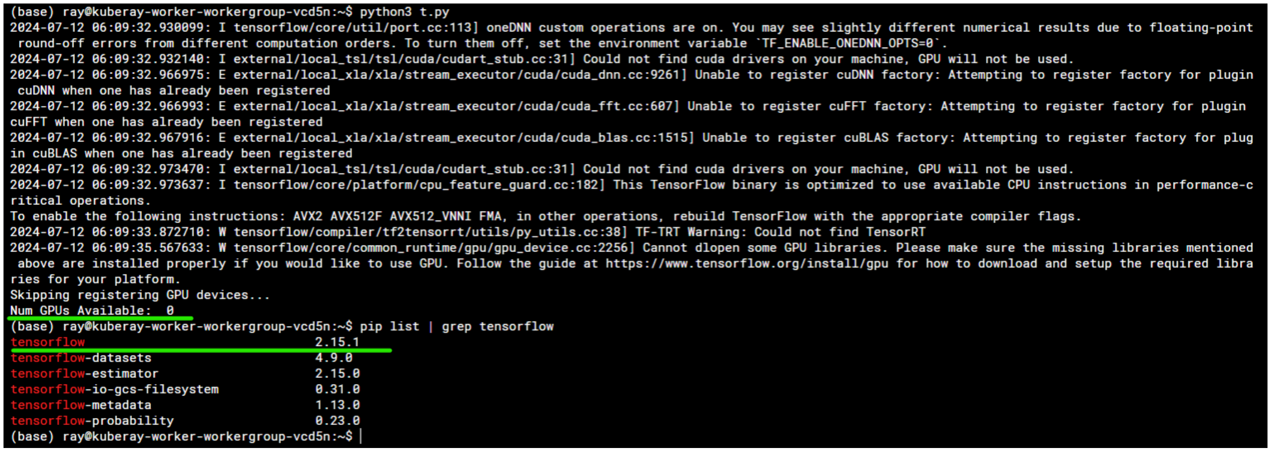
Ray example using GPUs:

For this example to work, ensure you have installed GPU version of TensorFlow.

```
@ray.remote(num_gpus=1)
def use_gpu():
    import tensorflow as tf

    # Create a TensorFlow session. TensorFlow will restrict itself to use
the
    # GPUs specified by the CUDA_VISIBLE_DEVICES environment variable.
    tf.Session()
```

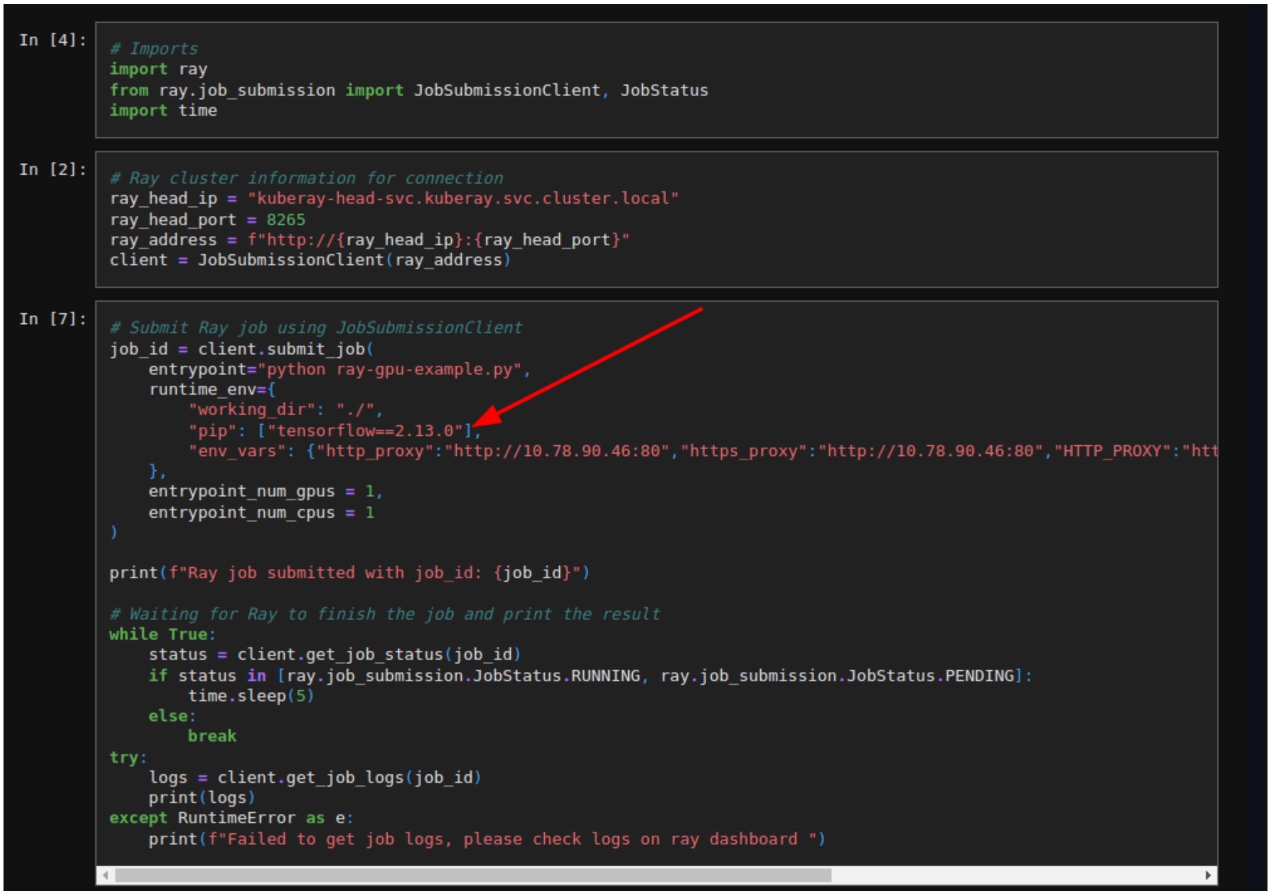
NOTE: As Ray does not have the GPU-specific API, you must properly configure Ray jobs to run on GPU. Without proper configuration, Ray jobs will run on CPUs.

When you submit the Ray GPU jobs using TensorFlow 2.15.1, TensorFlow 2.15.1 cannot find the CUDA driver and defaults to using the CPU for the job. This is related to the open-source issue (https://github.com/ray-project/ray/issues/46632).



```
(base) ray@kuberay-worker-workergroup-vcd5n:~$ python3 t.py
2024-07-12 06:09:32.930099: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point
 round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-07-12 06:09:32.932140: I external/local_tsl/tsl/cuda/cudart_stub.cc:31] Could not find cuda drivers on your machine, GPU will not be used.
2024-07-12 06:09:32.966975: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attempting to register factory for plugin
 cuDNN when one has already been registered
2024-07-12 06:09:32.966993: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
2024-07-12 06:09:32.967916: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Attempting to register factory for plug
in cuBLAS when one has already been registered
2024-07-12 06:09:32.973470: I external/local_tsl/tsl/cuda/cudart_stub.cc:31] Could not find cuda drivers on your machine, GPU will not be used.
2024-07-12 06:09:32.973637: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-c
ritical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-07-12 06:09:33.872710: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
2024-07-12 06:09:35.567633: W tensorflow/core/common_runtime/gpu/gpu_device.cc:2256] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned
 above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libra
ries for your platform.
Skipping registering GPU devices...
Num GPUs Available:  0
(base) ray@kuberay-worker-workergroup-vcd5n:~$ pip list | grep tensorflow
tensorflow                                2.15.1
tensorflow-datasets                       4.9.0
tensorflow-estimator                      2.15.0
tensorflow-io-gcs-filesystem              0.31.0
tensorflow-metadata                       1.13.0
tensorflow-probability                    0.23.0
(base) ray@kuberay-worker-workergroup-vcd5n:~$ |
```

To ensure that Ray jobs run on the GPU, manually update the TensorFlow version to 2.13.0 when submitting Ray GPU jobs.



```
In [4]:
# Imports
import ray
from ray.job_submission import JobSubmissionClient, JobStatus
import time
```

```
In [2]:
# Ray cluster information for connection
ray_head_ip = "kuberay-head-svc.kuberay.svc.cluster.local"
ray_head_port = 8265
ray_address = f"http://{ray_head_ip}:{ray_head_port}"
client = JobSubmissionClient(ray_address)
```

```
In [7]:
# Submit Ray job using JobSubmissionClient
job_id = client.submit_job(
    entrypoint="python ray-gpu-example.py",
    runtime_env={
        "working_dir": "./",
        "pip": ["tensorflow==2.13.0"],
        "env_vars": {"http_proxy":"http://10.78.90.46:80","https_proxy":"http://10.78.90.46:80","HTTP_PROXY":"htt
    },
    entrypoint_num_gpus = 1,
    entrypoint_num_cpus = 1
)

print(f"Ray job submitted with job_id: {job_id}")

# Waiting for Ray to finish the job and print the result
while True:
    status = client.get_job_status(job_id)
    if status in [ray.job_submission.JobStatus.RUNNING, ray.job_submission.JobStatus.PENDING]:
        time.sleep(5)
    else:
        break
try:
    logs = client.get_job_logs(job_id)
    print(logs)
except RuntimeError as e:
    print(f"Failed to get job logs, please check logs on ray dashboard ")
```

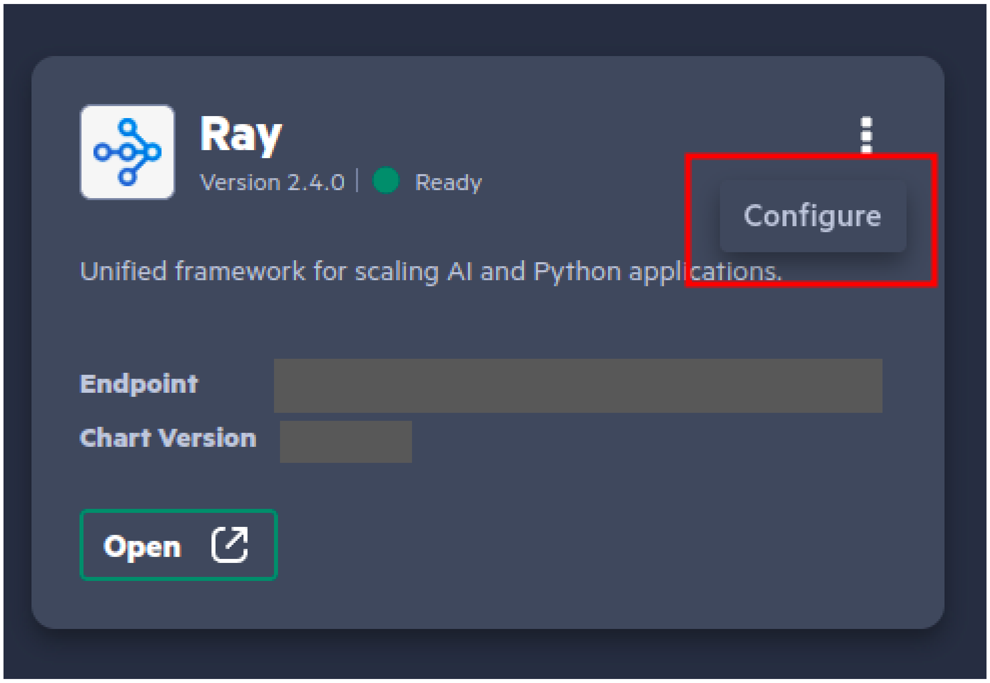
To learn more, see Using Ray with GPUs.

Disabling GPU Support for Ray

To disable GPU support for Ray after HPE Ezmeral Unified Analytics Software installation, follow these steps:

1. Click the **Tools & Frameworks** icon on the left navigation bar. Navigate to the **Ray** tile under the **Data Science** tab.

2. Click the **three dots** menu on the Ray tile and click **Configure**.



Unified framework for scaling AI and Python applications.

3. Set the value of `gpu.enabled` to `false`.

4. Click **Configure**.

Related tasks

Running Ray GPU Example on page 56

Describes how to run the Ray GPU example in HPE Ezmeral Unified Analytics Software.

More information

GPU Support on page 137

Provides information about support for NVIDIA GPU, MIG partitioning, preparing hosts for GPU-enabled environment, adding hosts and enabling GPU in HPE Ezmeral Unified Analytics Software.

Using whylogs with Ray

Note: This feature is presented as a developer preview. Developer previews are not tested for production environments, and should be used with caution.

Describes how to use whylogs with Ray.

Prerequisites

Sign in to HPE Ezmeral Unified Analytics Software as a member.

About this task

In HPE Ezmeral Unified Analytics Software, whylogs is integrated to work with Ray in a distributed environment. You can use whylogs with Ray for logging and analyzing the distributed data or monitoring the ML models in a distributed environment.

To use whylogs with Ray, refer to the Ray example in the GitHub. The basic steps are outlined as follows:

1. Create a notebook or import the notebook into HPE Ezmeral Unified Analytics Software. See Creating and Managing Notebook Servers on page 436.

2. Import the required libraries and modules from whylogs.

3. Use Ray for distributed data processing tasks.
4. Log the data with whylogs and store the results.
5. Once you finish running your notebook, navigate back to the HPE Ezmeral Unified Analytics Software home screen.
6. In the left navigation bar, go to **Data Engineering > Data Sources**.
7. Click **Browse**.
8. Go to the `/shared/<ray-whylogs>` folder which is a path set in your notebook to store the logs from whylogs. You can see that the data profiles and the drift summary report are stored in the shared volume in the `.html` and `.bin` formats.
9. To download a summary report, select **Download** from the Actions menu.

Results

You can analyze the summary report to detect drifts and monitor your data in a distributed environment.

Ray Best Practices

Lists the best practices for Ray.

Stabilize the Head Pod

In the Ray cluster, a head pod has a key role and therefore it should be stable. Hewlett Packard Enterprise recommends not scheduling any workload on the head pod.

The worker nodes handle all workloads in the default deployment of HPE Ezmeral Unified Analytics Software.

To make the head pod stable, when creating the Ray cluster, set

`{"num-cpus": "0"}` in `"rayStartParams"` of `"headGroupSpec"` such that the Ray scheduler skips the head node when scheduling workloads.

NOTE: This is set by default in HPE Ezmeral Unified Analytics Software.

Notebooks

Provides a brief overview of Notebooks in HPE Ezmeral Unified Analytics Software.

Notebooks are an interactive computational environment to develop and run your data science applications. You can use Notebooks to run code snippets, view the results, and then save the data in HPE Ezmeral Unified Analytics Software. Notebook files are saved with a `.ipynb` extension.

You can edit notebook files, change parameters, display the results, and document the methodology, results, summary, and findings within the same file.

You can run your commands, visualize data, and get outputs and results using the following Kubeflow Notebook interfaces:

- JupyterLab
- Visual Studio Code

Features and Functionality

Kubeflow in HPE Ezmeral Unified Analytics Software supports the following features and functionality:

- All the actions are done as your logged-in user and not as the Jovyan user.
- The notebooks are integrated with the SDK (Feast, MLlflow, Kubeflow Pipelines, Katib, Ray, EzPresto).
- Special data science notebook that includes common machine learning and data science libraries.
- Ability to install new packages to the notebook at runtime or create a custom notebook image.

To experience notebooks in HPE Ezmeral Unified Analytics Software, refer to the following tutorials:

- Financial Time Series Workflow on page 21
- MNIST Digits Recognition Workflow on page 31

Notebook Images Overview

Describes notebook images available in HPE Ezmeral Unified Analytics Software and their uses.

Notebook images contain all the necessary software dependencies and configurations needed to run machine learning workflows. By using notebook images, you can collaborate, share, and deploy models with minimal compatibility issues.

Image Format

The images follow the following format:

```
<base-repository>/<image-name>:<image-tag>
```

For example:

```
gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:ezaf-fy23-q4-sp4-r9
```

Here,

- base-repository: `gcr.io/mapr-252711/kubeflow/notebooks`
- image-name: `jupyter-scipy`
- image-tag: `ezaf-fy23-q4-sp4-r9`

Supported Notebook Images

The following table describes the notebook images available in HPE Ezmeral Unified Analytics Software and their uses.

Notebook Images	Descriptions	Uses
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:<image-tag>`	This image is packaged with data science packages, including Pandas for data manipulation, Matplotlib and Bokeh for advanced plotting, and statistical tools such as SciPy and Statsmodels.	Use this image to perform data analysis, manipulation, and visualization that doesn't require machine learning libraries such as TensorFlow or PyTorch.

Notebook Images	Descriptions	Uses
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-full:<image-tag>`	This image is packaged with data science packages and is integrated with PyTorch machine learning libraries for CPU-based tasks. This image does not have GPU acceleration capability.	Use this image to perform data analysis, manipulation, and visualization for CPU-based machine learning tasks using PyTorch library.
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-cuda-full:<image-tag>`	This image is packaged with data science packages and is integrated with PyTorch machine learning libraries for GPU-based tasks.	Use this image to perform data analysis, manipulation, and visualization for GPU-based machine learning tasks using PyTorch library for faster model training and data processing.
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:<image-tag>`	This image is packaged with data science packages and is integrated with TensorFlow machine learning libraries for CPU-based tasks. This image does not have GPU acceleration capability.	Use this image to perform data analysis, manipulation, and visualization for CPU-based machine learning tasks using TensorFlow library.
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:<image-tag>`	This image is packaged with data science packages and is integrated with TensorFlow machine learning libraries for GPU-based tasks.	Use this image to perform data analysis, manipulation, and visualization for GPU-based machine learning tasks using TensorFlow library for faster model training and data processing.
`gcr.io/mapr-252711/kubeflow/notebooks/jupyter-data-science:<image-tag>`	This image is integrated with Tensorflow and PyTorch packages, including various other tools for data analysis, machine learning, and visualization.	Use this image that is integrated with data science libraries to perform data science tasks requiring deep learning capabilities of TensorFlow and PyTorch.
`gcr.io/mapr-252711/kubeflow/notebooks/codeserver:<image-tag>`	This image enables you to run Visual Studio Code in the browser where you can edit and develop code in a remote server setup. This image features a VS Code environment, providing a code-server that runs Visual Studio Code in the browser, allowing for rich code editing and development experience in a remote server setup. In HPE Ezmeral Unified Analytics Software, the codeserver image includes VS Code and Python installation, and VS Code Python extension.	Use this image to run Visual Studio Code in the browser where you can edit and develop code in a remote server setup.

Image and Package Support

For a list of supported notebook images and included packages in HPE Ezmeral Unified Analytics Software, see Notebook Images on page 202.

Creating and Managing Notebook Servers

Describes how to create and manage notebook servers in HPE Ezmeral Unified Analytics Software.

Prerequisites

Sign in to HPE Ezmeral Unified Analytics Software.

TIP:

- When you create a new notebook, the notebook name must consist of lowercase alphanumeric characters, with or without dashes (-). The name cannot start with a number; the name must start with a letter (a-z). For example, you can name a notebook `my-notebook-1`, but you cannot name a notebook `1-my-notebook`.
- If your username starts with a number, such as `3user`, your default user notebook name also starts with a number (`3user-notebook`), which is not supported. For additional information, see Troubleshooting Notebooks.

About this task

Create and manage notebook servers in HPE Ezmeral Unified Analytics Software.

Procedure

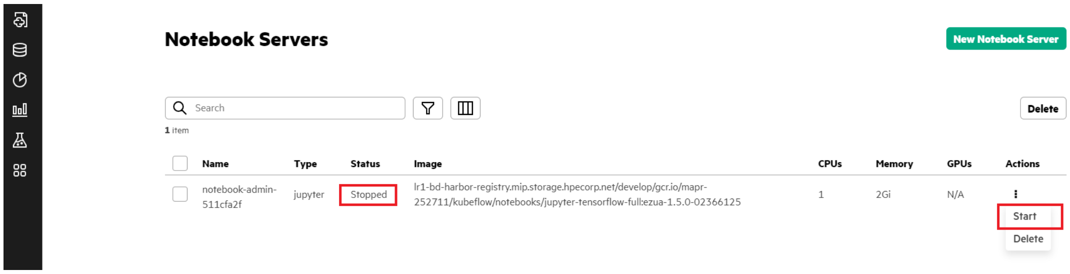
1. Click **Notebooks icon** on the left navigation bar of HPE Ezmeral Unified Analytics Software screen.

   You are now in the **Notebook Servers** screen. You can see a default `<notebook-user-namespace-name>` Jupyter notebook has been created with tensorflow image by using Kubeflow Notebooks. By default, the Unified Analytics creates the default notebook in a Stopped status.

   A notebook can be created with two types of PVCs: an existing one and a new one. Each notebook usually has a special notebook PVC (workspace volume) that makes the notebook's home folder persistent. The notebook can also contain user data PVCs. The existing PVCs must be created before creating the notebook. The new PVCs are created when you create the notebook.

   - To connect to a default Jupyter notebook (for example, `<notebook-user-namespace-name>`), follow these steps:

     a. Click the **Actions** menu.



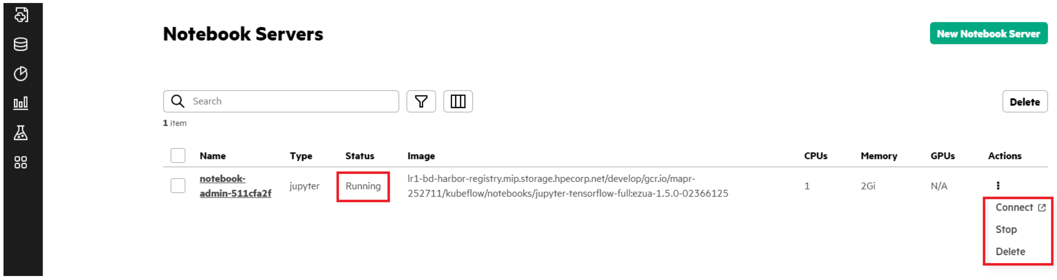
Notebook Servers

1 item

Name	Type	Status	Image	CPUs	Memory	GPUs	Actions
notebook-admin-511cfa2f	jupyter	Stopped	lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezua-1.5.0-02366125	1	2Gi	N/A	⋮

     b. Select **Start** and wait for the notebook to be in the `Running` status.

     c. Click **<notebook-user-namespace-name>** or select **Connect**.



Notebook Servers

1 item

Name	Type	Status	Image	CPUs	Memory	GPUs	Actions
notebook-admin-511cfa2f	jupyter	Running	lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezua-1.5.0-02366125	1	2Gi	N/A	⋮

- To create notebook servers, click **New Notebook Server**. You can choose JupyterLab or Visual Studio Code as your notebook server within Kubeflow Notebooks. To learn more, see Kubeflow Notebooks.

When any notebook is created, the following volumes are added automatically:

- `<username>`: This directory is mounted to user-pvc volume. Only the current user has access to the data stored in the `<username>` folder.
- `shared`: This directory can be accessed by all authorized users. The `shared` directory contains all the notebook examples.
- `logs`: This directory contains the log files. To learn more, see Logging on page 287.

You can run your commands, visualize data, and get outputs and results by using notebook servers.

2. To view actions that you can perform on **Notebook Servers** screen, click the **menu icon** in the **Actions** column.

Start:	To start notebook servers, select **Start**.
Stop	Top stop notebook servers from running, select **Stop**.
Connect:	To connect to notebook servers, select **Connect**.
Delete:	To delete the notebook server, select **Delete**.

NOTE: When you stop or delete the notebook, the PVCs remain and are not deleted along with the notebook, and the processes that were running within the notebook are forcefully stopped. In this case, you must manually restart the notebook processes as required.

3. Delete multiple notebook servers at once:
   a) To select multiple notebook servers, click the check box besides **Name** in the table.
   b) Click **Delete** on the top right pane of the table.
4. To display notebook servers according to the status, click the **Filter** icon.
5. To select the columns to display on your applications table, click the **Columns** icon.

Related tasks

Creating GPU-Enabled Notebook Servers on page 438
Describes how to create and deploy the GPU-enabled notebook servers.

Creating GPU-Enabled Notebook Servers

Describes how to create and deploy the GPU-enabled notebook servers.

Prerequisites

Sign in to HPE Ezmeral Unified Analytics Software.

About this task

Create GPU-enabled notebook servers in HPE Ezmeral Unified Analytics Software.

Procedure

1. Click **Notebooks icon** on the left navigation bar of HPE Ezmeral Unified Analytics Software screen.

2. Click **New Notebook Server**. You will be navigated to the Kubeflow Notebooks UI. You can choose JupyterLab as your notebook server within Kubeflow Notebooks.

3. Configure the notebook server with the following options:
   - Select one of the following docker images:
     - (Tensorflow CUDA image) `gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:<image-tag>`
     - (PyTorch CUDA image) `gcr.io/mapr-252711/kubeflow/notebooks/jupyter-pytorch-cuda-full:<image-tag>`
   - Set **Requested memory in Gi** to at least two to three Gi.
   - Set **GPUs** as follows:
     - **Number of GPUs**: 1

       NOTE: With MIG configuration, only one GPU can be assigned per application. To learn more on what happens when you assign more than one GPU to the notebook server, see GPU on page 165. For details regarding GPU, see GPU Support on page 137.
     - **GPU Vendor**: Nvidia

4. Click **Launch**.

Results

The new GPU-enabled notebook server is created.

Related tasks

Creating and Managing Notebook Servers on page 436
Describes how to create and manage notebook servers in HPE Ezmeral Unified Analytics Software.

More information

GPU Support on page 137
Provides information about support for NVIDIA GPU, MIG partitioning, preparing hosts for GPU-enabled environment, adding hosts and enabling GPU in HPE Ezmeral Unified Analytics Software.

Building Custom Kubeflow Jupyter Notebook Image

Describes how to build the custom Kubeflow Jupyter notebook image.

About this task

Build a custom image with one of the default notebooks available in the Kubeflow dashboard as a base image. The notebook will be created using the custom image.

You can build the custom image for both air-gapped and non-air-gapped environments for all three types of packages – OS level packages, conda packages and pip packages.

To build the custom Kubeflow Jupyter notebook image, perform:

Procedure

1. Create `requirements.txt` file.

   For example: The content of the file can be following:

   ```
   ###requirements.txt
   # pandas packages
   pandas=1.5.0
   numpy=1.24.2
   # Some other packages
   ###
   ```

2. Create `Dockerfile`.

   ```
   ARG BASE_IMG=gcr.io/mapr-252711/kubeflow/notebooks/jupyter-scipy:
   <image-tag>
   FROM $BASE_IMG
   COPY requirements.txt /tmp/requirements.txt
   RUN python3 -m pip install -r /tmp/
   requirements.txt --quiet --no-cache-dir \
    && rm -f /tmp/requirements.txt
   ```

3. Build the image with `docker build` command. Replace the `<image>` with actual image name, `<tag>` with actual tag name, and `<base_img>` with actual base image.

   ```
   docker build -t  <image>:<example>  .
   ```

   (OR) If the default base image is not suitable,

   a. Choose one of the default images as the base image. See Notebook Images on page 202.

   b. Run `docker build -t <image>:<example> --build-arg BASE_IMG=<base_img> .`

4. Push the image to the registry.

   ```
   docker push <image>:<example>
   ```

5. Use the custom image option when creating the notebook server in the Kubeflow UI.

Installing Custom Packages in Kubeflow Notebooks at Runtime

Describes how to install custom packages in existing Kubeflow notebooks that persist between restarts.

You can only install custom packages in a Kubeflow notebook in connected HPE Ezmeral Unified Analytics Software environments for two types of packages – conda packages and pip packages.

You cannot install custom packages in air-gapped environments. Packages installed to the base environment do not persist; the packages are removed after the notebook restarts.

By default, the base conda environment is activated for all notebook users. All notebook users can perform the following tasks:

- Install packages to the base environment
- Create and install your own conda environment
- Use the conda environment of another notebook user, if permitted by the environment owner

You can install packages that persist (save between restarts). You can also install packages that do not persist (do not save between restarts).

- If packages *do not* have to persist between restarts, install the packages to the base conda environment. This applies to both single-user and multi-user modes.
- If packages *must* persist between restarts, create an individual conda environment. This applies to both single-user and multi-user modes.

The following sections describe how to create an individual conda environment where you can install packages that persist between restarts in single and multi-user modes:

NOTE: Run commands in the notebook terminal.

Single-User Mode

Complete the following steps in the notebook if you want to install custom packages that persist between restarts in single-user mode:

1. Create an individual conda environment:

```
conda create --prefix ~/.conda/envs/kf-users-env --clone base
```

2. Activate the conda environment:

```
conda activate kf-users-env
```

Multi-User Mode

Any user with access to the notebook, typically the owner, can create the conda environment. The conda environment is shared with other users (between contributors). All users get equivalent permissions. Users can use the existing packages, as well as install and remove the packages.

Complete the following steps in the notebook if you want to install custom packages that persist between restarts in multi-user mode:

1. Create the conda environment:

```
umask 0000 && conda create --prefix ~/.conda/envs/kf-users-env --clone
base
```

2. Activate the conda environment:

```
conda activate kf-users-env
```

3. Add users (contributors) to the conda environment:

```
conda config --append envs_dirs  /home/<notebook_owner_username>/.conda/
envs
```

4. Activate the conda environment for users:

```
conda activate kf-users-env
```

5. Install the conda package:

```
conda install package-name=<version>
```

6. Install the PIP package:

```
pip install package-name==<version>
```

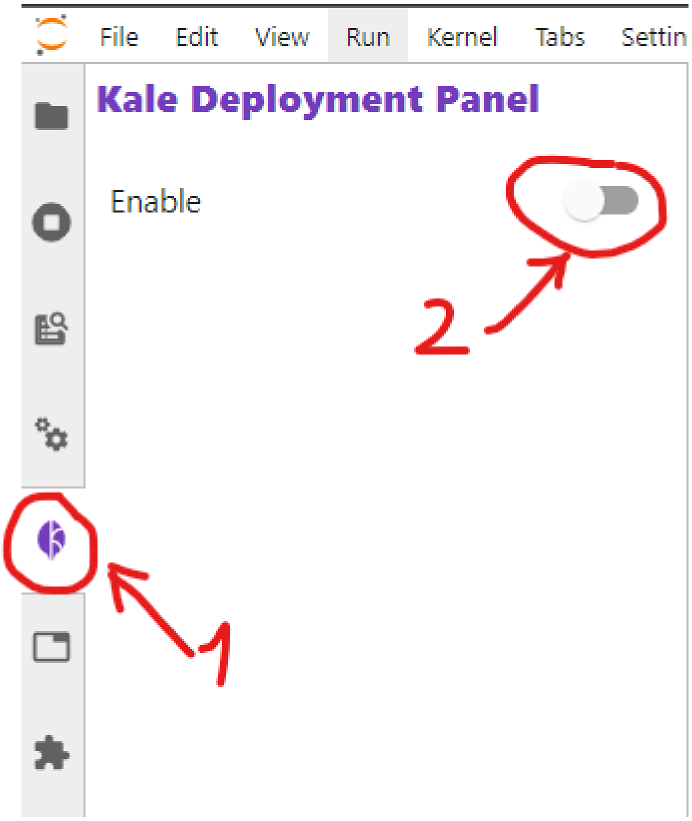
Enabling Kale Extension in Kubeflow Notebook

Describes how to enable and use the Kale extension, and specify GPU resources using Kale extension in a Kubeflow notebook.

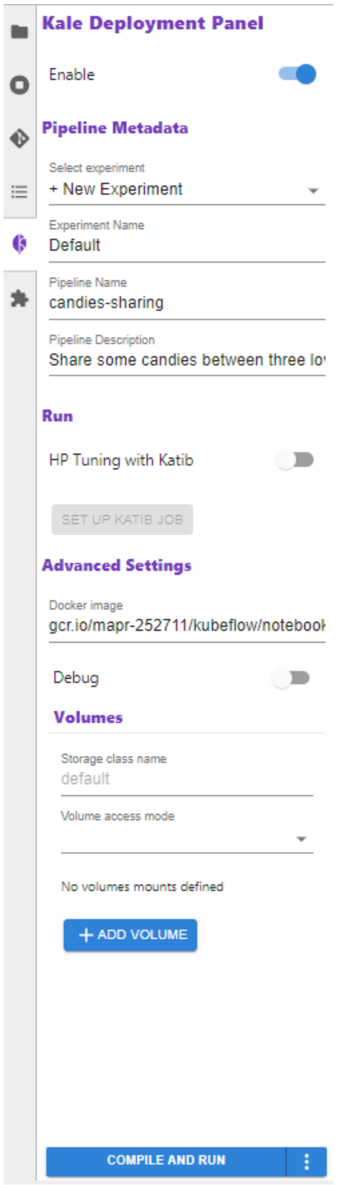
Enabling Kale Extension

To enable the Kale extension in Kubeflow notebook, follow these steps:

1. Sign in to HPE Ezmeral Unified Analytics Software.
2. Connect to the notebook server. See Creating and Managing Notebook Servers on page 436.
3. Once the notebook is launced, you can enable the Kale extension.
   a. Click the **Kale** icon on the left navigaton bar.
   b. Toggle the **Enable** button.



4. Once you enable the Kale extension, you can view the Kale extension layout as next:



Kale Deployment Panel

Enable

Pipeline Metadata

Select experiment: + New Experiment

Experiment Name: Default

Pipeline Name: candies-sharing

Pipeline Description: Share some candies between three lo

Run

HP Tuning with Katib

Advanced Settings

Docker image: gcr.io/mapr-252711/kubeflow/noteboo

Debug

Volumes

Storage class name: default

Volume access mode

No volumes mounts defined

Pipeline Metadata

Define the name of the experiment and pipeline, description.

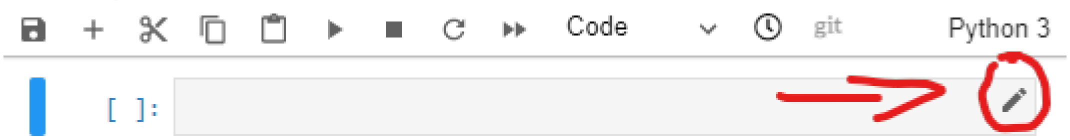
Run

Enable the Katib feature for this pipeline and setup appropriate hyper-parameters.

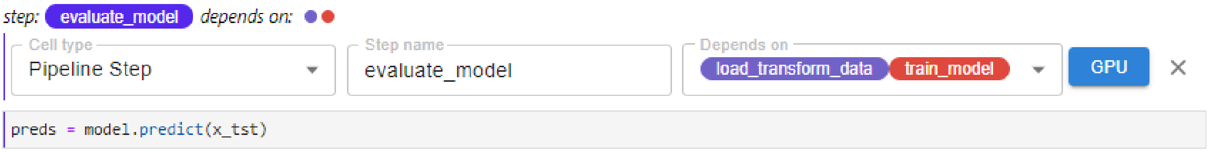
Advanced Settings

Click **Advanced Settings** to open this section. Here you can set a Docker image, which is used for all steps of the current pipeline. Currently, the Rok snapshot feature is not supported. However, you can manually create or use the existing volume for this pipeline.

5. Click the pencil icon to edit the cell.



Here, you can edit the information about the cell for Kale.



step: evaluate_model depends on: ●●

Cell type: Pipeline Step

Step name: evaluate_model

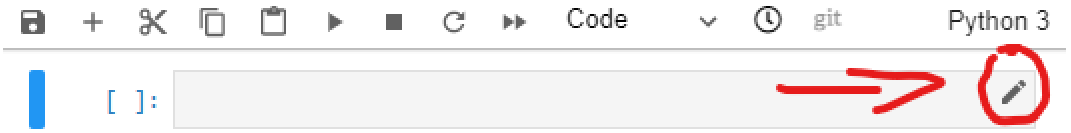
Depends on: load_transform_data, train_model

```
preds = model.predict(x_tst)
```

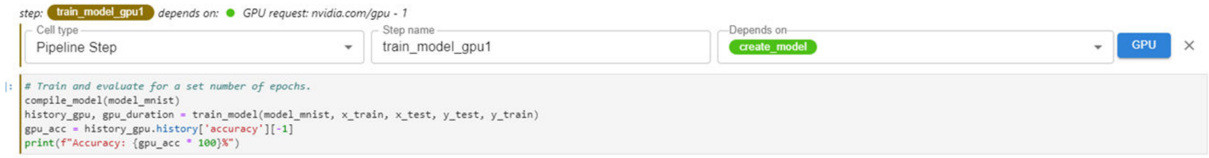
Specifying GPU Resources in the Kale Extension

To specify the GPU resources in the Kale extension, follow these steps:

1. Create GPU-enabled notebook server. See Creating GPU-Enabled Notebook Servers on page 438.
2. Enable Kale extension. See Enabling Kale Extension on page 442.
3. Click the pencil icon to edit the cell.



4. Click **GPU**.



step: train_model_gpu1 depends on: ● GPU request: nvidia.com/gpu - 1

Cell type: Pipeline Step

Step name: train_model_gpu1

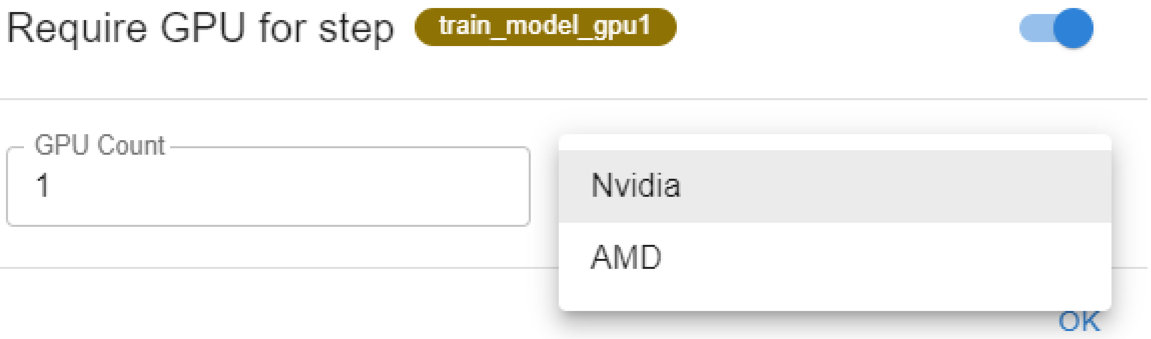
Depends on: create_model

```
# Train and evaluate for a set number of epochs.
compile_model(model_mnist)
history_gpu, gpu_duration = train_model(model_mnist, x_train, x_test, y_test, y_train)
gpu_acc = history_gpu.history['accuracy'][-1]
print(f"Accuracy: {gpu_acc * 100}%")
```

5. Specify the GPU resources as:
   a. **GPU Count**: 1

      NOTE: With MIG configuration, only one GPU can be assigned per application. To learn more on what happens when you assign more than one GPU to the Kale extension, see GPU on page 165. For details regarding GPU, see GPU Support on page 137.

   b. **GPU Vendor**: Nvidia



Require GPU for step train_model_gpu1

GPU Count: 1

More information

https://github.com/kubeflow-kale/kale

Candy Sharing Tutorial (Kale) on page 18

Describes how Kale converts Notebook to pipeline by applying notebook annotations.

GPU Support on page 137
Provides information about support for NVIDIA GPU, MIG partitioning, preparing hosts for GPU-enabled environment, adding hosts and enabling GPU in HPE Ezmeral Unified Analytics Software.

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 Unified Analytics Software 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 extends the capabilities of a notebook. Magic commands start with the `%` character.

HPE Ezmeral Unified Analytics Software supports built-in magic functions and the custom magics that are described in this topic. HPE Ezmeral Unified Analytics Software supports line magics and cell magics.

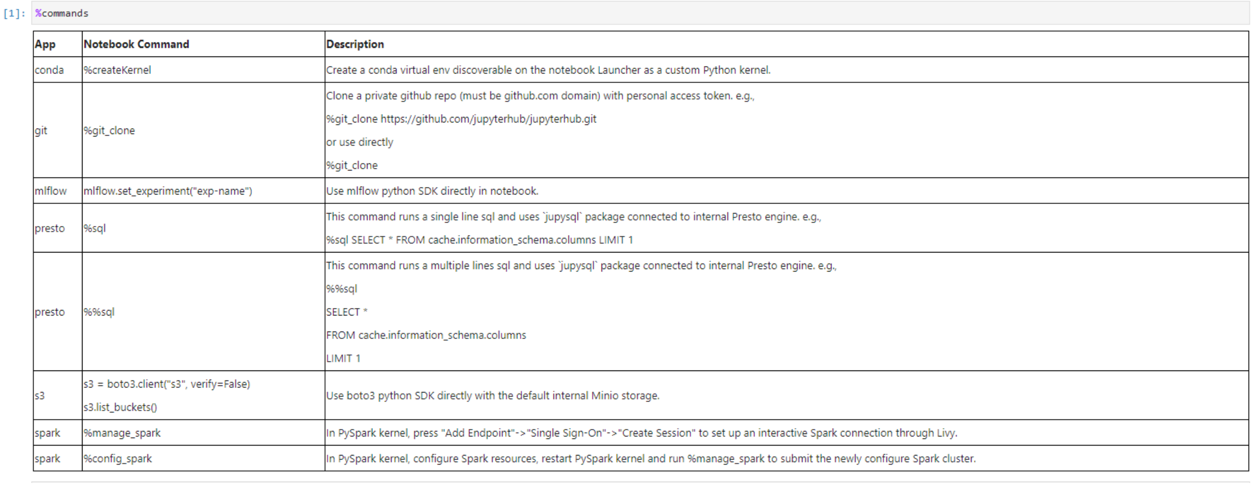
Line magic commands do not require a cell body and start with a single `%` character.

Cell magic commands start with `%%` and require additional lines of input (a cell body).

To use these magic functions, you must create a notebook. See Creating and Managing Notebook Servers on page 436.

%commands

The `%commands` command lists the magic commands and SDKs that are customized by Hewlett Packard Enterprise and are available in this notebook.



```
[1]: %commands
```

App	Notebook Command	Description
conda	%createKernel	Create a conda virtual env discoverable on the notebook Launcher as a custom Python kernel.
git	%git_clone	Clone a private github repo (must be github.com domain) with personal access token. e.g., %git_clone https://github.com/jupyterhub/jupyterhub.git or use directly %git_clone
mlflow	mlflow.set_experiment("exp-name")	Use mlflow python SDK directly in notebook.
presto	%sql	This command runs a single line sql and uses `jupysql` package connected to internal Presto engine. e.g., %sql SELECT * FROM cache.information_schema.columns LIMIT 1
presto	%%sql	This command runs a multiple lines sql and uses `jupysql` package connected to internal Presto engine. e.g., %%sql SELECT * FROM cache.information_schema.columns LIMIT 1

s3	s3 = boto3.client("s3", verify=False)
	s3.list_buckets()
		Use boto3 python SDK directly with the default internal Minio storage.

spark	%manage_spark	In PySpark kernel, press "Add Endpoint"->"Single Sign-On"->"Create Session" to set up an interactive Spark connection through Livy.
spark	%config_spark	In PySpark kernel, configure Spark resources, restart PySpark kernel and run %manage_spark to submit the newly configure Spark cluster.

%createKernel

The `%createKernel` command creates a custom Python kernel in the notebook.

The custom Python kernel can be selected as the kernel for a notebook session to work within a specific virtual environment with its own set of dependencies and configurations. By using the custom Python kernel, you can isolate your Python packages and dependencies from other projects or applications such that each project has its own environment and is not affected by changes made to other environments.

To create a custom Python kernel from conda package installation, perform:

1. Create a notebook with at least 4 Gi memory. See Creating and Managing Notebook Servers on page 436.

2. (Optional) If the cluster is behind proxy, set the following proxy environment variables.

   ```
   %env https_proxy=<your-https-proxy>
   %env http_proxy=<your-http-proxy>
   %env no_proxy=<your-no-proxy>
   ```

   You can retrieve the proxy settings from the `ezua-cluster-config` configmap in the `ezua-system` namespace.

   NOTE: If you are using the AWS or Azure environment, do not set the proxy environment variables.

3. Enter the `%createKernel` command in a notebook.

   You can also directly enter packages as arguments with `%createKernel` magic function.

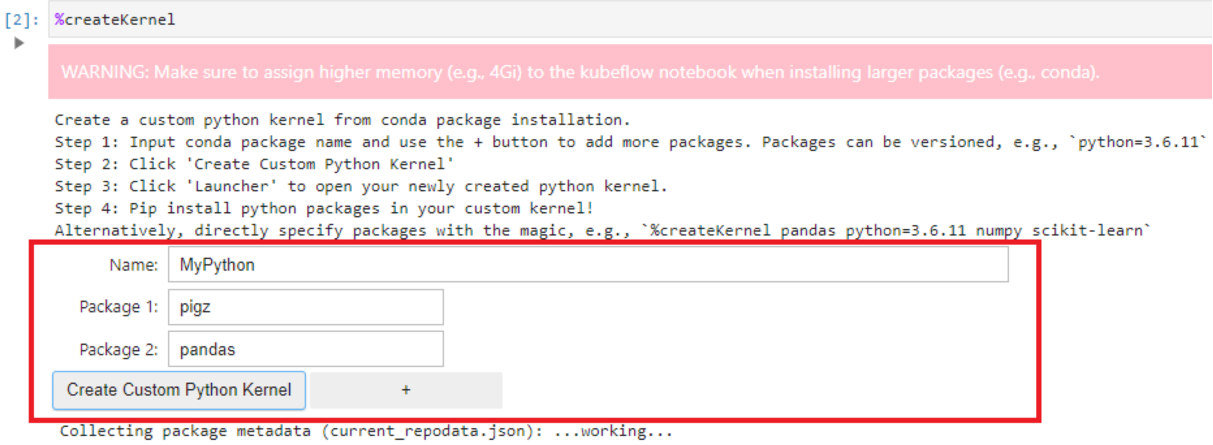
   For example:

   ```
   %createKernel pigz pandas
   ```

4. Enter the name for your Python kernel in the **Name** box. In this example, we use **MyPython** as the custom Python kernel name.

5. Enter the conda package name in the **Package 1** box. To enter additional packages, click the **+** button.

6. Click the **Create Custom Python Kernel** button.



```
[2]: %createKernel
```

WARNING: Make sure to assign higher memory (e.g., 4Gi) to the kubeflow notebook when installing larger packages (e.g., conda).

```
Create a custom python kernel from conda package installation.
Step 1: Input conda package name and use the + button to add more packages. Packages can be versioned, e.g., `python=3.6.11`
Step 2: Click 'Create Custom Python Kernel'
Step 3: Click 'Launcher' to open your newly created python kernel.
Step 4: Pip install python packages in your custom kernel!
Alternatively, directly specify packages with the magic, e.g., `%createKernel pandas python=3.6.11 numpy scikit-learn`
```

Name: MyPython

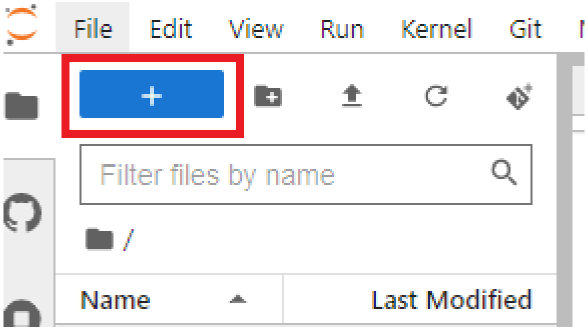
Package 1: pigz

Package 2: pandas

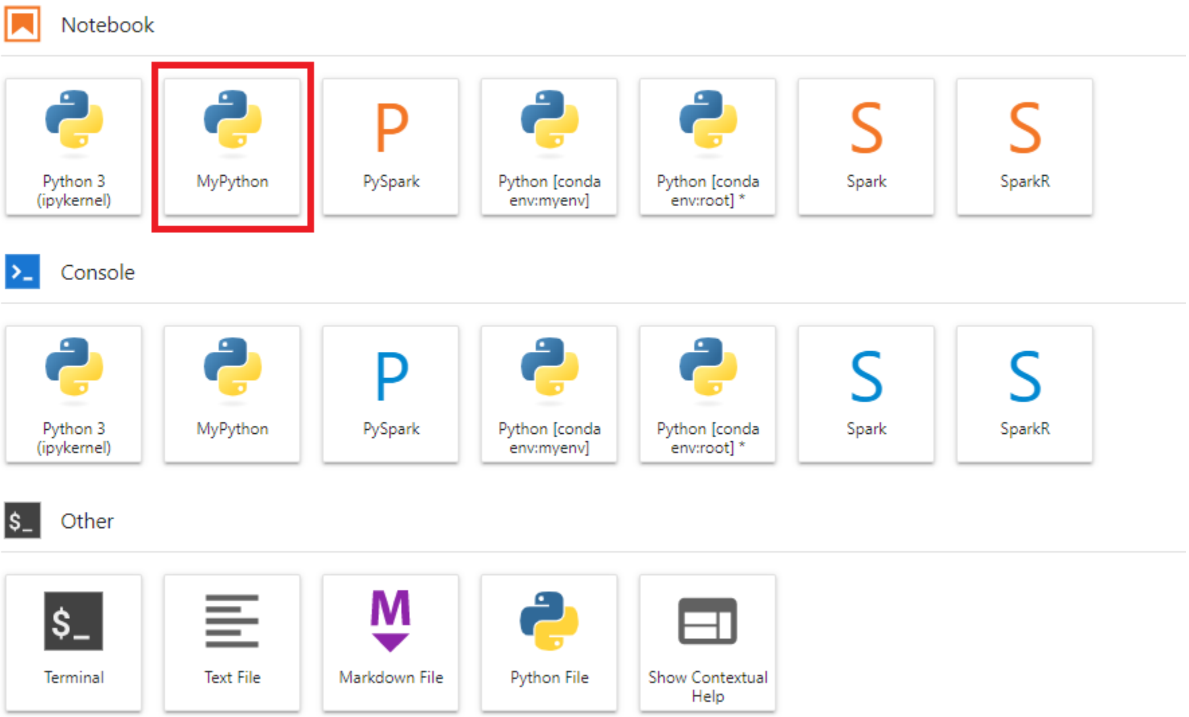
Create Custom Python Kernel +

```
Collecting package metadata (current_repodata.json): ...working...
```

7. Click the **New Launcher** button.



8. You can now see your custom Python kernel -**MyPython** kernel among the available kernels.

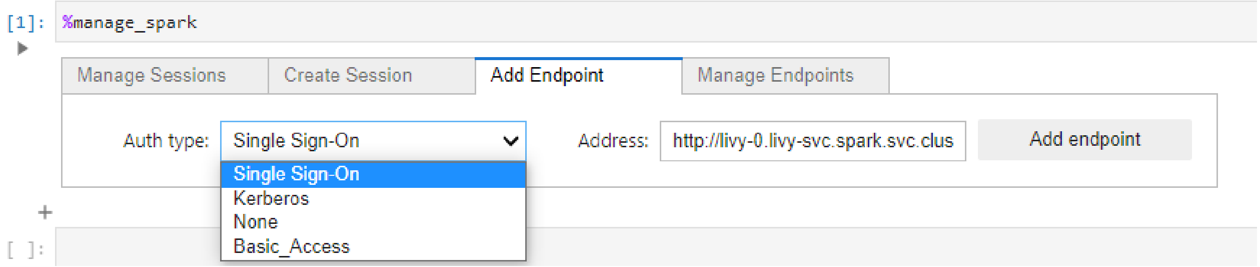


%manage_spark

The `%manage_spark` command enables you to connect to the Livy server and start a new Spark session. You must use Spark-related kernels such as PySpark, Spark, or SparkR to use `sparkmagic`. When you run the `%manage_spark` command in a notebook cell, a new user interface (UI) widget is displayed, which allows you to configure and manage a Spark cluster. You can use different tabs in this UI widget to manage sessions, create sessions, add endpoints, and manage endpoints.

Add Endpoint: To add endpoints, set the following boxes and then click **Add endpoint**.

- Auth type: The default authentication for the internal Livy endpoint is Single Sign-On. To connect to other Livy clusters, select the authentication type of your choice.
- Address: Enter endpoint address.



```
[1]: %manage_spark
```

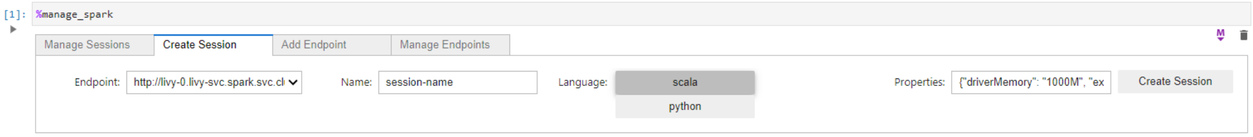
Auth type: Single Sign-On

Address: http://livy-0.livy-svc.spark.svc.clus

Create Session: To create sessions, set the following boxes and then click **Create Session**.

- Endpoint: Select endpoint for your session.
- Name: Enter session name.
- Language: Choose either Scala or Python as a runtime.

- Properties: Edit the Spark configurations.



```
[1]: %manage_spark
```

Endpoint: http://livy-0.livy-svc.spark.svc.cl

Name: session-name

Language: scala

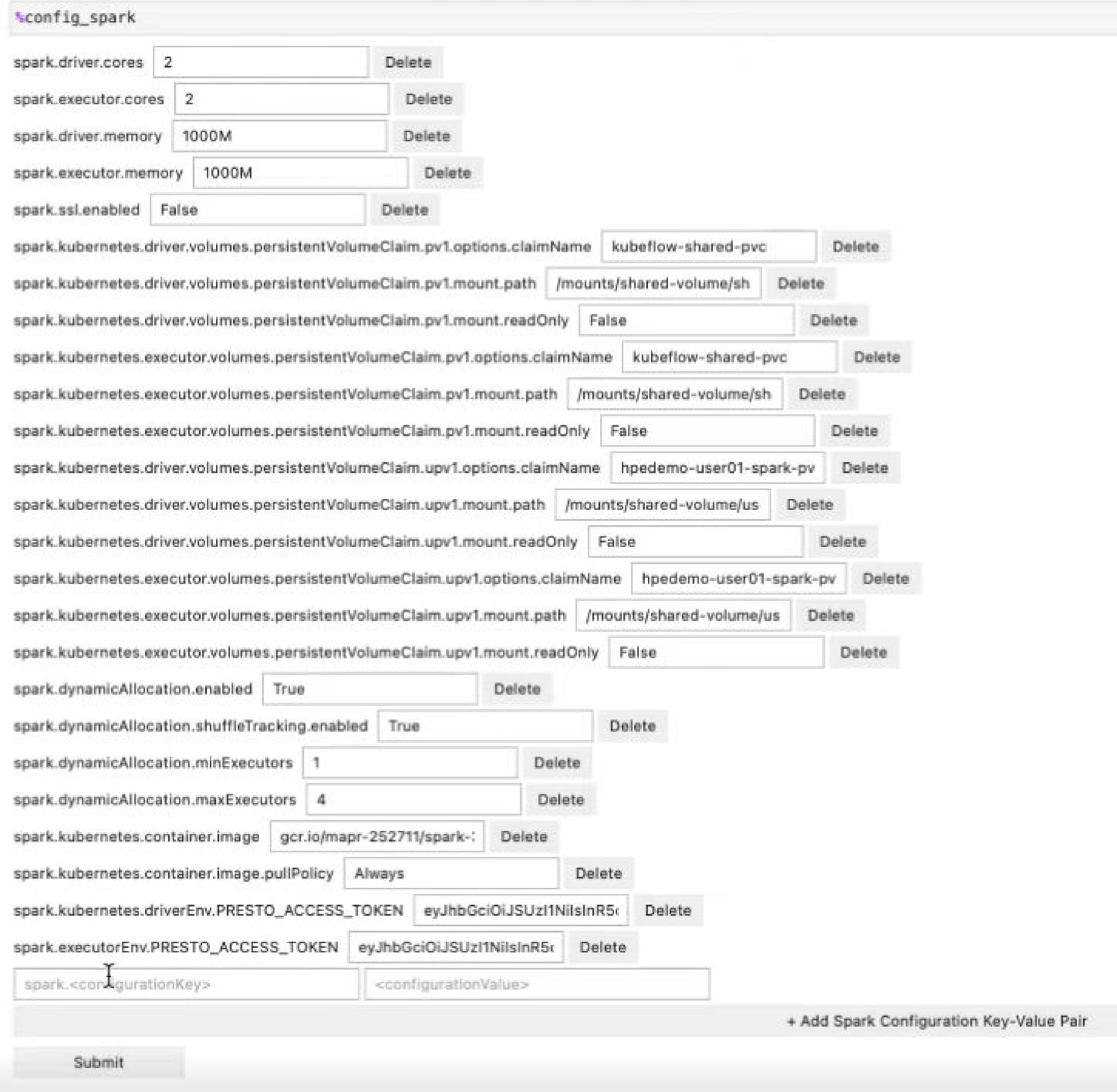
Properties: {"driverMemory": "1000M", "ex

%config_spark

You can use the `%config_spark` magic command to customize the Spark jobs submitted from the PySpark kernel. You can add or delete the Spark configurations when submitting a Livy session.

To customize the Spark configurations, follow these steps:

1. Run `%config_spark`.
2. Click the **+Add Spark Configuration Key-Value Pair** button.
3. Enter the key and value for Spark configurations in their respective boxes.
4. After you have finished adding the key-value pairs, click **Submit**.
5. Restart the PySpark kernel and run `%manage_spark` to see the changes applied in the **Properties** section of the **Create Session** tab.



```
%config_spark
```

spark.driver.cores	2	Delete
spark.executor.cores	2	Delete
spark.driver.memory	1000M	Delete
spark.executor.memory	1000M	Delete
spark.ssl.enabled	False	Delete
spark.kubernetes.driver.volumes.persistentVolumeClaim.pv1.options.claimName	kubeflow-shared-pvc	Delete
spark.kubernetes.driver.volumes.persistentVolumeClaim.pv1.mount.path	/mounts/shared-volume/sh	Delete
spark.kubernetes.driver.volumes.persistentVolumeClaim.pv1.mount.readOnly	False	Delete
spark.kubernetes.executor.volumes.persistentVolumeClaim.pv1.options.claimName	kubeflow-shared-pvc	Delete
spark.kubernetes.executor.volumes.persistentVolumeClaim.pv1.mount.path	/mounts/shared-volume/sh	Delete
spark.kubernetes.executor.volumes.persistentVolumeClaim.pv1.mount.readOnly	False	Delete
spark.kubernetes.driver.volumes.persistentVolumeClaim.upv1.options.claimName	hpedemo-user01-spark-pv	Delete
spark.kubernetes.driver.volumes.persistentVolumeClaim.upv1.mount.path	/mounts/shared-volume/us	Delete
spark.kubernetes.driver.volumes.persistentVolumeClaim.upv1.mount.readOnly	False	Delete
spark.kubernetes.executor.volumes.persistentVolumeClaim.upv1.options.claimName	hpedemo-user01-spark-pv	Delete
spark.kubernetes.executor.volumes.persistentVolumeClaim.upv1.mount.path	/mounts/shared-volume/us	Delete
spark.kubernetes.executor.volumes.persistentVolumeClaim.upv1.mount.readOnly	False	Delete
spark.dynamicAllocation.enabled	True	Delete
spark.dynamicAllocation.shuffleTracking.enabled	True	Delete
spark.dynamicAllocation.minExecutors	1	Delete
spark.dynamicAllocation.maxExecutors	4	Delete
spark.kubernetes.container.image	gcr.io/mapr-252711/spark-:	Delete
spark.kubernetes.container.image.pullPolicy	Always	Delete
spark.kubernetes.driverEnv.PRESTO_ACCESS_TOKEN	eyJhbGciOiJSUzI1NiIsInR5c	Delete
spark.executorEnv.PRESTO_ACCESS_TOKEN	eyJhbGciOiJSUzI1NiIsInR5c	Delete
spark.<configurationKey>	<configurationValue>	

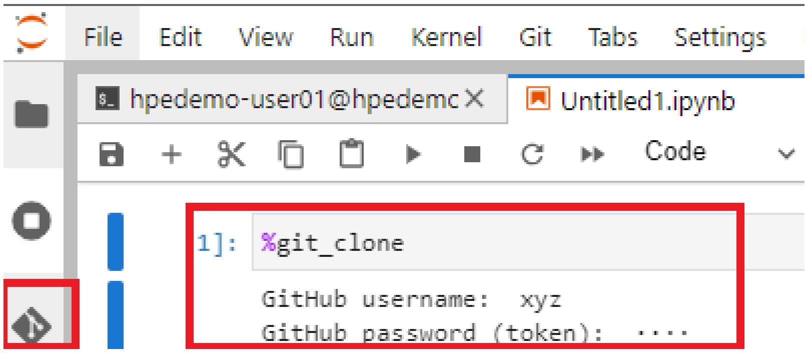
You can also edit the values for other Spark configurations or delete any configurations using this magic command.

For example: To learn more about customizing Spark configurations when enabling the GPU support for Livy sessions, see Enabling GPU Support for Livy Sessions Created Using Notebooks on page 403.

%git_clone

The `%git_clone` magic enables you to clone your private GitHub repository from the notebook.

To clone the repository, enter your GitHub username and GitHub password.



```
1]: %git_clone

    GitHub username:  xyz
    GitHub password (token):  ····
```

After you have finished cloning the repository, you can use the **Git** extension in the notebook for version control.

%sql and %%sql

You can use the `%sql` magic command in Jupyter Notebook to interactively work with SQL databases. To learn more about how to connect to databases, see connecting to a database.

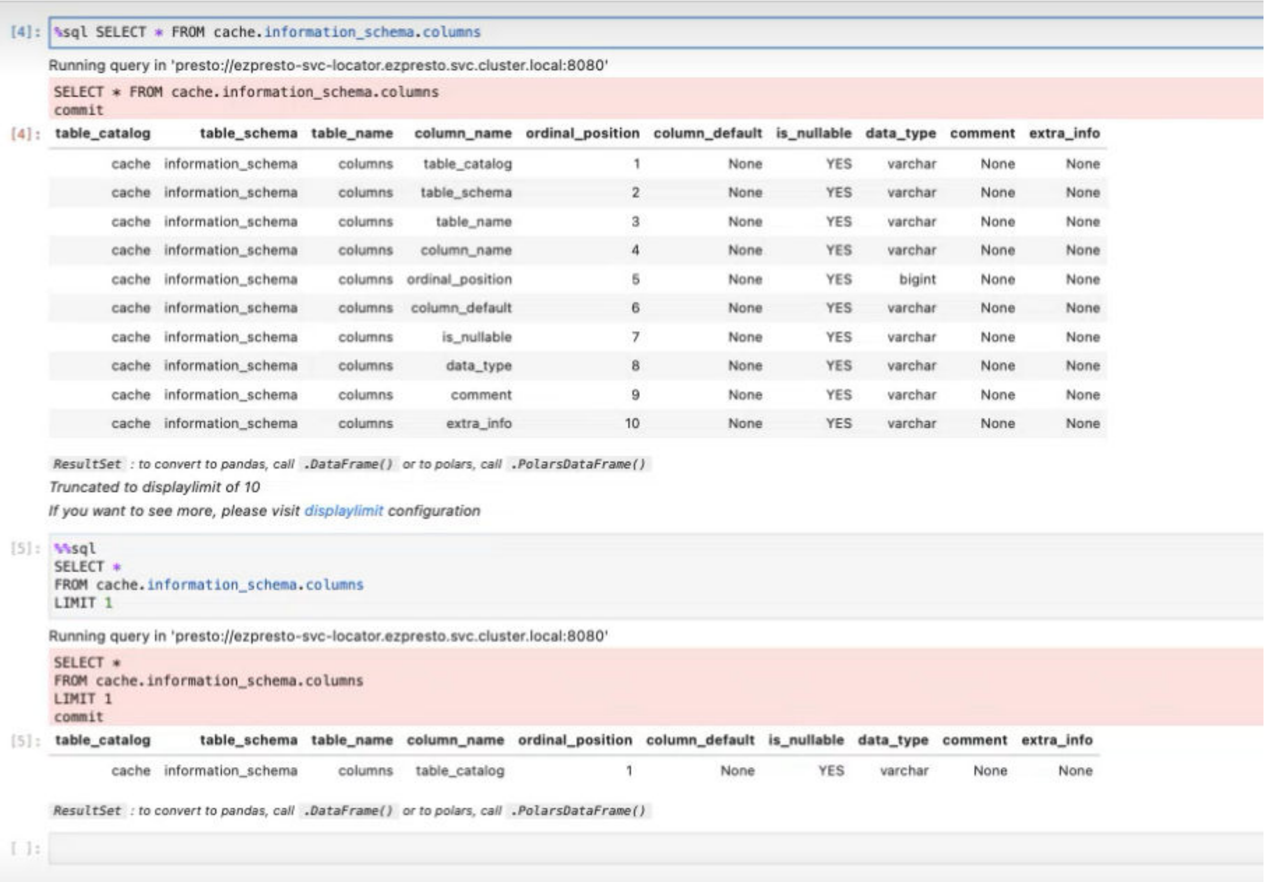
You must use Python kernels to use `%sql` and `%%sql` magic commands. You can directly write and execute SQL queries within a notebook cell. When you run the notebook cell containing `%sql` and your SQL query, the magic command sends the query to the database, runs it, and retrieves the result.

In HPE Ezmeral Unified Analytics Software, you can connect to all SQL databases and submit queries through EzPresto using the `%sql` magic as follows:

```
%sql SELECT * FROM cache.information_schema.columns
```

You can use the `%%sql` magic command to define and run an entire SQL script or block. This means you can write and run a series of SQL statements in the same cell using `%%sql` magic.

The results of the SQL query or queries are displayed in the notebook as a table that makes it easy to analyze and visualize the data.



```
[4]: %sql SELECT * FROM cache.information_schema.columns
```

Running query in 'presto://ezpresto-svc-locator.ezpresto.svc.cluster.local:8080'

```
SELECT * FROM cache.information_schema.columns
commit
```

[4]:

table_catalog	table_schema	table_name	column_name	ordinal_position	column_default	is_nullable	data_type	comment	extra_info
cache	information_schema	columns	table_catalog	1	None	YES	varchar	None	None
cache	information_schema	columns	table_schema	2	None	YES	varchar	None	None
cache	information_schema	columns	table_name	3	None	YES	varchar	None	None
cache	information_schema	columns	column_name	4	None	YES	varchar	None	None
cache	information_schema	columns	ordinal_position	5	None	YES	bigint	None	None
cache	information_schema	columns	column_default	6	None	YES	varchar	None	None
cache	information_schema	columns	is_nullable	7	None	YES	varchar	None	None
cache	information_schema	columns	data_type	8	None	YES	varchar	None	None
cache	information_schema	columns	comment	9	None	YES	varchar	None	None
cache	information_schema	columns	extra_info	10	None	YES	varchar	None	None

ResultSet : to convert to pandas, call .DataFrame() or to polars, call .PolarsDataFrame()

Truncated to displaylimit of 10

If you want to see more, please visit displaylimit configuration

```
[5]: %%sql
     SELECT *
     FROM cache.information_schema.columns
     LIMIT 1
```

Running query in 'presto://ezpresto-svc-locator.ezpresto.svc.cluster.local:8080'

```
SELECT *
FROM cache.information_schema.columns
LIMIT 1
commit
```

[5]:

table_catalog	table_schema	table_name	column_name	ordinal_position	column_default	is_nullable	data_type	comment	extra_info
cache	information_schema	columns	table_catalog	1	None	YES	varchar	None	None

ResultSet : to convert to pandas, call .DataFrame() or to polars, call .PolarsDataFrame()

[ ]:

%update_token

The `%update_token` magic function updates the cached auth token. If you encounter a JWT token expiration error while running cells in the notebbok, you can resolve it by running the `%update_token` magic function. This function updates the JWT in environment variables and any other locations where the token is utilized.

You can use the `%update_token` to refresh tokens for the following cases:

- Authentication when establishing a connection with PrestoDB.
- Authentication with local s3 minio object storage.
- Authentication with KServe external API.

Getting Help

To display help about a magic command, enter the command followed by a ? (question mark). For example:

```
%manage_spark?
```

Creating the Conda Environment

Describes how to create a `conda` environment in HPE Ezmeral Unified Analytics Software.

You can use `conda` package management system to create a new virtual environment.

To create a virtual environment which includes the Python version and `ipykernel` package, run:

```
conda create -n <your-env-name> python=<python-version> ipykernel
```

NOTE: To create a conda environment, you must use a notebook with at least 3 CPU and 3 Gi of memory.

For example: The following command creates a new environment named `py27`, which includes Python version 2.7 and the `ipykernel` package.

```
conda create -n py27 python=2.7 ipykernel
```

You can also create a custom Python kernel using the `%createKernel` magic command. For details, see Notebook Magic Functions on page 445.

Accessing MinIO S3 using Boto3

Describes how to use Boto3 to interact wtih MinIO from a Jupyter Notebook.

You can use Boto3 to interact with Minio S3 services. Boto3 is a Python library that enables you to create, configure, and manage S3 services from a Jupyter Notebook.

The following example shows you how to run Boto3 in a Jupyter Notebook to list the existing Minio S3 buckets:

```
import boto3
import os

access_key_id = os.environ["AUTH_TOKEN"]
secret_access_key = "xxx"

LOCAL_S3_PROXY_SERVICE_URL = 'http://
local-s3-service.ezdata-system.svc.cluster.local:30000'

s3 = boto3.client('s3',
aws_access_key_id=access_key_id,
aws_secret_access_key=secret_access_key,
endpoint_url=LOCAL_S3_PROXY_SERVICE_URL
)

s3.list_buckets()
```

NOTE: If your token has expired, run the following magic command to refresh your tokens.

```
%update_token
```

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. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowlegements

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX® is a registered trademark of The Open Group.

All third-party marks are property of their respective owners.

Glossary

Provides links to the HPE glossary and other informative content.

HPE Glossary

Find definitions of enterprise IT terms across the HPE portfolio and links to related information and resources in the HPE Glossary.

Unified Analytics

For a general definition of Unified Analytics and links to additional information, see Unified Data Analytics.

HPE Ezmeral Unified Data Analytics

For a marketing overview, see HPE Ezmeral Unified Data Analytics.