Hewlett Packard
Enterprise

HPE AI Essentials Software 1.9.x Documentation

Published: August 2025
Edition: 1.9.x

HPE AI Essentials Software 1.9.x Documentation

Abstract

This documentation is intended for Private Cloud AI Administrators and Private Cloud AI Users who want to use HPE AI Essentials Software to build end-to-end Generative AI solutions. For Private Cloud AI Administrators, this documentation describes how to connect external data sources, perform data, user, and resource management tasks, deploy pre-built GenAI solution accelerators, monitor the cluster, and configure tools and frameworks. For Private Cloud AI Users, including Data Engineers, Data Scientists, and Data Analysts, this documentation describes how to use the included tools and frameworks to perform ML and AI tasks, such as designing and implementing data pipelines, managing models, building and deploying Retrieval-Augmented Generation (RAG) solutions with NVIDIA NIM, and using the Playground to fine-tune and interact with AI solutions.

Published: August 2025
Edition: 1.9.x

Notices

Acknowlegements

Table of contents

About

Provides an overview of HPE AI Essentials Software.

HPE AI Essentials Software is part of the HPE Private Cloud AI software and data foundation layer that accelerates the time from AI pilot to production. HPE AI Essentials Software provides a ready-to-run set of AI and open-source tools to build end-to-end Gen AI solutions.

You can access Gen AI to utilize model endpoints, Knowledge Base, and Playground. Gen AI enables you to generate API tokens, integrate Retrieval-Augmented Generation (RAG) with GPU-Optimized Large Language Models, and interact with models for real-time testing and fine-tuning. To learn more, see Gen AI.

From novice to expert, HPE AI Essentials Software provides every user immediate access to a complete suite of proprietary and open-source tools. The low-code features and automated accelerators empower beginners to quickly build generative AI applications such as chatbots. Developers benefit from APIs, notebooks, and advanced tools that enable rapid experimentation, iteration, and deployment, eliminating IT bottlenecks for resource acquisition and provisioning.

HPE AI Essentials Software is:
Simple

- 3-click GenAI workload deployment
- Developer self-service access
- Composable and extensible

Secure

- Secure and unified access to all data
- Role-Based Access Control (RBAC)
- Workload isolation for data users

Cloud-Managed

- 1-click expansion
- Automated life cycle management
- Comprehensive resource management

HPE AI Essentials Software groups open-source tools by persona to tailor the user experience:
Data Engineers

- **Airflow**, a tool for building and managing complex data pipelines with automated ETL workflows. To learn more, see Airflow.
- **EzPresto**, a high-performance, distributed SQL query engine that runs performant SQL queries on data, wherever the data is stored. To learn more, see EzPresto.
- **Superset**, a business intelligence (BI) web application, integrated with EzPresto, that transforms queries into interactive dashboards and visualizations. To learn more, see Superset.

Data Analysts

- **Livy**, an API that removes complexities and enables you to interact with Spark through the tools and interfaces you are familiar with. To learn more, see Creating Interactive Sessions.
- **Spark History**, a web interface that displays details about Spark applications that run in the cluster and assists in monitoring, tuning, debugging and reproducing queries. To learn more, see Spark History Server.
- **Spark Operator** on Kubernetes plays a critical role in providing a stable and scalable Spark environment. You do not interact directly with the Spark Operator. To learn more, see Creating Spark Applications.

Data Scientists

- **Kubeflow** brings the benefits of containerization and orchestration to machine learning (ML), making it easier to scale, reproduce, and manage complex ML pipelines. To learn more, see Kubeflow.
- **Ray** delivers a unified framework for scaling AI and Python applications. Ray excels at distributed computing making it an effective tool for the computationally intensive tasks common in machine and deep learning. To learn more, see Ray.
- **MLflow** manages ML lifecycles, from experimentation and tracking to model deployment and monitoring through a centralized platform for tracking experiments, comparing results, and deploying models. To learn more, see MLflow.
- **MLIS** is a solution designed to simplify and control the deployment, management, and monitoring of machine learning (ML) models at any scale. Built to manage large language models (LLMs), MLIS helps ML practitioners to efficiently launch and maintain ML models without requiring deep specialized expertise. To learn more, see MLIS.

Get Started

Describes how to get started with HPE AI Essentials Software.

The following sections provide links to topics for administrators and members to get started with HPE AI Essentials Software.

Private Cloud AI Administrator

The Private Cloud AI Administrator may be interested in the following topics:

- Identity and Access Management
- Importing Frameworks and Managing the Application Lifecycle

- Configuring Endpoints

Private Cloud AI User

The Private Cloud AI User (non-administrative users) may be interested in the following topics:

- Data Engineering
- Data Analytics
- Data Science
- Notebooks

Subtopics

Tutorials
Provides a set of tutorials that you can use to experience HPE AI Essentials Software and the included applications, such as tutorials for data science and data analytics workflows with notebooks and applications such as Spark, MLflow, Airflow, and EzPresto.
Related Documentation
This page describes end-user documentation for the HPE Private Cloud AI solution.
Third-Party Licenses
Additional License Authorizations
Provides a link to the HPE AI Essentials Software additional license authorizations (ALA) information.

Tutorials

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

Be sure to review the following documentation before using the notebook files on GitHub.

Subtopics

Data Source Connectivity and Exploration
Provides basic steps for using the Data Engineering space within HPE AI Essentials Software.
BI Reporting (Superset) Basics
Provides basic steps for using the BI Reporting (Superset) space within HPE AI Essentials Software.
Financial Time Series Workflow
Describes how to use HPE AI Essentials 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.
MLflow Bike Sharing Use Case
Provides an end-to-end workflow in HPE AI Essentials Software for an MLflow prediction model to determine bike rentals per hour based on weather and time.
MNIST Digits Recognition Workflow
Provides an end-to-end workflow in HPE AI Essentials Software for an MNIST digits recognition example.
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 AI Essentials Software.
Retail Store Analysis Dashboard (Superset)
Provides an end-to-end workflow example for a retail store analysis scenario in HPE AI Essentials Software using EzPresto and Superset.
Running Independent Tune Trials (Ray Tune)
Provides an end-to-end workflow for running independent Tune trials in HPE AI Essentials Software.
Running Ray GPU Example
Describes how to run the Ray GPU example in HPE AI Essentials Software.
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 AI Essentials Software.
Running Very Large Language Models (VLLM) with Ray Serve
Provides an end-to-end example for creating a notebook server and building a very large language model (VLLM) using Ray Serve in HPE AI Essentials Software.
Submitting a Spark Wordcount Application
Provides an end-to-end example for creating and submitting a wordcount Spark Application in HPE AI Essentials Software.

Data Source Connectivity and Exploration

Provides basic steps for using the Data Engineering space within HPE AI Essentials Software.

You can connect to data sources and work with data within the Data Engineering space of HPE AI Essentials 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 AI Essentials Software and use them in your data pipelines.

Tutorial Objective

Although you can perform more complex tasks in HPE AI Essentials 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 AI Essentials Software and the tutorial on one monitor.

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

A – Sign in to HPE AI Essentials Software

Sign in to HPE AI Essentials Software with the URL provided by your administrator.

> **ATTENTION**
> This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

B – Connect Data Sources

Connect HPE AI Essentials 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 panel, select Data Engineering > Data Sources. The **Data Sources** screen appears.
2. 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. On the **Structured Data** tab, click **Add New**.

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

c. 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. On the **Structured Data** tab, click **Add New Data Source**.

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

c. 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. On the **Structured Data** tab, click **Add New Data Source**.

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

c. 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 from 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 panel, 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** side 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** in the **Selected Datasets** side drawer.

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
   ```

   NOTE

   If you want EzPresto to execute queries against the view with the permissions of the user that submits the query, include SECURITY INVOKER when you create the view:

   ```
   create view hiveview.demoschema.customer_info_view SECURITY INVOKER 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
   ```

   If you do not include SECURITY INVOKER, EzPresto executes queries against the view with the permissions of the user that created the view, which can cause queries to fail.

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.

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 AI Essentials 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, 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 AI Essentials Software.

You can add data sets that you created in HPE AI Essentials Software to Superset and visualize the data in dashboards. You can access dashboards (Superset) from the BI Reporting space within HPE AI Essentials 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 AI Essentials 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 AI Essentials Software and the tutorial on one monitor.

IMPORTANT

This tutorial demonstrates how to perform a series of tasks in HPE AI Essentials Software to complete an example workflow. The data and information used in this tutorial is for example purposes only. You must connect HPE AI Essentials Software 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.

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.

> **ATTENTION**
> This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

A – Sign in to HPE AI Essentials Software

Sign in to HPE AI Essentials 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 AI Essentials Software. Once connected to the Presto database, you can access your data sets in HPE AI Essentials Software from Superset.

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

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

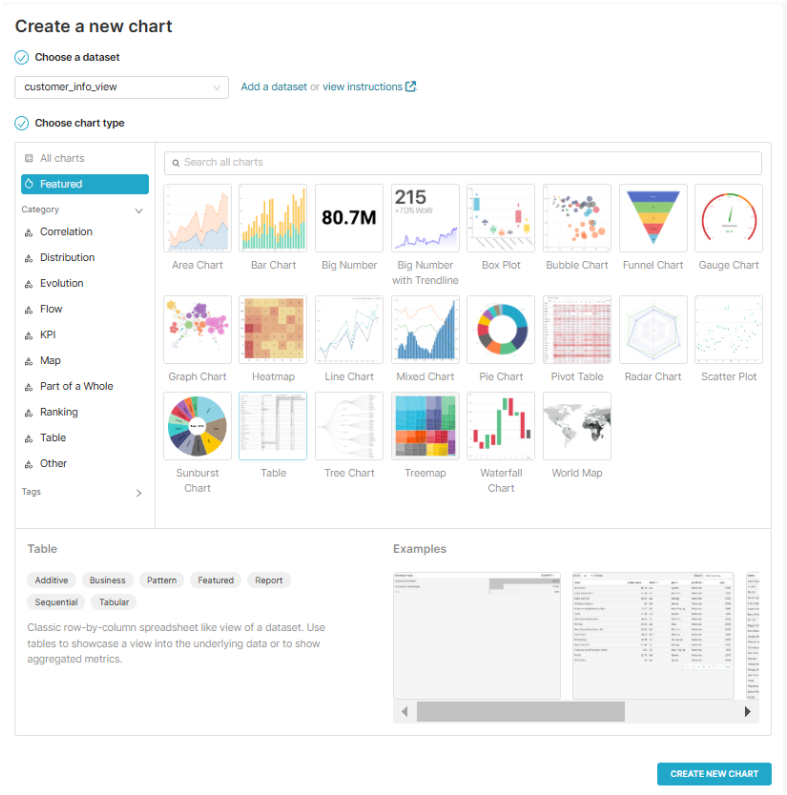
> **TIP**
> If you signed in to Superset through the HPE AI Essentials Software UI, you do not have to enter your user credentials for the EzPresto connection URI because HPE AI Essentials Software authenticates you when you sign in to the system.

1. To open Superset, in the left navigation pane of HPE AI Essentials 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**.

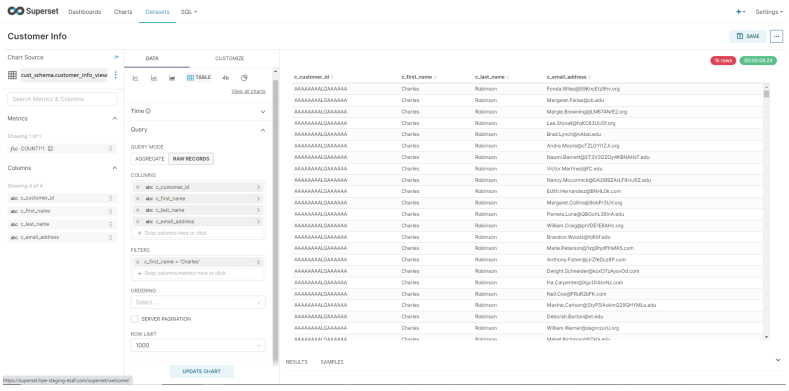
C – Add a Data Set to a Chart

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 **Featured** and choose **Table**.
6. Click **CREATE NEW CHART**.



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.

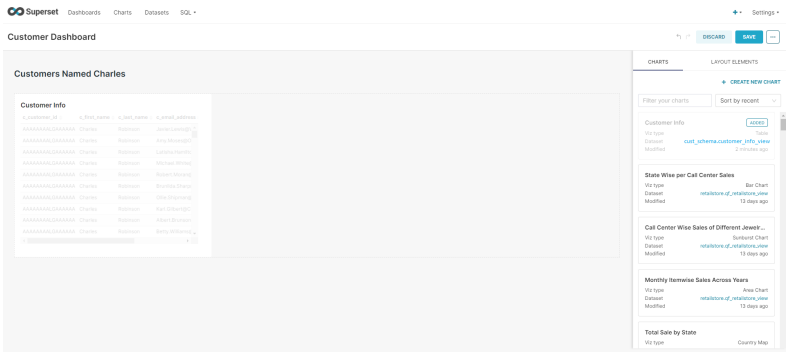


D – Create a Dashboard and Add the Chart

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.



End of Tutorial

You have completed this tutorial. This tutorial demonstrated the integration of the HPE AI Essentials Software SQL query engine (EzPresto) 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), which shows you how to create a database connection, visualize data, and monitor queries used in visualizations.

Financial Time Series Workflow

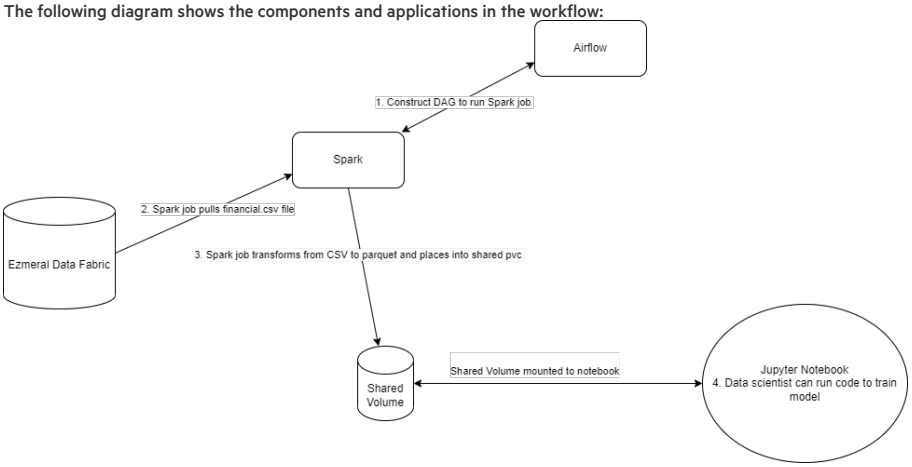
Describes how to use HPE AI Essentials Software to run a Spark application from an Airflow DAG and then run a Jupyter notebook to analyze and visualize data that the Spark application puts into a shared directory in the shared volume that the data scientist's notebook is mounted to.

> **IMPORTANT**
> Be sure to review the following documentation before using the notebook files on GitHub.

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:



Steps

Sign in to HPE AI Essentials Software and perform the following steps:

1. Prerequisites
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

ATTENTION

This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

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:

NOTE

If you are an administrator and you want to complete this tutorial, perform the steps on the **For Administrators** section and then complete the steps on the **For Members** section. If you are a member and you want to complete this tutorial, your administrator must complete the steps on the **For Administrators** section before you complete the steps on the **For Members** section.

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 aie-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/aie-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)
```

For Members

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

1. In the HPE AI Essentials Software, go to Tools & Frameworks.
2. Click Open on Kubeflow.
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/aie-tutorials/current-release/Data-Analytics/Spark` directory to the `/mnt/user` 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/aie-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/aie-tutorials/current-release/Data-Science/Kubeflow` directory, perform:

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

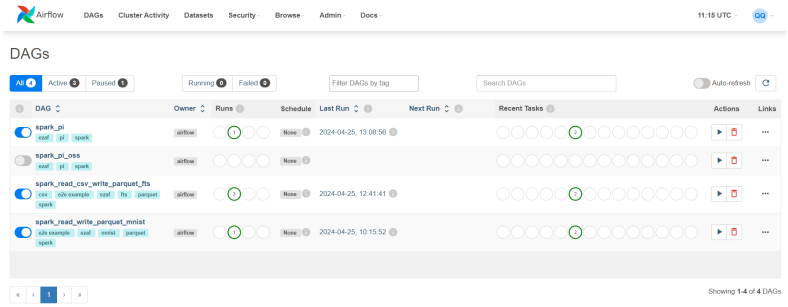
```
import os
namespace = os.getenv('NOTEBOOK_NAMESPACE')
!sed -e "s/\$AUTH_TOKEN/$AUTH_TOKEN/" /mnt/user/object_store_secret.yaml.tpl >
```

```
/tmp/object_store_secret.yaml
!kubectl apply -f /tmp/object_store_secret.yaml -n {namespace}
```

A - Run a DAG in Airflow

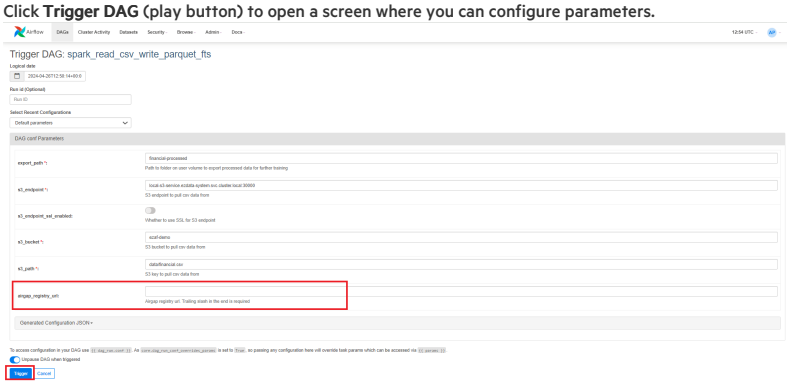
In Airflow, run the DAG named `spark_read_csv_write_parquet_fts`. The DAG runs a Spark application that reads CSV data `(financial.csv)` from an S3 bucket, transforms the data into Parquet format, and writes the transformed Parquet data into the shared volume.
Run the DAG

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



> **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 to learn how to configure your repository.

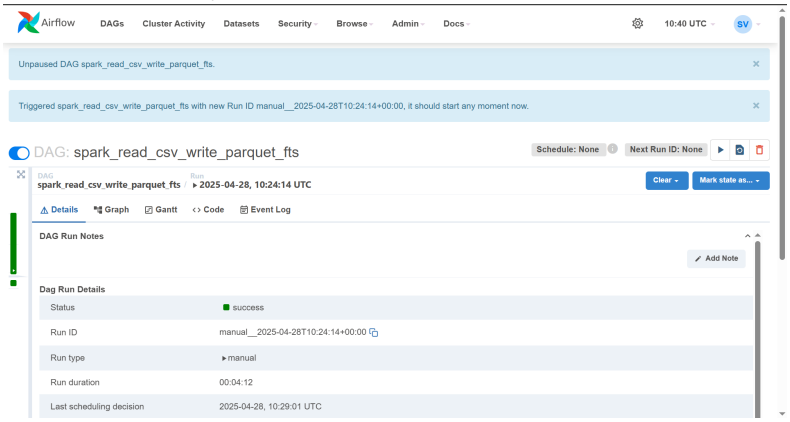
4. Click Code to view the DAG code.
5. Click Graph to view the graphical representation of the DAG.
6. Click **Trigger DAG** (play button) to open a screen where you can configure parameters.



7. 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"
```

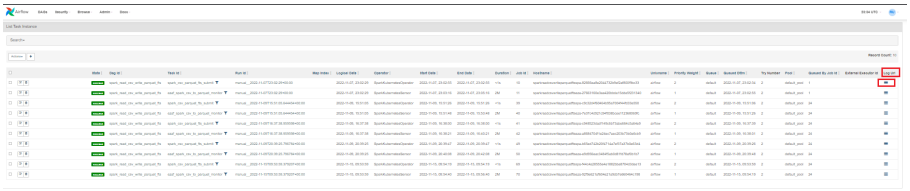
8. To view details for the DAG, click Details.



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"
```



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.

11. Copy the `financial-processed` folder from the `shared` to the `user` folder.

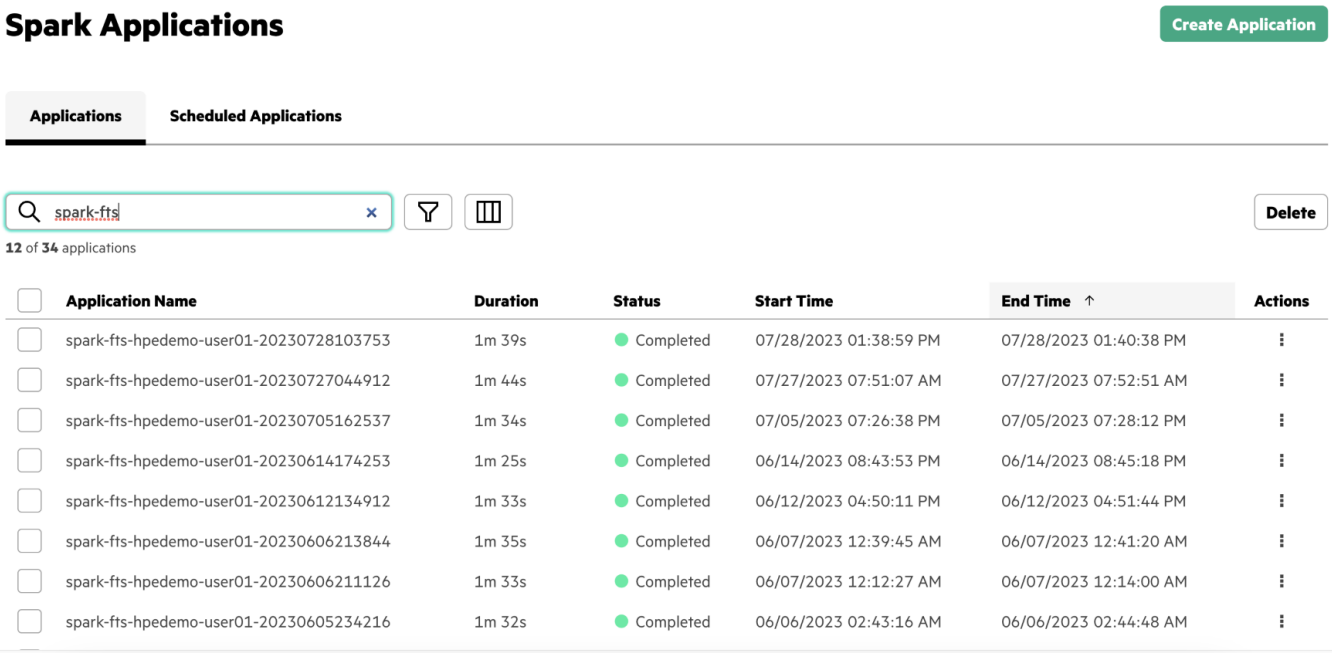
B – View the Spark Application

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 click Open on Spark Operator.

.



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.
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 Notebook Launcher, select the second cell of the notebook and click **Run the selected cells and advance** (play icon).
4. 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 AI Essentials Software for an MLflow prediction model to determine bike rentals per hour based on weather and time.

> **IMPORTANT**
> Be sure to review the following documentation before using the notebook files on GitHub.

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 AI Essentials 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 AI Essentials Software and perform the following steps:

> **ATTENTION**
> This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

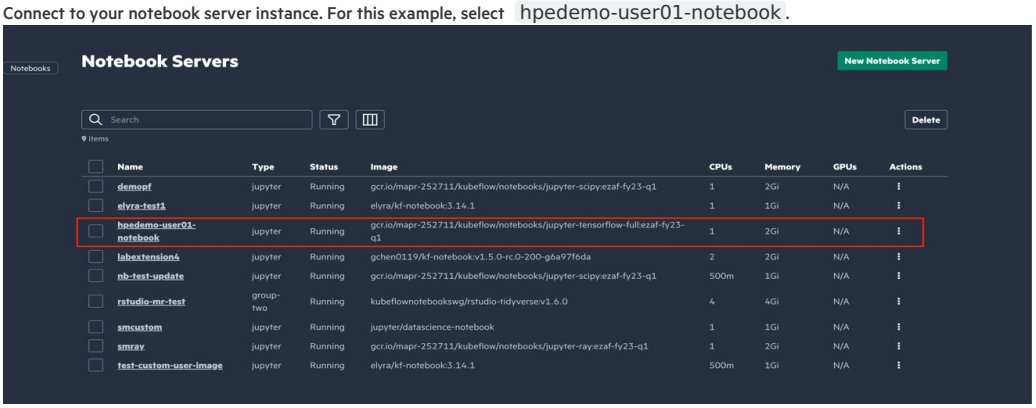
Run the Bike Sharing Use Case

Track Experiment, Runs, and Register a Model in MLflow

Use the Model for Prediction

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`.



3. Copy the `MLFlow` folder from the `shared/aie-tutorials/current-release/Data-Science/MLflow` directory into the `<username>` 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 `aie-tutorials/Data-Science`.

d. Navigate back to the `shared` directory.

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

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

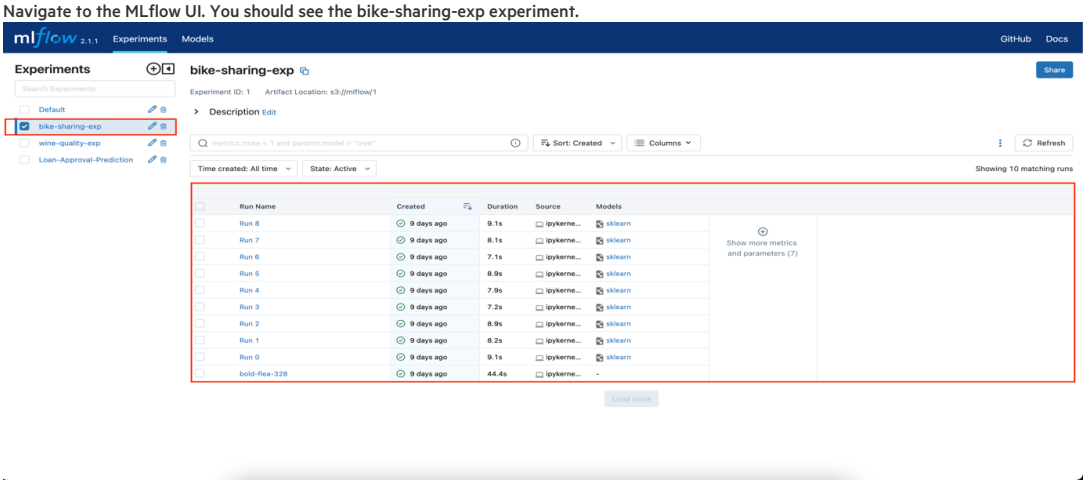
```
os.environ["AWS_ACCESS_KEY_ID"] = os.environ['MLFLOW_TRACKING_TOKEN']
os.environ["AWS_SECRET_ACCESS_KEY"] = "s3"
os.environ["AWS_ENDPOINT_URL"] = 'http://local-s3-service.ezdata-
system.svc.cluster.local:30000'
os.environ["MLFLOW_S3_ENDPOINT_URL"] = os.environ["AWS_ENDPOINT_URL"]
os.environ["MLFLOW_S3_IGNORE_TLS"]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

5. Run the notebook cells.
   Running the notebook returns the details of the best model:

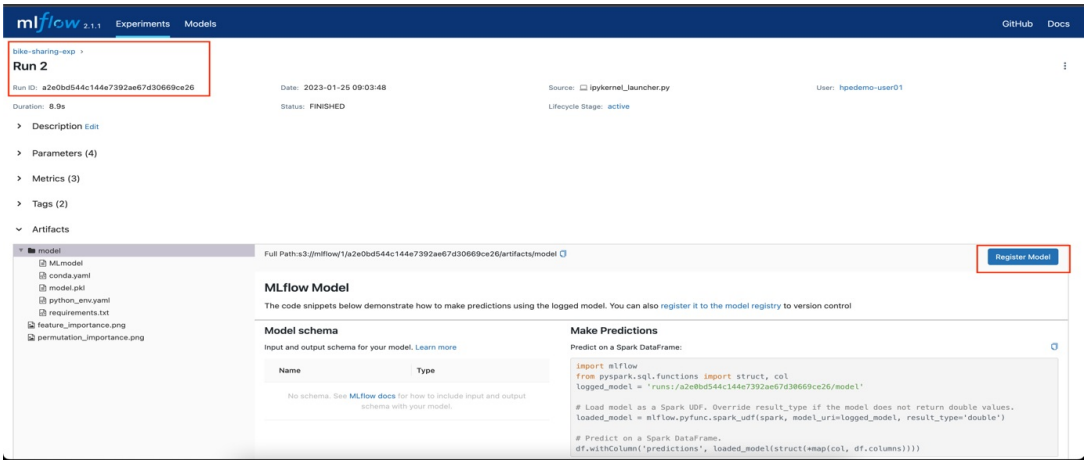
```
Best run info:
Run id: a2e0bd544c144e7392ae67d30669ce26
Run parameters: {'learning_rate': '0.1', 'max_depth': '6', 'Training size': '62568', 'Test size': '5214'}
Run score: RMSE_CV = 44.9510


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

Track Experiment, Runs, and Register a Model in MLflow

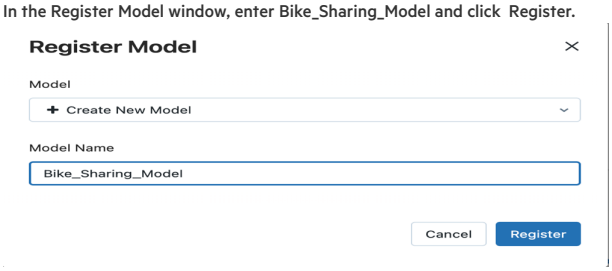
1. Navigate to the MLflow UI. You should see the bike-sharing-exp experiment.



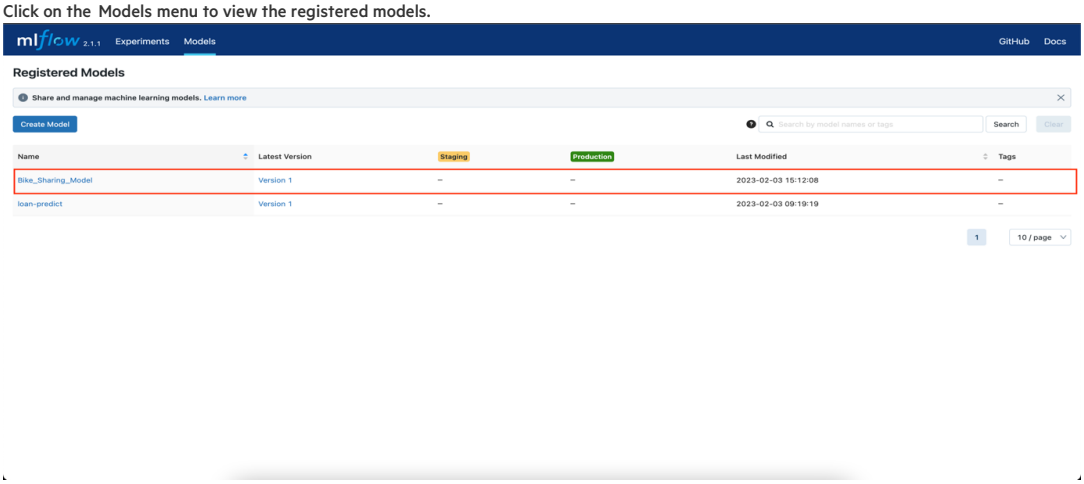
2. Select the best model and then select Register Model. In this example, the best model is run 2.



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



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



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` service goes into the running state.
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

MNIST Digits Recognition Workflow

Provides an end-to-end workflow in HPE AI Essentials Software for an MNIST digits recognition example.

> **IMPORTANT**
> Before you use the notebook files in GitHub, review this document.

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 AI Essentials 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

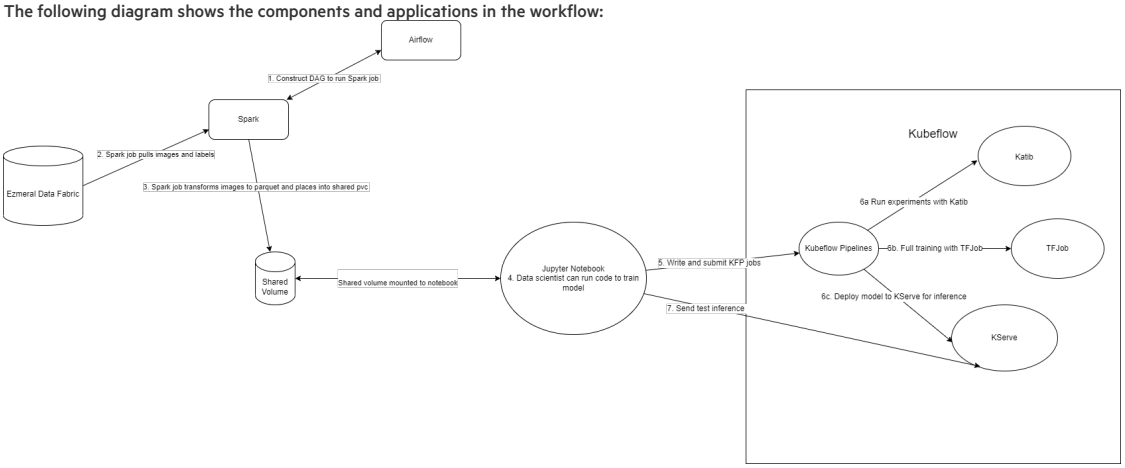
Coded Airflow DAG that runs the Spark application.

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:



Steps

Sign in to HPE AI Essentials Software and perform the following steps:

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

Prerequisites

> **ATTENTION**
> This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

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, complete the prerequisites for your role:

Role that performs the prerequisite task	Prerequisite

Administrators only	Create an S3 object store bucket and load the data that the Spark application reads from the local-S3 Object Store. Members cannot complete the tutorial until you perform this task.
	To copy the required datasets to the `ezaf-demo` bucket at `data/mnist`, run the following code:
	`# Code to copy mnist digit dataset from aie-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/aie-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_files[i], Bucket=bucket, Key=dest_objects[i])`

Members and **Administrators** (Administrators only need to perform this prerequisite if planning to complete the tutorial.)	If the `MNIST-Digits-Recognition` folder is not in your `<username>` folder, copy the folder from the `shared/aie-tutorials/current-release/Data-Science/Kubeflow` directory into your `<username>` folder. The `shared` directory is accessible to all users; however, you cannot run examples from the `shared` directory. You must move them to your `<username>` directory. If the `MNIST-Digits-Recognition` folder is not available in the `shared/aie-tutorials/current-release/Data-Science/Kubeflow` directory, complete the following steps: 1. Go to **GitHub repository for tutorials**. 2. Clone the repository. 3. Navigate to `aie-tutorials/Data-Science/Kubeflow`. 4. Navigate back to the `<username>` directory. 5. Copy the `MNIST-Digits-Recognition` folder from the `aie-tutorials/Data-Science/Kubeflow` directory into the `<username>` directory.

Members and **Administrators** (Administrators only need to perform this prerequisite if planning to complete the tutorial.)	Before you can complete the following steps, an administrator must complete the administrator prerequisite (create an S3 object store bucket and load the data that the Spark application reads).
	1. In the HPE AI Essentials Software, go to Tools & Frameworks.
	2. Click Open on Kubeflow.
	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/aie-tutorials/current-release/Data-Analytics/Spark` directory to the `/mnt/user` folder.
	7. In the `<username>/MNIST-Digits-Recognition` folder, open the `mnist_katib_tf_kserve_example.ipynb` file.
	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 os`
	`namespace = os.getenv('NOTEBOOK_NAMESPACE')`
	`!sed -e "s/\$AUTH_TOKEN/$AUTH_TOKEN/" /mnt/user/object_store_secret.yaml.tpl > /tmp/object_store_secret.yaml`
	`!kubectl apply -f /tmp/object_store_secret.yaml -n {namespace}`

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.

To run the **spark_read_write_parquet_mnist** DAG:

1. Go to Airflow using either of the following methods:
   - Click **Data Engineering > Airflow Pipelines**.
   - Click Open on Airflow.

2. In **Airflow**, select **Dags** in the left menu bar.
3. In the **Search Dags** field, paste the name of the DAG: **spark_read_write_parquet_mnist**

> **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 to find the steps to configure your repository.

4. In the **Dag** column, click **spark_read_write_parquet_mnist**.
5. On the **spark_read_write_parquet_mnist** Dag page, click the **Code** tab to see the code for the DAG.
6. In the upper right corner, click **Trigger** to run the DAG. The **Trigger DAG - spark_read_write_parquet_mnist** window appears. This window allows you to configure parameters; however, do not change any parameters for this tutorial.
7. Click Trigger. You can monitor the state of the DAG by clicking the **Runs** tab. If the DAG runs successfully, you can locate the data inside the following directory in your notebook server:

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

8. To view details for the DAG, click the **Details** tab, or click **Events** to view the audit log events.

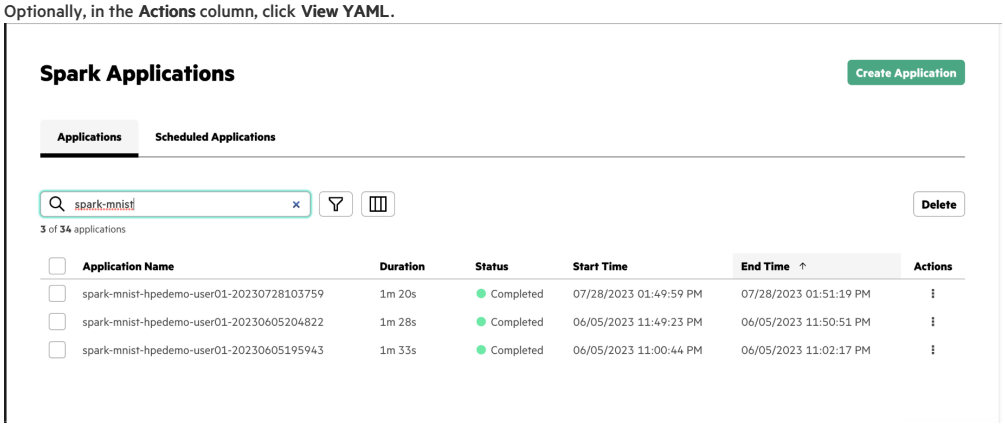
> **IMPORTANT**
>
> The cluster clears the logs that result from the DAG runs. The duration after which the cluster clears the logs depends on the Airflow task, cluster configuration, and policy.

9. Copy the `mnist_spark_data` folder from the `shared` directory to the `admin-63407545`.

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 Open on **Spark Operator**.
2. Identify the **spark-mnist-<username>-<timestamp>** application, for example **spark-mnist-hpedemo-user01-20230728103759**, and view the status of the application..
3. Optionally, in the **Actions** column, click **View YAML**.



C- Update Path of Spark Generated Results

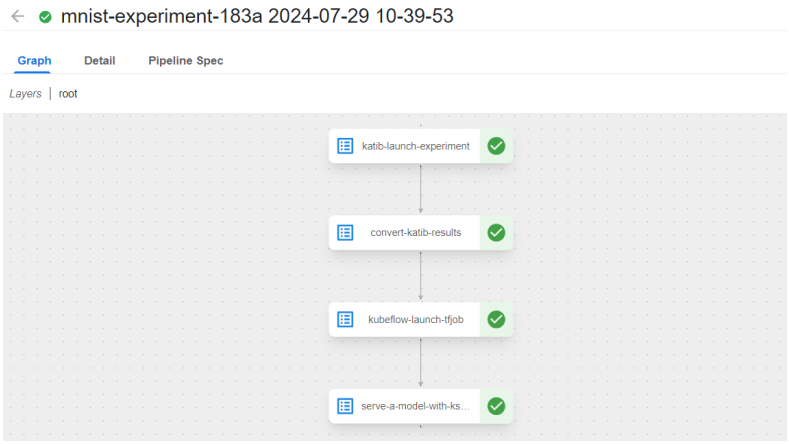
1. Open `mnist_katib_tf_kserve_example.ipynb` file.
2. In the third cell of the `mnist_katib_tf_kserve_example.ipynb` file, update the user folder name as follows:

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

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.



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 AI Essentials 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 AI Essentials Software.

Prerequisites

- Be sure to review the following documentation before using the notebook files on GitHub.
- Sign in to HPE AI Essentials 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:

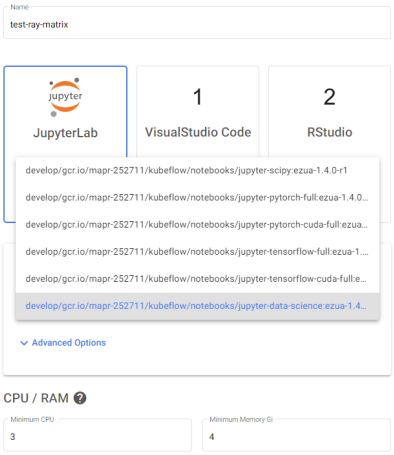
1. Generate a synthetic dataset of rental properties with attributes such as square footage, number of bedrooms, number of bathrooms, and furnishing status to train the prediction model.
2. Format and input the generated data to the model for training.
3. Use the **Random Forest** model to predict the 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.

> **ATTENTION**
> This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

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.



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.

> **NOTE**
> If you do not see the `Ray-Serve` folder in the `<username>` directory, copy the folder from the `shared/aie-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.
>
> If the `Ray-Serve` folder is not available in the `shared/aie-tutorials/current-release/Data-Science/Ray` directory, perform:
>
> a. Go to **GitHub repository for tutorials**.
> b. Clone the repository.
> c. Navigate to `aie-tutorials/Data-Science/Ray`.
> d. Navigate back to the `<username>` directory.
> e. Copy the `Ray-Serve` folder from the `aie-tutorials/Data-Science/Ray` directory into the `<username>` directory.

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.

10. View the deployed application in Ray Dashboard under the Serve tab.
    a. Click the Tools & Frameworks icon on the left navigation bar.
    b. Click Open on Ray.
    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:

    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 AI Essentials 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 AI Essentials 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 AI Essentials Software and connects HPE AI Essentials Software 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 AI Essentials Software includes the following components and applications to support an end-to-end workflow for this scenario:

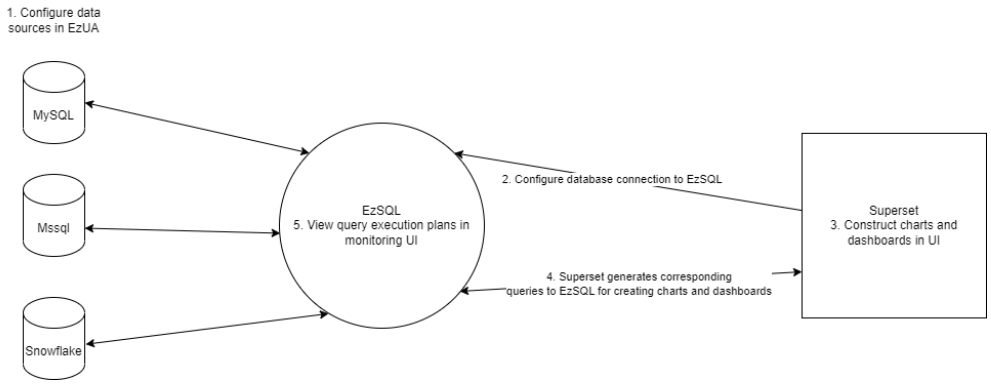
EzPresto

: 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 HPE AI Essentials Software 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:



Steps

Sign in to HPE AI Essentials Software and perform the following steps:

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

IMPORTANT

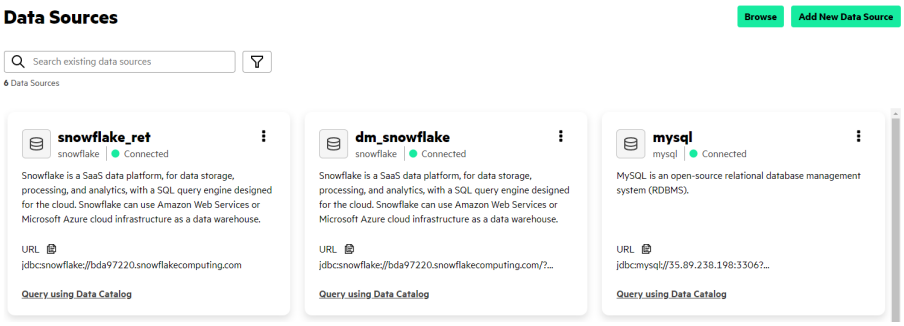
- This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.
- This tutorial demonstrates how to perform a series of tasks in HPE AI Essentials Software to complete an example workflow. The data and information used in this tutorial are for example purposes only. You must connect HPE AI Essentials Software to your own data sources and use the data sets available to you in your data sources.

A - Connect Data Sources

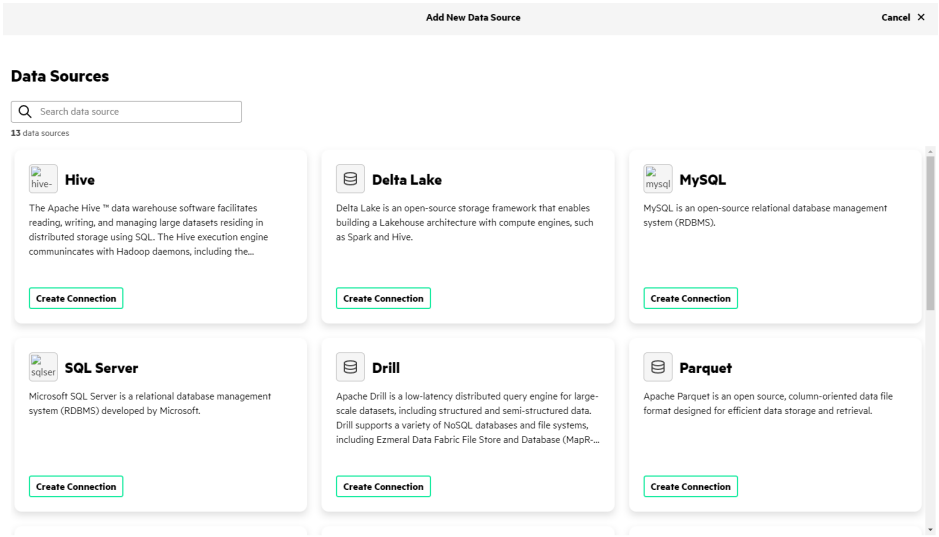
Connect HPE AI Essentials Software to external data sources that contain the data sets (tables and views) you want to work with. This tutorial uses MySQL, SQL Server, and Hive as the connected data source examples.

To connect a data source:

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



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



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 AI Essentials 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
   ```

NOTE

If you want EzPresto to execute queries against the view with the permissions of the user that submits the query, include SECURITY INVOKER when you create the view:

```
create view hiveview.demoschema.qf_retailstore_view SECURITY INVOKER 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
```

If you do not include SECURITY INVOKER, EzPresto executes queries against the view with the permissions of the user that created the view, which can cause queries to fail.

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 AI Essentials 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 AI Essentials Software administrator.

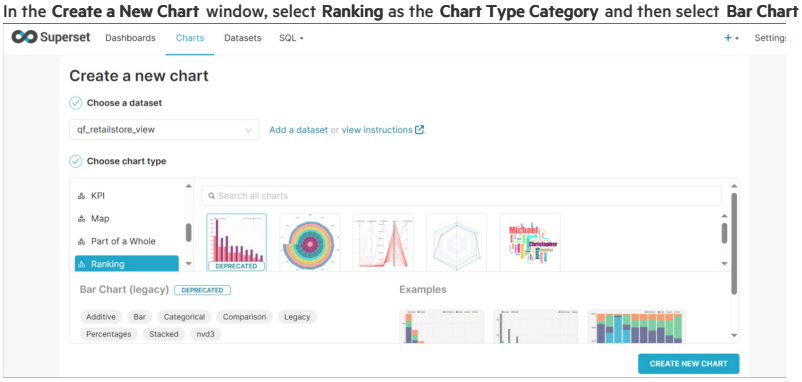
1. To open Superset, in the left navigation pane of HPE AI Essentials Software, select BI Reporting > Dashboards. Superset opens in a new tab.
2. In Superset, select Settings > Database Connections.
3. Click **+DATABASE**.
4. In the **Connect a database** window, select the **Presto** tile.
5. Enter 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 AI Essentials 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 the **Choose Project** window, click **Open**.
3. In Superset, click the **Datasets** tab.
4. Click **+DATASET**.
5. 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*.
6. Click **ADD DATASET AND CREATE CHART**.
7. In the **Create a New Chart** window, select **Ranking** as the **Chart Type Category** and then select **Bar Chart (legacy)**.



8. Click **CREATE NEW CHART**.
9. 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**.

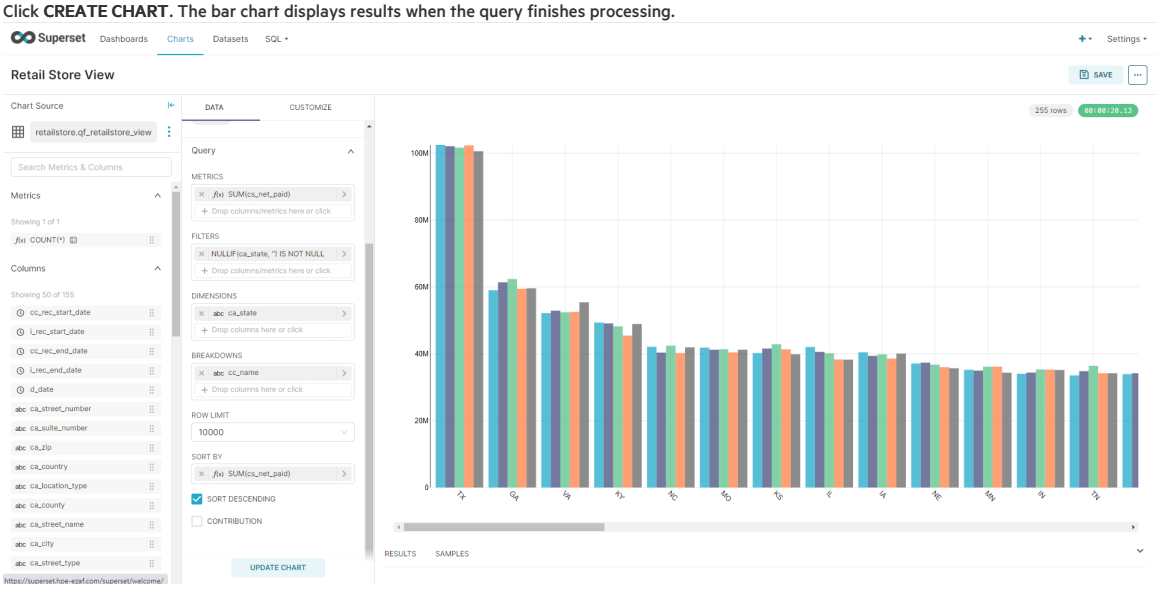
   FILTERS
   a. Click into the **FILTERS** field (located on the DATA tab).
   b. In the window that opens, select the **CUSTOM SQL** tab.
   c. Select the **WHERE** filter and enter the following:

   ```
   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.



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.

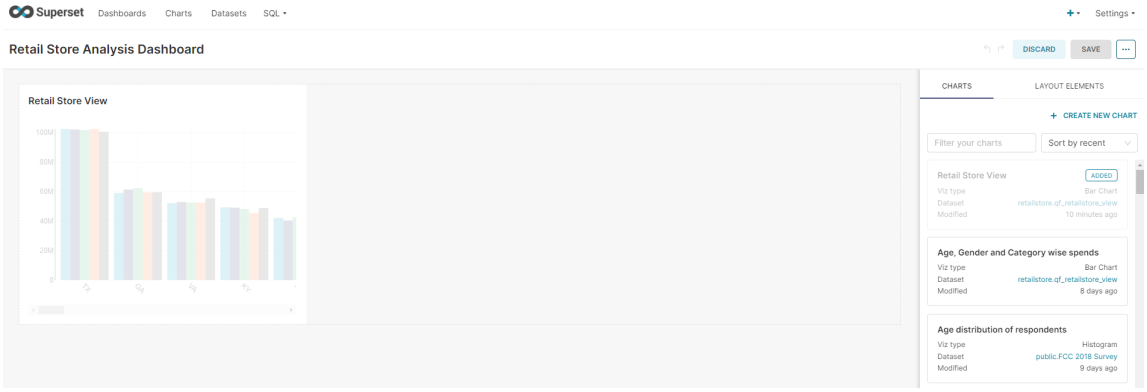
F – Create a Superset Dashboard and Add the Chart (Visualized Data)

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.



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

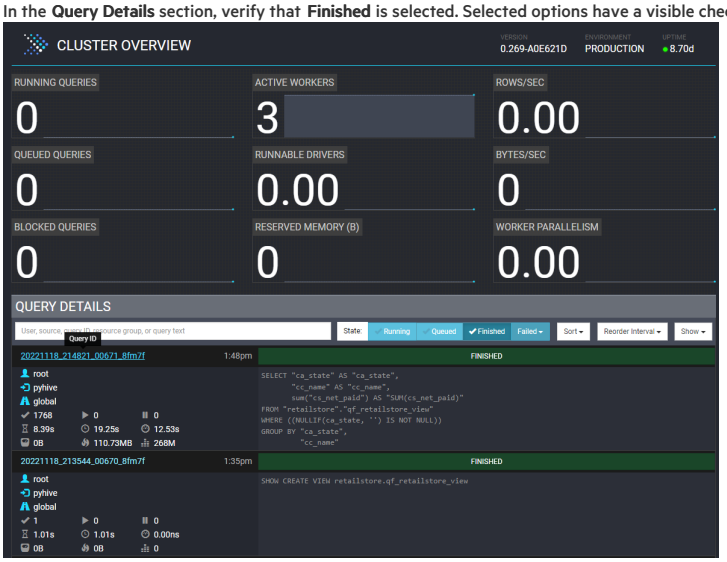
> **NOTE**
>
> Any time you open a chart or dashboard, Superset and the SQL query engine work together to visualize data. Loading a dashboard page triggers the queries against the database. As the queries run, buffering icons display until the data loads. When data is loaded, the visualizations display.

G – Monitor Queries

You can monitor queries generated through Superset through the EzPresto endpoint. You can access the EzPresto endpoint from the EzPresto tile in the Tools & Frameworks space in HPE AI Essentials Software.

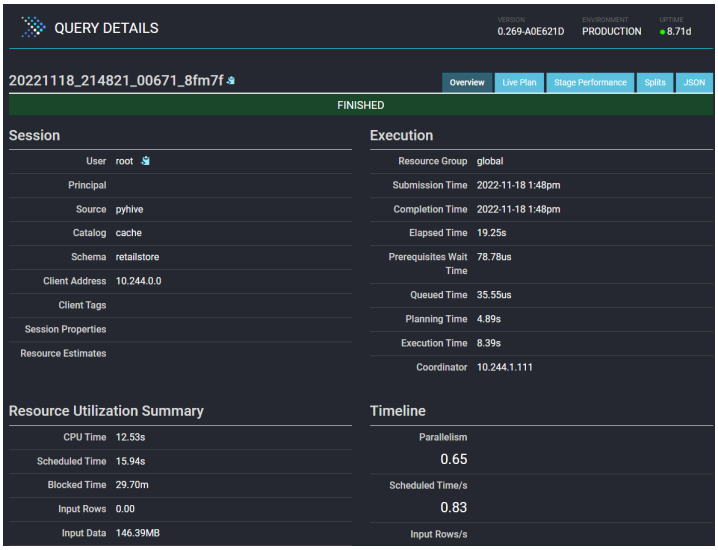
Complete the following steps to monitor the query that the chart generates:

1. Return to the HPE AI Essentials Software UI.
2. In the left navigation bar, select Tools & Frameworks.
3. In the EzPresto tile, click the kebab menu (three dots) and select **View Details**.
4. In the drawer that opens, click the **Endpoint** URL.
5. In the **Choose Project** window, click **Open**.
6. In the **Query Details** section, verify that **Finished** is selected. Selected options have a visible checkmark.

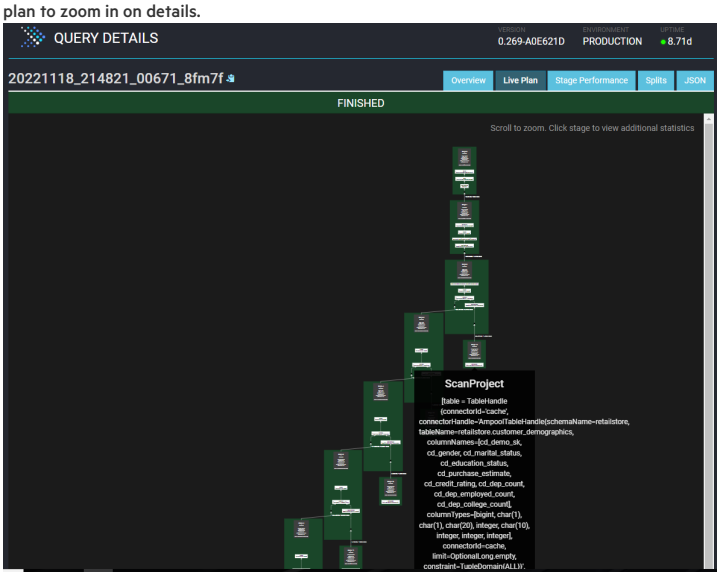


You can see the query that ran to populate the *Retail Store View* bar chart in the *Retail Store Analysis Dashboard*.

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



8. 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.



End of Tutorial

You have completed this tutorial. This tutorial demonstrated the integration of the HPE AI Essentials Software SQL query engine (EzPresto) 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 Cluster Monitoring tool.

Running Independent Tune Trials (Ray Tune)

Provides an end-to-end workflow for running independent Tune trials in HPE AI Essentials Software.

Prerequisites

- Be sure to review the following documentation before using the notebook files on GitHub.
- Sign in to HPE AI Essentials 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:

ATTENTION

- This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

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.
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.

   NOTE

   If you do not see the `Ray-Tune` folder in the `<username>` directory, copy the folder from the `shared/aie-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.

   If the `Ray-Tune` folder is not available in the `shared/aie-tutorials/current-release/Data-Science/Ray` directory, perform:

   a. Go to **GitHub repository for tutorials**.

   b. Clone the repository.

   c. Navigate to `aie-tutorials/Data-Science/Ray`.

   d. Navigate back to the `<username>` directory.

   e. Copy the `Ray-Tune` folder from the `aie-tutorials/Data-Science/Ray` directory into the `<username>` directory.
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 AI Essentials Software.

Prerequisites

- Be sure to review the following documentation before using the notebook files on GitHub.
- Sign in to HPE AI Essentials 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 AI Essentials Software Installation or Enabling GPU Support and Configuring Resources After HPE AI Essentials Software Installation.

About this task

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

You will complete the following steps:

> **ATTENTION**
> This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

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.
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.

   > **NOTE**
   > If you do not see the `Ray-GPU` folder in the `<username>` directory, copy the folder from the `shared/aie-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.
   >
   > If the `Ray-GPU` folder is not available in the `shared/aie-tutorials/current-release/Data-Science/Ray` directory, perform:
   >
   > a. Go to **GitHub repository for tutorials**.
   > b. Clone the repository.
   > c. Navigate to `aie-tutorials/Data-Science/Ray`.
   > d. Navigate back to the `<username>` directory.
   > e. Copy the `Ray-GPU` folder from the `aie-tutorials/Data-Science/Ray` directory into the `<username>` directory.

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 HPE AI Essentials Software 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 AI Essentials Software.

Prerequisites

- Be sure to review the following documentation before using the notebook files on GitHub.
- Sign in to HPE AI Essentials 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.

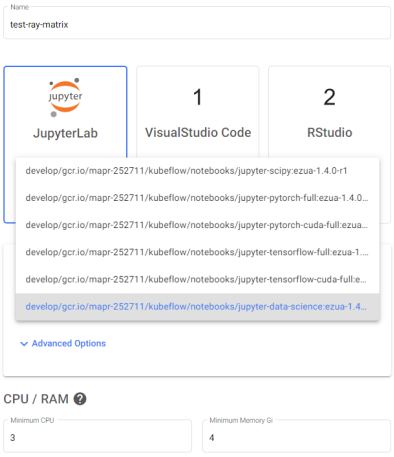
3. Record the duration for matrix generation and multiplication to observe Ray's efficiency under heavy workloads.

Procedure

1. **ATTENTION**

   - This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

   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.



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.

   NOTE

   If you do not see the `Matrix_Multiplication` folder in the `<username>` directory, copy the folder from the `shared/aie-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.

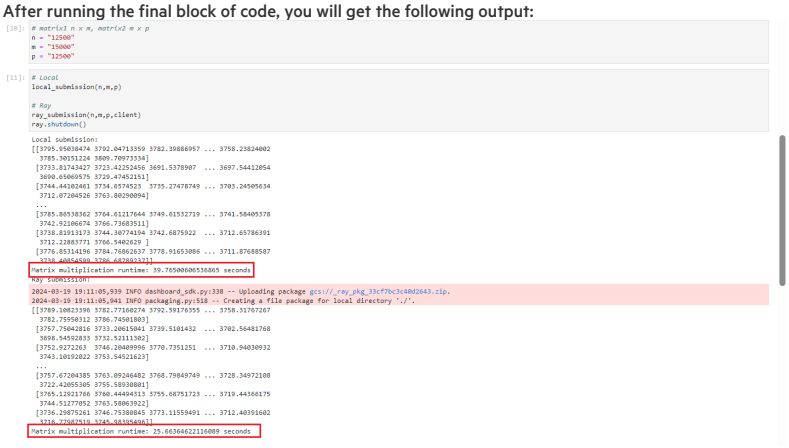
   If the `Ray-CPU` folder is not available in the `shared/aie-tutorials/current-release/Data-Science/Ray/Ray-CPU` directory, perform:

   a. Go to **GitHub repository for tutorials**.
   b. Clone the repository.
   c. Navigate to `aie-tutorials/Data-Science/Ray/Ray-CPU`.
   d. Navigate back to the `<username>` directory.
   e. Copy the `Ray-CPU` folder from the `aie-tutorials/Data-Science/Ray/Ray-CPU` directory into the `<username>` directory.

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:



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.

Running Very Large Language Models (VLLM) with Ray Serve

Provides an end-to-end example for creating a notebook server and building a very large language model (VLLM) using Ray Serve in HPE AI Essentials Software.

Prerequisites

- Be sure to review the following documentation before using the notebook files on GitHub.
- Sign in to HPE AI Essentials 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
     ```

- To install `vllm` in the Ray kernel, run the following commands in the terminal:

  ```
  conda activate ray

  # Proxies are optional based on your env
  export HTTP_PROXY=<>
  export HTTPS_PROXY=<>
  export https_proxy=<>
  && export http_proxy=<>

  pip install vllm==0.6.3
  ```

  NOTE

  You can install a specific version of vLLM, but make sure the chosen version is compatible with the already installed Ray 2.44.1 package. Avoid selecting a version that requires upgrading or downgrading Ray.

- Verify that the GPU support is enabled in your Ray cluster. See Running Ray GPU Example.
- Ensure the cluster has sufficient resources, as vLLM may require additional memory and storage.

About this task

In this tutorial, you will complete the following steps to successfully deploy your VLLM model using Ray Serve:
Prepare the Application

Ensure your Ray Serve application is correctly set up in the `ray_serve_vllm_example.py` file. This includes defining the VLLM model and making sure it is properly referenced during creation.

Build the Ray Serve Application

Use the `serve build` command to generate a configuration file for your deployment. This configuration file will specify how your VLLM application should be deployed.

Configure Runtime Options

You will configure the following:
Workaround for **`--working-dir`** option:

Due to current limitations with Ray Serve's `--working-dir` option, upload the working directory to Google Cloud Storage (GCS) and include it in the deployment configuration.

Install **`vllm`** package:

Ensure the `vllm` package is installed under `pip:packages`.

Deploy the Model

Deploy the VLLM model using the generated configuration file. This step will start the deployment process and make your model available for serving.

Send Sample Requests

After deployment, send sample requests to the deployed VLLM model to test its functionality and ensure it is working as expected.

Shutdown the Deployment

Once you are done with testing, terminate the deployment to free up resources.

Procedure

1. **ATTENTION**

   This tutorial uses components that are not compatible with shared projects. To successfully complete this tutorial, you must run it within your private project. To learn more about Projects, see Projects.

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.

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.

> **NOTE**
>
> If you do not see the `Ray-LLM` folder in the `<username>` directory, copy the folder from the `shared/aie-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.
>
> If the `Ray-LLM` folder is not available in the `shared/aie-tutorials/current-release/Data-Science/Ray` directory, perform:
>
> a. Go to **GitHub repository for tutorials**.
>
> b. Clone the repository.
>
> c. Navigate to `aie-tutorials/Data-Science/Ray`.
>
> d. Navigate back to the `<username>` directory.
>
> e. Copy the `Ray-LLM` folder from the `aie-tutorials/Data-Science/Ray` directory into the `<username>` directory.

4. Ensure that both `ray-serve-vllm-executor.ipynb` and `ray_serve_vllm_example.py` are available in the same dedicated directory.
5. Open the `ray-serve-vllm-executor.ipynb` file in the `<username>/Ray-LLM/vLLM-Serve` directory.
6. Select the first cell of the `ray-serve-vllm-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 `ray_vllm_deployment_config.yaml` configuration file.

7. 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 `ray_vllm_deployment_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()
```

```
2024-10-16 13:55:17,346 INFO dashboard_sdk.py:338 -- Uploading package gcs://_ray_pkg_db274df5fea1cead.zip.
2024-10-16 13:55:17,347 INFO packaging.py:530 -- Creating a file package for local directory './'.
```

8. Open the `ray_vllm_deployment_config.yaml` file.
9. Specify the generated URI and `pip` installation in the `ray_vllm_deployment_config.yaml` file as follows:

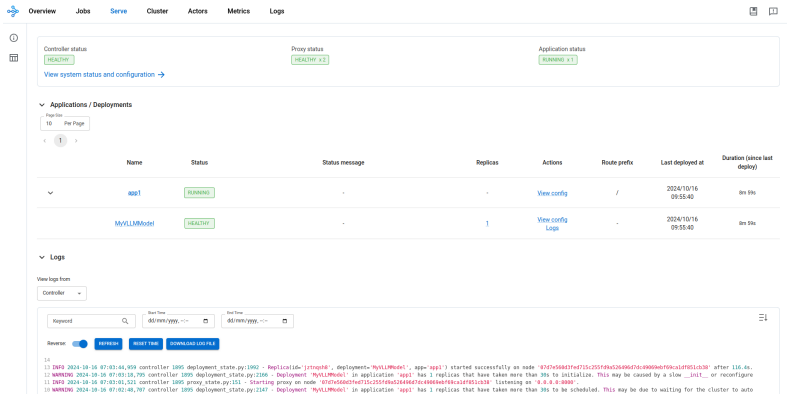
```
runtime_env:
  working_dir: "gcs://_ray_pkg_db274df5fea1cead.zip"
  pip:
    # Specify the versions of packages you want to install
    packages:
      - vllm==0.6.3
  # Optional: Specify additional environment variables if needed
  env_vars:
    http_proxy: "http://<hostname>:<port>"
    https_proxy: "http://<hostname>:<port>"
    HTTP_PROXY: "http://<hostname>:<port>"
    HTTPS_PROXY: "http://<hostname>:<port>"
```

> **TIP**
>
> You can locate a sample `ray_vllm_deployment_config.yaml` in the `/Ray-LLM/vLLM-Serve/resources` folder.

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

```
def send_sample_request():
    import requests

    prompt = "How do I cook rice?"
    sample_input = {"prompt": prompt, "stream": True}
    output = requests.post("http://kuberay-head-svc.kuberay.svc.cluster.local:8000/", json=sample_input)
    for line in output.iter_lines():
        print(line.decode("utf-8"))
```

11. View the deployed application in Ray Dashboard under the Serve tab.
    a. Click the Tools & Frameworks icon on the left navigation bar.
    b. Click Open on Ray.
    c. Navigate to the Serve tab.
    d. Locate and view your deployed application.



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

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

13. To send sample requests to the deployed VLLM model to test its functionality and ensure it is working as expected, run the following blocks of code:

```
In [17]:
def send_sample_request():
    import requests

    prompt = "How do I cook rice?"
    sample_input = {"prompt": prompt, "stream": True}
    output = requests.post("http://kuberay-head-svc.kuberay.svc.cluster.local:8000/", json=sample_input)
    for line in output.iter_lines():
        print(line.decode("utf-8"))

In [18]:
send_sample_request()

{"text": "\n"}
{"text": "What"}
{"text": " kind"}
{"text": " of"}
{"text": " rice"}
{"text": "?"}
{"text": " Green"}
{"text": " tea"}
{"text": ","}
{"text": " milk"}
{"text": " rice"}
{"text": "..."}
{"text": "\n"}
{"text": "K"}
{"text": "orean"}
{"text": " rice"}
```

14. After obtaining results, terminate deployment.

Results

This tutorial shows that by using Ray Serve and Ray cluster deployed in HPE AI Essentials Software, you can efficiently set up and deploy your VLLMs.

Submitting a Spark Wordcount Application

Provides an end-to-end example for creating and submitting a wordcount Spark Application in HPE AI Essentials Software.

Prerequisites

- Be sure to review the following documentation before using the files on GitHub.
- Sign in to HPE AI Essentials 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 AI Essentials 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 and click Open on Spark Operator.
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:

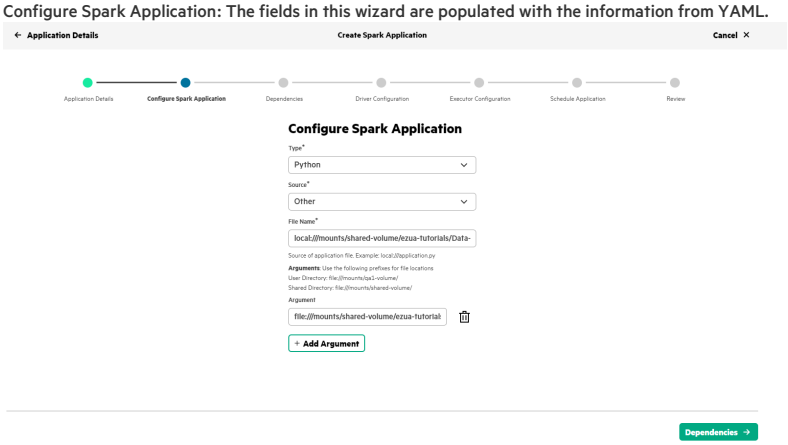
      Update the application name as `username-word-count`.

      > **NOTE**
      > The application name must be unique.

      Description:

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

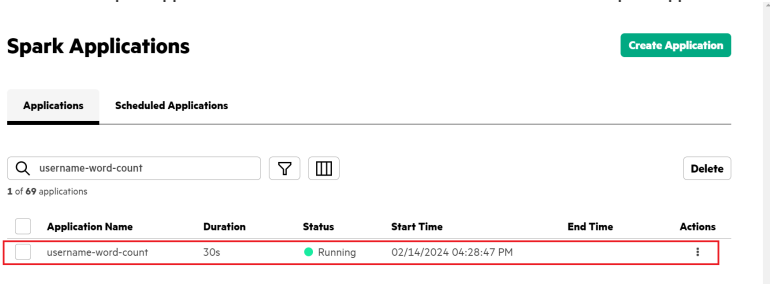
   b. Configure Spark Application: The fields in this wizard are populated with the information from YAML.



   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: 1
      - Memory: 512m
   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: 1
      - Memory per Executor: 512m
   f. Click Schedule Application. If you want to schedule a Spark application, see Creating Spark Applications 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.



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.

Related Documentation

This page describes end-user documentation for the HPE Private Cloud AI solution.

Title	See this title if you are a ...	Summary/Abstract
HPE Private Cloud AI Release Notes	• Cloud administrator • Private cloud AI user	This document describes new features, known issues, resolved issues, and important notes for administering or using the HPE Private Cloud AI system.
HPE Private Cloud AI Developer System Installation Guide	• System installer • Cloud administrator • Security administrator	This guide describes how to install the HPE Private Cloud AI Developer System configuration. The guide covers preparing, installing, configuring, and powering up the Developer System, as well as connecting to the Private Cloud AI service from the Data Services Cloud Console. The guide is intended for use by HPE Private Cloud AI customers.
HPE Private Cloud AI User Guide	• Cloud administrator • Private cloud AI user • Security administrator	This guide is the primary reference for Cloud Administrators, AI Administrators, and AI Users who work with the HPE Private Cloud AI solution. The guide describes the solution architecture, components, and configurations, with a focus on HPE AI Essentials Software - the core engine that powers the private cloud AI infrastructure. It covers essential topics, including user roles and permissions, system management, troubleshooting procedures, and maintenance protocols.
Using Private Cloud AI for Cloud Administrators	• Cloud administrator	This documentation is intended for cloud administrators who are using the Data Services Cloud Console UI to set up, upgrade, and monitor the HPE Private Cloud AI infrastructure. This content is available from within the HPE Private Cloud AI user interface as context-sensitive help and in HPE Support Center as a collection of topics.
HPE AI Essentials Documentation	• Data engineer • Data scientist • Private cloud AI user	This documentation is intended for Private Cloud AI Administrators and Private Cloud AI Users who want to use HPE AI Essentials Software to build end-to-end Generative AI solutions. For Private Cloud AI Administrators, this documentation describes how to connect external data sources, perform data, user, and resource management tasks, deploy pre-built GenAI solution accelerators, monitor the cluster, and configure tools and frameworks. For Private Cloud AI Users, including Data Engineers, Data Scientists, and Data Analysts, this documentation describes how to use the included tools and frameworks to perform ML and AI tasks, such as designing and implementing data pipelines, managing models, building and deploying Retrieval-Augmented Generation (RAG) solutions with NVIDIA NIM, and using the Playground to fine-tune and interact with AI solutions.
HPE AI Essentials Software Overview Video	• Data engineer • Data scientist • Private cloud AI user	The HPE AI Essentials Software video provides a brief tour of the HPE AI Essentials Software interface, highlighting the AI and data-foundation tools within the unified control plane, including NVIDIA NIM, Knowledge Base, and pre-built Solution Accelerators; and provides a look at some of the administrative features, including data access management.
HPE AI Essentials Software Guided Tours	• Data engineer • Data scientist • Private cloud AI user	Explore HPE AI Essentials Software using interactive guided tours. Learn the platform's core functionality, complete key tasks, and understand platform utilization by roles such as Data Scientists and Data Engineers. Step-by-step guidance is available directly within the platform interface for all users. Just click the guided tours icon in the menu to get started.
HPE Private Cloud AI Firmware and Software Compatibility Matrix	• Cloud administrator • Data engineer	This document outlines the software and firmware versions for all components based on the recipe version used in the HPE Private Cloud AI solution deployment. It is intended for Private Cloud AI Users, HPE Integration Center (factory) engineers, cloud administrators, and data engineers. The document is available on the HPE Support Center.

Third-Party Licenses

To download a spreadsheet listing the third-party components in HPE AI Essentials Software in Microsoft Excel format (.xlsx), click **Third-Party Licenses**.

Additional License Authorizations

Provides a link to the HPE AI Essentials Software additional license authorizations (ALA) information.

Administration

Provides information about managing applications and clusters in HPE AI Essentials Software.

Subtopics

Using HPE AI Essentials Software on the Developer System Configuration

Describes the support for the Developer System configuration in HPE AI Essentials Software.

HPE AI Essentials Software fully supports the new Developer System configuration. The Developer System provides the same HPE AI Essentials Software experience inside a smaller hardware footprint. To support the Developer System, HPE AI Essentials is validated to run with a single control node and a single worker node. When deployed on the Developer System, the software runs collocated with a local NFS server on the worker node.

Developer System use cases span the entire AI/ML spectrum but are limited in scale. The Developer System targets small-scale inference and broad prototyping. RAG workflows with very large LLMs and large-scale inference are not supported, due to the presence of only two GPUs in the Developer System.

> **NOTE**
> Developer System does not support Data Lakehouse Gateway.

For more information about the Developer System, see the **HPE Private Cloud AI Administration Guide** or the **HPE Private Cloud AI Developer System Installation Guide**.

Importing Frameworks and Managing the Application Lifecycle

Describes how to import, manage, and secure tools and frameworks in HPE AI Essentials Software.

The Private Cloud AI Administrator can import, run, and manage customized Kubernetes applications and frameworks in HPE AI Essentials Software. The Private Cloud AI Administrator can manage imported applications as well as the applications that were included with HPE AI Essentials Software at the time of installation.

Imported and included applications appear in the Tools & Frameworks screen in HPE AI Essentials 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.

You can also access your favorite applications using the favorite functionality. Select your favorite applications by clicking the star icon in application tiles on the **Tools & Frameworks** screen. Applications that you favor appear at the top of the page. Favorite applications also appear on the **Favorite Tools & Frameworks** tile on the **Home** page of HPE AI Essentials Software.

Importing Custom Kubernetes Applications

You can import your custom Kubernetes applications through the HPE AI Essentials Software 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 AI Essentials Software. HPE AI Essentials Software supports SSO for imported applications.

- To import applications through the HPE AI Essentials Software UI, see Importing Frameworks.
- To import applications using the API, see **https://github.com/HPEEzmeral/byoa-tutorials**.

Managing Tools and Frameworks

HPE AI Essentials Software provides the following options to manage tools and frameworks:

- View Details

- Configure
- Delete imported applications
- Update imported applications

For detailed instructions, see the following:

- Managing Imported Tools and Frameworks
- Configuring Included Frameworks and Viewing Details

Subtopics

Importing Frameworks
Describes how to import frameworks in HPE AI Essentials Software.
SSO Support for Imported Frameworks
Describes SSO support for imported frameworks integrated with native authentication and applications configured with authentication proxy.
Managing Imported Tools and Frameworks
Describes how to configure, delete, and update imported tools and frameworks in HPE AI Essentials Software.
Configuring Included Frameworks and Viewing Details
Describes how to configure tools and frameworks included with the HPE AI Essentials Software installation.

Importing Frameworks

Describes how to import frameworks in HPE AI Essentials Software.

Prerequisites

- Sign in to HPE AI Essentials Software as Private Cloud AI 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"
```

- Configure SSO for the applications you want to import. See SSO Support for Imported Frameworks.
- 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 AI Essentials Software, you can bring your own Kubernetes customized runtime tools 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 AI Essentials 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.

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 AI Essentials 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 AI Essentials Software.

Prerequisites

- Sign in to HPE AI Essentials Software as Private Cloud AI 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:
   View Details

   To view details, select View Details. This opens the side-bar where you can view the application description, version number, chart version number, and the application endpoint.

   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 AI Essentials 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.

> **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 and Viewing Details

Configuring Included Frameworks and Viewing Details

Describes how to configure tools and frameworks included with the HPE AI Essentials Software installation.

Prerequisites: Sign in to HPE AI Essentials Software as Private Cloud AI Administrator.

You can configure the tools and frameworks that were installed with HPE AI Essentials Software from the Tools & Frameworks screen or Settings screen.

Viewing Details

To view details of included frameworks, follow these steps:

1. In the left navigation bar, click Tools & Frameworks.
2. On the application tile, click the three-dots button.
3. Select View Details. This opens the side-bar where you can view the application description, version number, chart version number, and the application endpoint.

Configuring via Tools & Frameworks

To configure the included frameworks from the Tools & Frameworks screen, follow these steps:

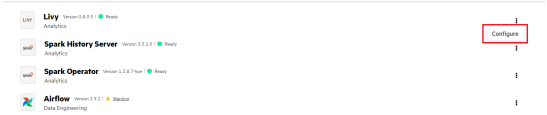
1. In the left navigation bar, click Tools & Frameworks.
2. On the application tile, click the three-dots button.
3. Select Configure to open the editor.
4. In the editor, modify the `values.yaml` file.
5. To apply the changes, click Configure, or to close the editor without any changes, click Cancel.

> **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.



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.

Installing Included Frameworks Post HPE AI Essentials Software Installation

Describes how to install included frameworks post HPE AI Essentials Software installation.

Prerequisites

1. Sign in to HPE AI Essentials Software as Private Cloud AI Administrator.
2. Ensure you have the required CPU and memory resources to install the framework.

About this task

The following frameworks might be excluded during the HPE AI Essentials Software installation but can be installed later as needed:

- EzPresto
- Livy
- MLIS
- Superset

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.

> **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 AI Essentials Software to external S3 object storage in AWS, MinIO, and HPE Ezmeral Data Fabric Object Store.

Administrators can connect HPE AI Essentials 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 AI Essentials Software with the access credentials (access key and secret key); the user does not need the access credentials because HPE AI Essentials Software uses a proxy to communicate with clients.

Clients talk to the HPE AI Essentials 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 AI Essentials Software UI), as well as the bucket they want the client to access.

Connecting HPE AI Essentials Software to Object Storage

Regardless of which object store you connect, 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 AI Essentials 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 AI Essentials Software and cannot be deleted. Do not connect to this data source.

4. Click **Add New Object Store**.
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
  ```

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 HPE GreenLake for File Storage URL.
- **Access Key** - Enter the HPE GreenLake for File Storage access key.
- **Secret Key** - Enter the HPE GreenLake for File Storage 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.

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.

Connecting to HPE Ezmeral Data Fabric

Describes how to connect HPE AI Essentials Software to an external HPE Ezmeral Data Fabric cluster.

To connect HPE AI Essentials 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 AI Essentials Software to an external HPE Ezmeral Data Fabric cluster, you must provide the `maprusersticket` to create a connection that enables users to access HPE Ezmeral Data Fabric from HPE AI Essentials Software.
To get the `maprusersticket`, run the following command on the HPE Ezmeral Data Fabric cluster:

```
sudo cat /opt/mapr/conf/maprusersticket
```

Volume path

This is the path to the volume in HPE Ezmeral Data Fabric that you want to connect to from HPE AI Essentials Software. For information about volumes, see **Managing Data with Volumes**.

Complete the following steps to connect HPE AI Essentials Software to an external HPE Ezmeral Data Fabric cluster:

1. Sign in to HPE AI Essentials 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 `maprusersticket` 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 AI Essentials 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 HPE AI Essentials Software to an external HPE GreenLake for File Storage cluster.

Complete the following steps to connect HPE AI Essentials Software to an external HPE GreenLake for File Storage cluster:

1. Sign in to HPE AI Essentials 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 HPE AI Essentials Software cluster.
Storage	Enter the amount of storage available to HPE AI Essentials Software through the HPE GreenLake for File Storage connection.
Server	Enter the HPE GreenLake for File Storage endpoint that you want HPE AI Essentials Software 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 AI Essentials 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 Storage connection **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 AI Essentials Software and how to configure them.

Configure endpoints in HPE AI Essentials 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 AI Essentials 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.

Use the following OTel endpoint format:

```
<host>:<port>
```

NOTE

The OTel endpoint allows your external monitoring systems to receive cluster metrics and logs in OTEL format. Data persists in Prometheus for 60 days, but with OTEL you can store it in your preferred long-term storage solution.

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

To integrate with cloud-based monitoring solutions requiring URL-based connections, deploy a local OpenTelemetry Collector within the same network as HPE AI Essentials Software. This collector receives telemetry data through the host/port configuration supported by HPE AI Essentials Software and forwards it to the cloud endpoint using the full URL format. For example, see the **Grafana OpenTelemetry Collector Setup** and **Datadog OpenTelemetry Collector Exporter** for platform-specific configurations.

JDBC Endpoint

The JDBC endpoint is automatically created when you install and configure HPE AI Essentials Software.

To connect EzPresto to external applications, see Connecting External Applications to EzPresto via JDBC.

Managing Projects

Describes how to create and delete shared projects. Also describes how to set quotas on shared projects.

Private Cloud AI Administrators can create multiple projects for team collaboration in HPE AI Essentials Software. Private Cloud AI Administrators can also delete projects.

Deleting projects does not delete users from HPE AI Essentials Software; however, it does revoke access to workloads and data sources that the members had access to through the project. If users were granted direct access on data, deleting a project does not affect user access to data.

The following table lists the tasks related to creating a project:

Task	Details and Documentation
1- Create the project	You can create multiple projects for team collaboration. See Creating a Shared Project.
2 - Set a vGPU quota on the project	The vGPU quota sets the maximum number of vGPUs for a shared project. This number can exceed the amount of vGPUs in the cluster, which is listed on the page by **Total Cluster vGPUs**. The system displays a warning if the vGPUs set exceed the cluster vGPUs. See Setting Quotas on a Shared Project.
3 - Add a user to the project	Add at least one user to the project. You can make this user a project admin. The project admin can add additional users to the project. See User Access to Projects.
4 - Grant the project data access	Grant projects access to the data sources, object stores, and model endpoints that team members need to run their applications and workloads. See Project Access to Data.

Creating a Shared Project

These steps describe how to create a new shared project. You cannot create a private project. The system automatically creates a private project for each user the first time they sign in to HPE AI Essentials Software.

To create a shared project:

1. In the left navigation panel, select **Projects**.
2. On the **Projects** page, click **Create New Project**.
3. Enter a unique name for the project.
4. Click **Create Project**. The new project appears on the **Projects** page.

The **Type** column lists projects as **Private** or **Shared**. Shared projects are for collaboration. Private projects are personal and cannot be shared.

Deleting a Shared Project

To delete a shared project:

1. In the left navigation panel, select **Projects**.
2. Locate the project and click the kebab menu in the **Actions** column.
3. Click **Delete**.
4. In the window that appears, type **DELETE** to confirm that you want to perform the action. You must type DELETE in uppercase.
5. Click **Delete**.

Setting Quotas on a Shared Project

The project quota you set can exceed the amount of vGPUs in the cluster, which is listed on the page by **Total Cluster vGPUs**. The system displays a warning if the vGPUs set exceed the cluster vGPUs.

To set the maximum number of vGPUs for a shared project:

1. In the left navigation panel, select **Administration > Resource Management**.
2. Select the **Project Quotas** tab.
3. Locate the project by the project name.
4. In the **Actions** column, click the kebab menu and select **Configure Project Quotas**.
5. Enter the number of vGPUs you want this project to use.
6. Click **Configure**.

GPU Support

Describes GPU resource management and GPUDirect RDMA in HPE AI Essentials Software.

GPUs accelerates deep learning models, data processing, and analytical workloads by using the parallel processing capabilities. GPUs efficiently handle large-scale computations which allows you to train models, create inference services, and process data frames, process SQL queries within Spark, and run experiments using Jupyter notebooks.

HPE AI Essentials Software efficiently utilizes GPU by dynamically reclaiming idle GPU resources. This prevents overallocation and underutilization of the GPU resources and increases efficiency. HPE AI Essentials Software uses a priority policy to ensure high-priority workloads receive the necessary resources. When a workload is finished using its GPU resources, HPE AI Essentials Software initiates GPU resource reclamation making them available for other workloads. For details, see GPU Resource Management.

HPE Private Cloud AI integrates GPUDirect RDMA to enhance GPU performance. GPUDirect RDMA enables the rapid transfer of data between GPUs and other devices by bypassing the CPU and host memory, creating direct GPU-to-GPU communication channels. The NVIDIA Network Operator manages networking components within an HPE AI Essentials Software system and enables the sharing of RDMA network interfaces on compute nodes that have GPU resources. For details, see NVIDIA GPUDirect Remote Direct Memory Access (RDMA).

Subtopics

GPU Resource Management
Describes the GPU idle reclaim policy used for GPU resource management.

NVIDIA GPUDirect Remote Direct Memory Access (RDMA)
Describes NVIDIA GPUDirect Remote Direct Memory Access (RDMA) in HPE AI Essentials Software, including how to configure GPUDirect RDMA in your applications.

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 AI Essentials 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 AI Essentials Software initiates GPU resource reclamation. This involves deallocating the resources and making them available for other workloads.

Custom Scheduler

HPE AI Essentials 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.

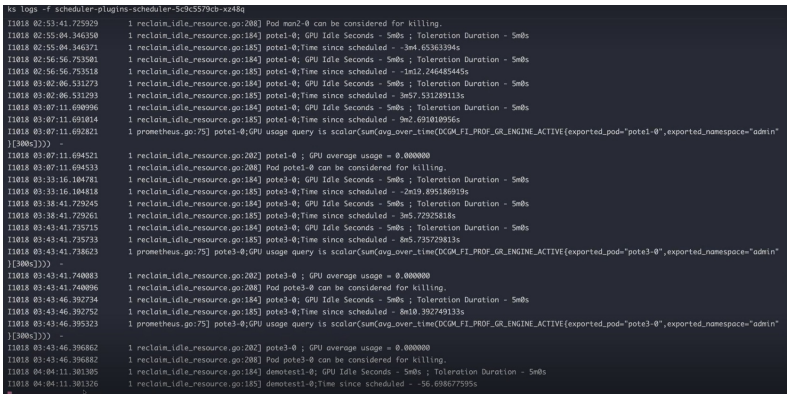
```
~ (0.482s)
ks get pods
NAME                                          READY   STATUS    RESTARTS   AGE
scheduler-plugins-controller-dc8fbd68-2plns   1/1     Running   0          8h
scheduler-plugins-scheduler-5c9c5579cb-xz48q  1/1     Running   0          5h6m
```

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
```



Custom Scheduler Configurations

HPE AI Essentials 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

Every GPU workload for Kubeflow, Spark, Livy, Ray 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 AI Essentials 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.

Priority class and priority level

HPE AI Essentials 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.

Subtopics

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 AI Essentials Software as Private Cloud AI Administrator.
2. In the left navigation bar, click Administration → Resource Management.

You are now in the GPU Control Panel screen.

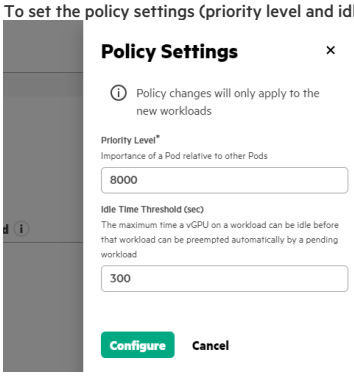
GPU Control Panel

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.



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

> **WARNING**
> Do not modify priority settings when pods are in a *pending* state, as this causes pod failure. Pods must be in either a *running* or *terminated* state when you modify the priority settings.

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.

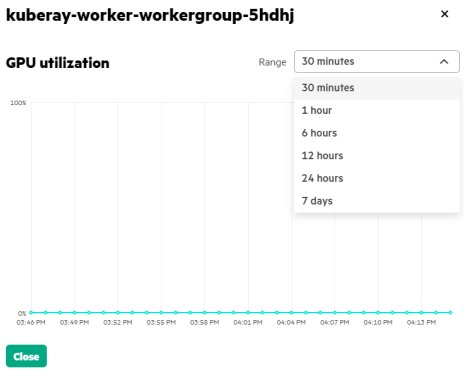
> **NOTE**
> - Changing the **Policy Settings (Priority Level and Idle Time Threshold)** values for **AI Services** to a higher number before deploying the Knowledge Base ensures that the GPU reclaim feature will not reassign the GPU to other workloads in the queue.
> - GPU resources are valuable and sharing them with other workloads during idle periods is important. Be cautious when changing policy settings, as this can impact other 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.



GPU Scheduling Workload Scenarios

Describes GPU scheduling workload scenarios and the notebook example for GPU idle reclaim.

In HPE AI Essentials Software, you can encounter the following GPU scheduling workload scenarios during the GPU idle reclamation.

GPU Idle Reclaim

In HPE AI Essentials 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 AI Essentials 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.

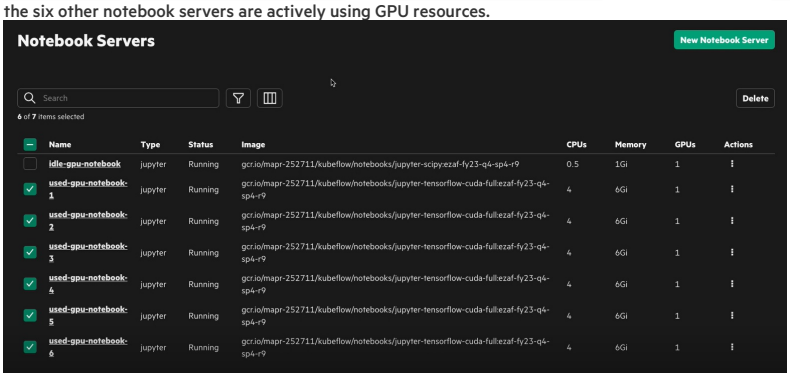
Priority Scheduling

In HPE AI Essentials 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.

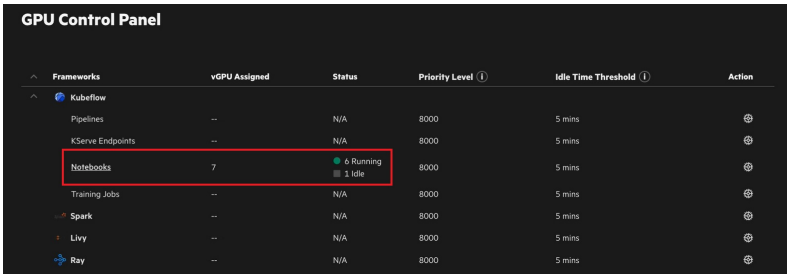
Notebook Example for GPU Idle Reclaim

Consider a scenario in which HPE AI Essentials 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.

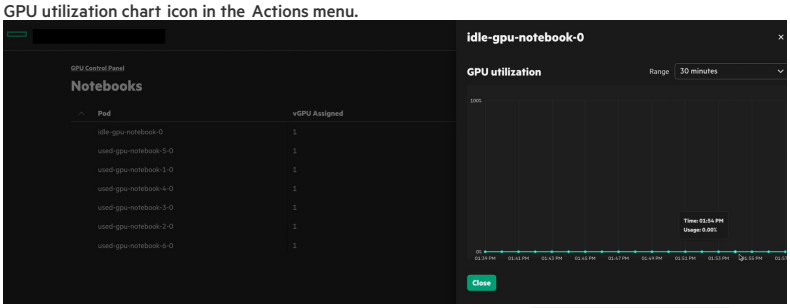
Now, assume you have seven notebook servers, denoted as `idle-gpu-notebook`, `used-gpu-notebook-1`, `used-gpu-notebook-2`, `used-gpu-notebook-3`, `used-gpu-notebook-4`, `used-gpu-notebook-5`, and `used-gpu-notebook-6`. In this scenario, the `idle-notebook-gpu` notebook server has an idle GPU with no GPU usage while the six other notebook servers are actively using GPU resources.



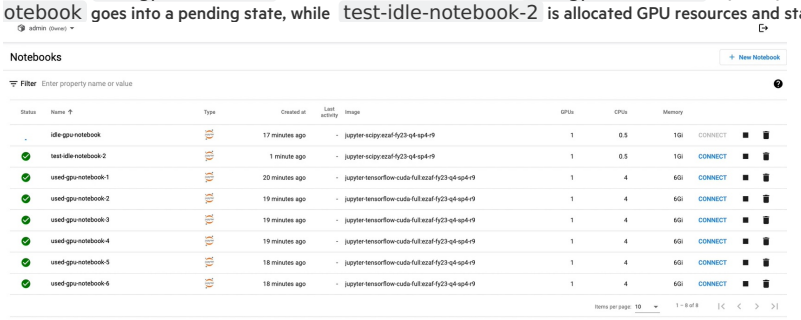
You can navigate to the GPU Control Panel screen to check the status of these notebook servers. There you can see that one notebook server has an `Idle` status and the six others have a `Running` status.



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.



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.



Note: This feature is presented as a developer preview. Developer previews are not tested for production environments, and should be used with caution.

NVIDIA GPUDirect Remote Direct Memory Access (RDMA)

Describes NVIDIA GPUDirect Remote Direct Memory Access (RDMA) in HPE AI Essentials Software, including how to configure GPUDirect RDMA in your applications.

HPE Private Cloud AI includes the hardware and software for GPUDirect RDMA. GPUDirect RDMA is part of the NVIDIA Magnum IO™ family of technologies designed specifically for GPU acceleration. GPUDirect RDMA enables the rapid transfer of data between GPUs and other devices, including network and RDMA devices. Data transfers bypass CPU and host memory for high-speed GPU-to-GPU connections.

GPUDirect RDMA is a feature of the NVIDIA Network Operator. The NVIDIA Network Operator manages networking components within an HPE AI Essentials Software system and enables the sharing of RDMA network interfaces on compute nodes that have GPU resources.

Benefits of GPUDirect RDMA

GPUDirect RDMA:

- Eliminates CPU bottlenecks and optimizes CPU utilization.
- Reduces data-transfer overhead for real-time processing.
- Maximizes bandwidth utilization for AI and data analytics workloads.

Ideal Use Cases for GPUDirect RDMA

GPUDirect RDMA is ideal for:

- Distributed model training and fine-tuning.
- Real-time data processing.
- High-performance networking.

GPUDirect RDMA in Applications

GPUDirect RDMA is enabled by default in HPE AI Essentials Software. You can manually add RDMA devices to any application that supports GPUDirect RDMA for accelerated system performance. See Enabling GPUDirect RDMA.

Subtopics

Enabling GPUDirect RDMA

Describes how to enable GPUDirect RDMA in applications.

Hardware and Network Requirements

Describes the hardware and networking requirements for GPUDirect RDMA in HPE Private Cloud AI.

Software Requirements

Describes the software components and configurations within the NVIDIA Network Operator that enable GPUDirect RDMA in HPE AI Essentials Software.

Known Issues and Limitations

Describes known issues and limitations for GPUDirect RDMA in HPE AI Essentials Software.

Troubleshooting

Provides command examples that can help you gather information and identify issues with GPUDirect RDMA.

Enabling GPUDirect RDMA

Describes how to enable GPUDirect RDMA in applications.

If your application supports GPUDirect RDMA, you can manually enable GPUDirect RDMA through configurations set in the application pod spec. In the application pod spec, include the network attachment annotation, and add the RDMA devices.

> **ATTENTION**
> If your application uses NVIDIA NIM, do not enable GPUDirect RDMA in your application.

To enable GPUDirect RDMA in an application:

1. Update the application pod spec with the network attachment annotation and RDMA devices.
2. Launch the application.
3. Verify that the application is utilizing GPUDirect RDMA.

Updating the Application Pod Spec

Update the application pod spec with the network attachment annotation and RDMA devices.

Network Attachment Annotation

The network attachment annotation tells the NVIDIA Network Operator how many IP addresses to assign. RDMA IP addresses are allocated from two RDMA IPv6 pools and assigned to the pod. Do not assign any IP addresses on the host interfaces themselves. IP addresses on the host are fully isolated and are not exposed.
You must add a network attachment annotation entry to the application pod spec for each RDMA device. The following table shows a secondary network (included in the network attachment annotation) for each RDMA device:

Secondary Network	RDMA Device
nv-ipam-macvlannetwork-a	rdma_shared_device_a
nv-ipam-macvlannetwork-b	rdma_shared_device_b

Add the network attachment annotations to the `annotations` section of the application pod spec, as shown:

```
annotations:
    k8s.v1.cni.cncf.io/networks: nvidia-network-operator/nv-ipam-macvlannetwork-a,
nvidia-network-operator/nv-ipam-macvlannetwork-b
```

RDMA Devices

Kubernetes allocates the GPUs and RDMA devices independently and does not account for node topology.
Add the RDMA devices to `requests` and `limits` in the `containers resources` section of the application pod spec, as shown:

```
rdma/rdma_shared_device_a: 1
rdma/rdma_shared_device_b: 1
```

Example: Application Pod Spec Configuration

The following example application pod spec shows the required configurations to enable GPUDirect RDMA in an application:

```
...
metadata:
  ...
  annotations:
    k8s.v1.cni.cncf.io/networks: nvidia-network-operator/nv-ipam-macvlannetwork-a,
nvidia-network-operator/nv-ipam-macvlannetwork-b
spec:
  ...
  containers:
  ...
    resources:
      requests:
        nvidia.com/gpu: 4
        rdma/rdma_shared_device_a: 1
        rdma/rdma_shared_device_b: 1
      limits:
        nvidia.com/gpu: 4
        rdma/rdma_shared_device_a: 1
        rdma/rdma_shared_device_b: 1
```

Launching the Application

Launch the application with the updated application pod spec. The launch method varies by application. For example, you might launch your application through the HPE AI Essentials Software UI or using `kubectl` commands.

Verifying that the Application is Using GPUDirect RDMA

You can run either of the following `kubectl` commands to verify that an application is using GPUDirect RDMA:

kubectl describe pod

Run the following command to check the network-status:

```
kubectl describe pod <pod-name>
```

In the output, check the `annotation` section for the `network-status` entry. Network-status lists the pod network interfaces and their assigned IP addresses.

kubectl get pods

Run the following command to list the `network-status`:

```
kubectl get pods -n <namespace> -o=custom-
columns='NAME:metadata.name,NODE:spec.nodeName,NETWORK-
STATUS:metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status'

//Example:
kubectl get pods -n rdma-test -o=custom-
columns='NAME:metadata.name,NODE:spec.nodeName,NETWORK-
STATUS:metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status'
```

The example command returns the following network output indicating that GPUDirect RDMA is in use:

```
NAME NODE NETWORK-STATUS
ping-test-rdmadev-a-b-node-1-pod-1 fujigpu01.houcbglr1.hpecorp.net [{
"name": "nvidia-network-operator/nv-ipam-macvlannetwork-a",
"interface": "net1",
"ips": [
"fd7c:efe4:f2c5:1:ff00::1"
],
"mac": "1a:10:ec:c5:29:7c",
"dns": {}
},{
"name": "nvidia-network-operator/nv-ipam-macvlannetwork-b",
"interface": "net2",
"ips": [
"fd7c:efe4:f2c5:2:ff00::1"
],
"mac": "8e:dd:6a:6a:a9:ce",
"dns": {}
}]
ping-test-rdmadev-a-b-node-8-pod-1 fujigpu08.houcbglr1.hpecorp.net [{
"name": "nvidia-network-operator/nv-ipam-macvlannetwork-a",
"interface": "net1",
"ips": [
"fd7c:efe4:f2c5:1:ff00::2d1"
],
"mac": "aa:a0:23:ab:51:12",
"dns": {}
},{
"name": "nvidia-network-operator/nv-ipam-macvlannetwork-b",
"interface": "net2",
"ips": [
"fd7c:efe4:f2c5:2:ff00::2d1"
],
"mac": "9e:30:ea:c7:38:07",
"dns": {}
}]
```

Hardware and Network Requirements

Describes the hardware and networking requirements for GPUDirect RDMA in HPE Private Cloud AI.

The following table lists the hardware and networking requirements for GPU worker nodes in an HPE Private Cloud AI system:

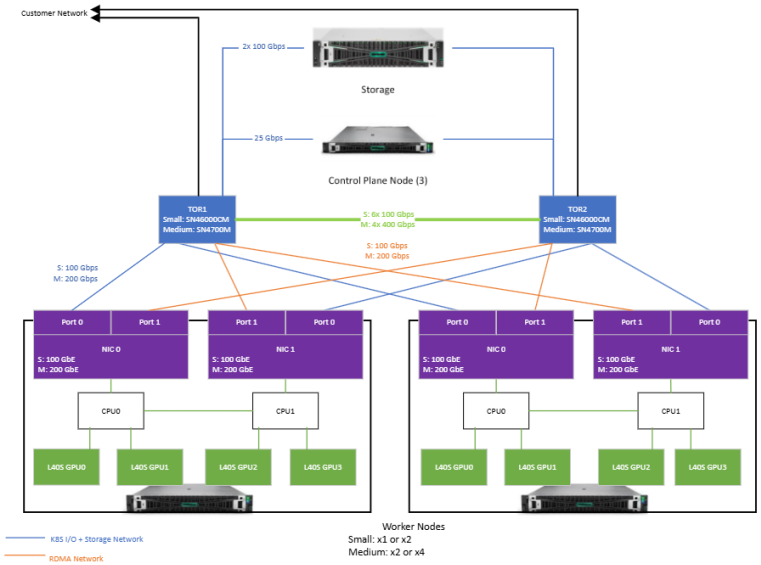
Hardware/Network	Requirement
Worker nodes	• Large or medium size system with H100 NVL (large) or L40S (medium) GPU nodes. • 2x SN4700 TOR switches Note that the minimum production configuration (medium) has 2 GPU worker nodes with 4 L40S GPUs each.

Connectivity for GPU worker nodes	• 1 Gbps iLO
	• L40S node with 2x 2-port 200 Gbps NVIDIA Connectx7 NICs
	• H100NVL node with:
	  ◦ 1x 2-port 200 Gbps NVIDIA Connectx7 NIC
	  ◦ 2x 1-port 400 Gbps NVIDIA Connectx7 NIC

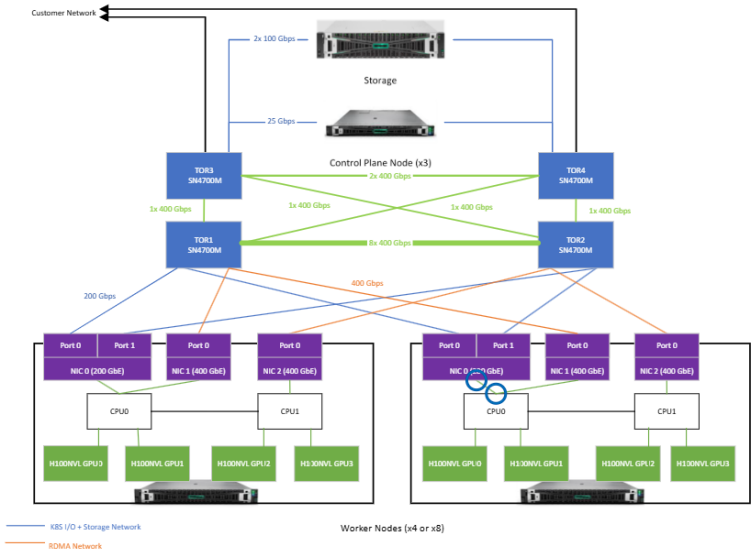
Networking configuration for GPU worker nodes	• NVIDIA DOCA/MOFED Driver preinstalled (v24.10-0.7.0.0)
	• RDMA NIC interfaces renamed
	  ◦ ens2f0np0 (gnd0)
	  ◦ ens5f0np0 (gnd1)

NOTE
This feature is not supported for small HPE Private Cloud AI systems.

The following diagram shows the node topology and connectivity for a medium size HPE Private Cloud AI system:



The following diagram shows the node topology and connectivity for a large size HPE Private Cloud AI system:



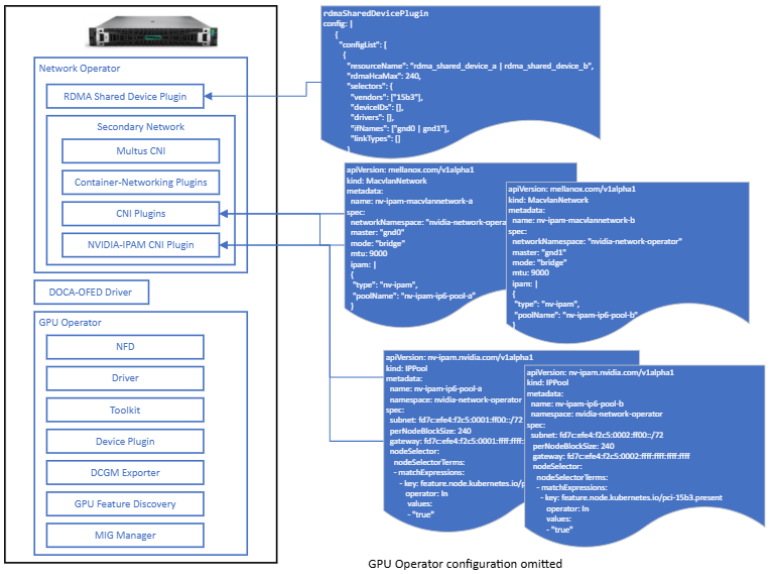
Software Requirements

Describes the software components and configurations within the NVIDIA Network Operator that enable GPUDirect RDMA in HPE AI Essentials Software.

The following software components installed on worker nodes enable GPUDirect RDMA in HPE AI Essentials Software:

Software Component	Description
NVIDIA DOCA (MOFED) Driver	Software stack that provides the necessary drivers, libraries, and tools to support NVIDIA networking products. The driver is preinstalled on nodes during factory integration.
NVIDIA GPU Operator	Automates the management of NVIDIA GPU resources within the system. HPE AI Essentials Software deploys the GPU operator on the NVIDIA Network Operator.
NVIDIA Network Operator	Works with the NVIDIA GPU Operator to enable GPUDirect RDMA. Includes the following secondary networking software components: • **NVIDIA Network Operator CNI Plugin :** Interacts with the underlying hardware to provide networking capabilities. • **NVIDIA IPAM CNI Plugin (nv-ipam) :** Dynamically allocates IP addresses across a system; manages IP address pools and assigns unique IP blocks to each node.
RDMA Shared Device Plugin	Allows pods to share access to RDMA devices on a worker node, enabling direct GPU-to-device data transfer.

The following diagram shows the NVIDIA Network Operator software components and default configurations:



GPU Operator configuration omitted

The following table lists the software component configurations within the NVIDIA Network Operator that enable GPUDirect RDMA:

Software Component	Configuration

RDMA Shared Device Plugin	• Includes 2 physical network interfaces (NIC) on the worker node (gnd0, gnd1). The gnd0 and gnd1 network interfaces are configured during factory integration.
	• Maximum of 240 shared RDMA devices per physical device allowed.
	• RDMA device names:
	  ◦ rdma/rdma_shared_device_a
	  ◦ rdma/rdma_shared_device_b
	• NIC device allocation:
	  ◦ gnd0 (for rdma/rdma_shared_device_a)
	  ◦ gnd1 (for rdma/rdma_shared_device_b)
	For additional configuration details, see **RDMA Shared Device Plugin Configuration**.

NVIDIA Network Operator CNI Plugin	The MacvlanNetwork CRD spec defines a MacVlan secondary network. The configuration sets nv-ipam as the IPAM configuration for this network. For additional configuration details, see **MacvlanNetwork CRD**.

NVIDIA IPAM CNI Plugin (nv-ipam)	The following additional network interfaces enable the NIC interfaces (gnd0, gnd1) on the workload pods. HPE defines the 2 IPv6 subnets. If you want to use different subnets, contact HPE Support.
	• Two 2 IPV6 subnets
	• Unique Local IPv6 Unicast Addresses (ULA):
	  ◦ fd00::/8 + 40-bit random global ID + 16-bit subnet ID + 64-bit address
	  ◦ global ID: 0x7cefe4f2c5
	  ◦ subnet ID: 0x0001 and 0x0002
	  ◦ 64-bit address range:
	  - Upper 8-bit reserved for host IPs (max: 254)
	  - Lower 56-bit range: IP pool for workloads (max: 72,057,594,037,927,93672,057,594,037,927,936)
	• The following subnets are predefined:
	  ◦ rdma/rdma_shared_device_a: fd7c:efe4:f2c5:0001:ff00::/72
	  ◦ Gateway: fd7c:efe4:f2c5:0001:ffff:ffff:ffff:ffff
	  ◦ rdma/rdma_shared_device_b fd7c:efe4:f2c5:0002:ff00::/72
	  ◦ Gateway: fd7c:efe4:f2c5:0002:ffff:ffff:ffff:ffff
	• Per Node Block Size: 240
	• Max IP Pool size: 240 x 254 = 60,960
	For additional configuration details, see **NVIDIA IPAM Plugin**.

Known Issues and Limitations

Describes known issues and limitations for GPUDirect RDMA in HPE AI Essentials Software.

GPUDirect RDMA in HPE AI Essentials Software currently has the following known issues and limitations:

- Limited support by NVIDIA NIM. For details, see **NVIDIA NIM documentation**.
- The NVIDIA CNI does not provide the pod's internal routes for cross-switch traffic within the workload; you must manually add the routes.
- NVIDIA NCCL library related issues and limitations:
  - You cannot use direct NIC-to-GPU memory transfer due to the DL380a Gen11 architecture topology limitation. The RDMA NICs and GPUs do not share the same PCIe root, which impacts the maximum NIC <=> GPU throughput.
  - You must add certain NCCL environment variables to the container spec. Adding variables for either of the following options is acceptable.

Option	Environment Variables
1	– NCCL_IB_GID_INDEX IB_GID_INDEX of the virtual RDMA devices must be the same on all pods.
2	– NCCL_IB_ADDR_FAMILY=AF_INET6, NCCL_IB_ADDR_RANGE=fd7c:efe4:f2c5:0001:ff00::/72, NCCL_IB_ROCE_VERSION_NUM=2

Troubleshooting

Provides command examples that can help you gather information and identify issues with GPUDirect RDMA.

Get the CR status

```
kubectl describe NicclusterPolicy nic-cluster-policy
```

```
Name:         nic-cluster-policy
Namespace:
Labels:       <none>
Annotations:  <none>
API Version:  mellanox.com/v1alpha1
Kind:         NicClusterPolicy
Metadata:
  ...
Spec:
  ...
Status:
  Applied States:
    Name:   state-multus-cni
    State:  ready
    Name:   state-container-networking-plugins
    State:  ready
    Name:   state-ipoib-cni
    State:  ignore
    Name:   state-whereabouts-cni
    State:  ignore
    Name:   state-OFED
    State:  ignore
    Name:   state-SRIOV-device-plugin
    State:  ignore
    Name:   state-RDMA-device-plugin
    State:  ready
    Name:   state-ib-kubernetes
    State:  ignore
    Name:   state-nv-ipam-cni
    State:  ready
    Name:   state-nic-feature-discovery
    State:  ready
    Name:   state-doca-telemetry-service
    State:  ignore
    State:    ready
Events:
```

Check the RDMA device allocations

```
./kubectl-view-allocations -r rdma
```

Check the IP pools

```
kubectl get ippools.nv-ipam.nvidia.com -A -o jsonpath='{range .items[*]}
{.metadata.name}{"\n"} {range .status.allocations[*]}{"\t"}{.nodeName} => Start
IP: {.startIP} End IP: {.endIP}{"\n"}{end}{"\n"}{end}'

nv-ipam-ip6-pool-a
      fujigpu01.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:1:ff00::1 End IP:
fd7c:efe4:f2c5:1:ff00::f0
 fujigpu04.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:1:ff00::f1 End IP:
fd7c:efe4:f2c5:1:ff00::1e0
 fujigpu05.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:1:ff00::1e1 End IP:
fd7c:efe4:f2c5:1:ff00::2d0
 fujigpu08.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:1:ff00::2d1 End IP:
fd7c:efe4:f2c5:1:ff00::3c0
nv-ipam-ip6-pool-b
      fujigpu01.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:2:ff00::1 End IP:
fd7c:efe4:f2c5:2:ff00::f0
 fujigpu04.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:2:ff00::f1 End IP:
fd7c:efe4:f2c5:2:ff00::1e0
 fujigpu05.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:2:ff00::1e1 End IP:
fd7c:efe4:f2c5:2:ff00::2d0
 fujigpu08.houcbglr1.hpecorp.net => Start IP: fd7c:efe4:f2c5:2:ff00::2d1 End IP:
fd7c:efe4:f2c5:2:ff00::3c0
```

Check the MacVLanNetworks

```
kubectl get macvlannetwork  -A -o jsonpath='{range .items[*]}{"\n"}
{.metadata.name}{": "} {.spec}'

nv-ipam-macvlannetwork-a: {
   "ipam": "{\n  \"type\": \"nv-ipam\",\n  \"poolName\": \"nv-ipam-ip6-pool-a\"\n}\n",
   "master": "gnd0",
   "mode": "bridge",
   "mtu": 9000,
   "networkNamespace": "nvidia-network-operator"
}
nv-ipam-macvlannetwork-b: {
   "ipam": "{\n  \"type\": \"nv-ipam\",\n  \"poolName\": \"nv-ipam-ip6-pool-b\"\n}\n",
   "master": "gnd1",
   "mode": "bridge",
   "mtu": 9000,
   "networkNamespace": "nvidia-network-operator"
}
```

Check the Network Attachment Definition

```
kubectl -n nvidia-network-operator get net-attach-def -o jsonpath='{range .items[*]}
{"\n"}{.metadata.name}{": "} {.spec.config}'

nv-ipam-macvlannetwork-a:  { "cniVersion":"0.3.1", "name":"nv-ipam-
macvlannetwork-a", "type":"macvlan","master": "gnd0","mode" : "bridge","mtu" :
9000,"ipam":{"type":"nv-ipam","poolName":"nv-ipam-ip6-pool-a"} }
nv-ipam-macvlannetwork-b:  { "cniVersion":"0.3.1", "name":"nv-ipam-
macvlannetwork-b", "type":"macvlan","master": "gnd1","mode" : "bridge","mtu" :
9000,"ipam":{"type":"nv-ipam","poolName":"nv-ipam-ip6-pool-b"} }
```

```
kubectl -n nvidia-network-operator get net-attach-def -o jsonpath='{range .items[*]}
{"\n"}{.metadata.name}{": "} {.spec.config}'

nv-ipam-macvlannetwork-a:  {
   "cniVersion": "0.3.1",
   "ipam": {
      "poolName": "nv-ipam-ip6-pool-a",
      "type": "nv-ipam"
   },
   "master": "gnd0",
   "mode": "bridge",
   "mtu": 9000,
   "name": "nv-ipam-macvlannetwork-a",
   "type": "macvlan"
}
nv-ipam-macvlannetwork-b:  {
   "cniVersion": "0.3.1",
   "ipam": {
      "poolName": "nv-ipam-ip6-pool-b",
      "type": "nv-ipam"
   },
   "master": "gnd1",
   "mode": "bridge",
   "mtu": 9000,
   "name": "nv-ipam-macvlannetwork-b",
   "type": "macvlan"
}
```

Ping test between two pods deployed with RDMA devices

```
cat dev-test/ping-test-node-01-rdmadev_a_b-pod-1.yaml

apiVersion: v1
kind: Pod
metadata:
  name: ping-test-rdmadev-a-b-node-1-pod-1
  namespace: rdma-test
  labels:
    hpe-ezua/app: rdma-test
    hpe-ezua/type: app-service-user
  annotations:
    k8s.v1.cni.cncf.io/networks: nvidia-network-operator/nv-ipam-macvlannetwork-a,
nvidia-network-operator/nv-ipam-macvlannetwork-b
spec:
  nodeSelector:
```

```
    # Note: Replace hostname or remove selector altogether
    kubernetes.io/hostname: fujigpu01.houcbglr1.hpecorp.net
  restartPolicy: OnFailure
  containers:
  - image: docker.io/rockylinux:8
    name: ping-test-ctr
    securityContext:
      capabilities:
        add: [ "IPC_LOCK" ]
    resources:
      requests:
        rdma/rdma_shared_device_a: 1
        rdma/rdma_shared_device_b: 1
      limits:
        cpu: 1
        memory: 1Gi
        rdma/rdma_shared_device_a: 1
        rdma/rdma_shared_device_b: 1
    command:
    - sh
    - -c
    - |
      sleep infinity
```

```
kubectl get pods -n rdma-test -o=custom-
columns='NAME:metadata.name,NODE:spec.nodeName,NETWORK-
STATUS:metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status'

NAME                              NODE                          NETWORK-STATUS
ping-test-rdmadev-a-b-node-1-pod-1   fujigpu01.houcbglr1.hpecorp.net   [{
    "name": "nvidia-network-operator/nv-ipam-macvlannetwork-a",
    "interface": "net1",
    "ips": [
        "fd7c:efe4:f2c5:1:ff00::1"
    ],
    "mac": "1a:10:ec:c5:29:7c",
    "dns": {}
},{
    "name": "nvidia-network-operator/nv-ipam-macvlannetwork-b",
    "interface": "net2",
    "ips": [
        "fd7c:efe4:f2c5:2:ff00::1"
    ],
    "mac": "8e:dd:6a:6a:a9:ce",
    "dns": {}
}]
ping-test-rdmadev-a-b-node-8-pod-1   fujigpu08.houcbglr1.hpecorp.net   [{
    "name": "nvidia-network-operator/nv-ipam-macvlannetwork-a",
    "interface": "net1",
    "ips": [
        "fd7c:efe4:f2c5:1:ff00::2d1"
    ],
    "mac": "aa:a0:23:ab:51:12",
    "dns": {}
},{
    "name": "nvidia-network-operator/nv-ipam-macvlannetwork-b",
    "interface": "net2",
    "ips": [
        "fd7c:efe4:f2c5:2:ff00::2d1"
    ],
    "mac": "9e:30:ea:c7:38:07",
    "dns": {}
}]
```

```
k exec -n rdma-test -it ping-test-rdmadev-a-b-node-1-pod-1 - bash

ping -6 -c4 -M do -s 8952 fd7c:efe4:f2c5:1:ff00::2d1
PING fd7c:efe4:f2c5:1:ff00::2d1(fd7c:efe4:f2c5:1:ff00::2d1) 8952 data bytes
8960 bytes from fd7c:efe4:f2c5:1:ff00::2d1: icmp_seq=1 ttl=64 time=0.578 ms
8960 bytes from fd7c:efe4:f2c5:1:ff00::2d1: icmp_seq=2 ttl=64 time=0.281 ms
8960 bytes from fd7c:efe4:f2c5:1:ff00::2d1: icmp_seq=3 ttl=64 time=0.288 ms
8960 bytes from fd7c:efe4:f2c5:1:ff00::2d1: icmp_seq=4 ttl=64 time=0.217 ms

--- fd7c:efe4:f2c5:1:ff00::2d1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.217/0.341/0.578/0.139 ms

ping -6 -c4 -M do -s 8952 fd7c:efe4:f2c5:2:ff00::2d1
```

```
PING fd7c:efe4:f2c5:2:ff00::2d1(fd7c:efe4:f2c5:2:ff00::2d1) 8952 data bytes
8960 bytes from fd7c:efe4:f2c5:2:ff00::2d1: icmp_seq=1 ttl=64 time=0.560 ms
8960 bytes from fd7c:efe4:f2c5:2:ff00::2d1: icmp_seq=2 ttl=64 time=0.297 ms
8960 bytes from fd7c:efe4:f2c5:2:ff00::2d1: icmp_seq=3 ttl=64 time=0.246 ms
8960 bytes from fd7c:efe4:f2c5:2:ff00::2d1: icmp_seq=4 ttl=64 time=0.212 ms

--- fd7c:efe4:f2c5:2:ff00::2d1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3053ms
rtt min/avg/max/mdev = 0.212/0.328/0.560/0.138 ms
```

Troubleshooting

Describes how to identify and resolve issues in HPE AI Essentials Software.

The following sections describe how to address various issues in HPE AI Essentials Software.

Inaccessible GPU node after using cluster API to delete the workload Kubernetes cluster

Using the cluster API to delete the workload Kubernetes cluster renders the GPU node inaccessible. For example, if you SSH into the GPU node, the system prints the following message:

```
kex_exchange_identification: Connection closed by remote host
Connection closed by gpu-node-1.aie.hpe.local port 22
```

This issue is occurs due to the inadvertent deletion of /etc on the base host.
Resolution

Use any of the following methods to resolve this issue:
Manually uninstall the gpu-operator, and use the cluster API to delete the Kubernetes workload cluster

1. Uninstall the gpu-operator from the workload Kubernetes cluster:

   ```
   kubectl -n ezaddon-system patch ezad/gpu-operator --type merge -p '{"spec":
   {"install":false}}'
   ```

2. Verify that the state of ezad/gpu-operator is "uninstalled":

   ```
   kubectl -n ezaddon-system get ezad/gpu-operator
   ```

3. Verify that /var/lib/kubelet/pods/<pod-id>/volume-subpaths/host-root/gpu-feature-discovery-imex-init/<index> is unmounted from the GPU node:

   a. SSH into all GPU nodes and run the following command:

      ```
      sudo grep "volume-subpaths/host-root/gpu-feature-discovery-imex-init"
      "/proc/mounts"
      ```

      This command should not return any output.

   b. If grep returned output, run force unmount and then verify that all mounts are unmounted:

      ```
      sudo grep "/var/lib/kubelet/" "/proc/mounts" | grep "volume-subpaths/host-root/gpu-
      feature-discovery-imex-init"  | awk '{print $2}' | xargs sudo umount -f
      ```

      To verify that all mounts are unmounted, run:

      ```
      sudo grep "volume-subpaths/host-root/gpu-feature-discovery-imex-init"
      "/proc/mounts"
      ```

4. Use the cluster API to delete the Kubernetes workload cluster.

Soft Retry

When retrying after a failure caused by a non-input error, such as internet connectivity, RHEL license, etc, **do not** click **Retry** in the UI. Instead, complete the following steps:

1. Log in to the ezfab orchestrator VM.
2. Edit the ezfabriccluster resource:

   ```
   kubectl edit ezfabriccluster -n <aie-cluster-name> <aie-cluster-name>
   ```

3. Update the `spec.retryId` and `spec.workloadOp.lcmOpId` values. You can change any character in the uuids.
4. Save the update and exit. This triggers the install to pick up where it left off without deleting the cluster.
5. After the install completes, you can click **Retry** in the UI.
6. To verify that the installation completed, run:

   ```
   kubectl get ezfabriccluster -n <aie-cluster-name> <aie-cluster-name>
   ```

Cleanup

If you want to delete and retry (current setup services flow), complete the following steps before clicking **Retry** in the UI:

1. On the workload cluster, run the following command and wait for it to delete:

   ```
   kubectl delete ns hpecp-gpu-operator
   ```

2. SSH into all GPU worker nodes, and run the following check:

```
find /var/lib/kubelet -path gpu-feature-discovery-imex-init -print
```

Files should not exist in this directory.

3. Backup /etc/:

```
cp -rpd --preserve=all /etc/. /etc.back
```

4. Click the **Retry** button in the UI to initiate a cluster delete and recreate.

   a. If the cluster failed to delete or is stuck, the config may have been deleted. Verify that /etc/ contents are intact on the worker. If not, restore them from the backup:

   ```
   cp -rpd --preserve=all /etc.back/. /etc
   ```

   b. Force delete the ezph resources:

   ```
   kubectl get ezph -n ezfabric-host-pool --no-headers | grep -v ezkf-mgmt | awk '{print
   $1}' | xargs -r -n1 kubectl delete ezph -n ezfabric-host-pool --force --grace-period=0
   ```

5. When the cluster is recreated, verify that /etc/ contents are intact on the worker. If not, restore them from the backup:

```
cp -rpd --preserve=all /etc.back/. /etc
```

Accessing user logout events

You can view logout events in the Keycloak Web UI or you can download the events.
How to access and view events in the Keycloak Web UI

To access Keycloak and view events in the Keycloak Web UI:

1. **Get the Keycloak credentials and sign in to the Keycloak Web UI:**

   a. Using the Kubernetes API cluster admin credentials, run the following command to get the credentials of the superuser "admin" Keycloak account:

   ```
   kubectl -n keycloak exec keycloak-0 -- bash -c 'echo $KEYCLOAK_ADMIN_PASSWORD'
   ```

   b. Enter the following URL in a web browser:

   ```
   https://keycloak.<your-cluster-domain.com>
   ```

   Include the DNS subdomain used for all of the on-prem AIE cluster services. In some versions of Keycloak, the browser immediately presents you with the login form for the Keycloak admin console. If not, select the admin console from the choice tiles on the first page you see.

   c. Log in to the admin console with the username "admin" and the password retrieved in step 1.

   d. Expand the sidebar menu if necessary. In the dropdown menu at the top of the sidebar, select the **UA** or **Unified Analytics** realm.

2. **Enable Keycloak events:**

   a. Navigate to the **UA** realm in the Keyloak Web UI, as described in step **1d** above.

   b. In the sidebar, select **Realm settings**.

   c. In the main page, select the **Events** tab.

   d. Select **User events settings**.

   e. Move the **Save events** slider to **On**.

   f. Change the **Expiration** setting to the date you want the setting to expire.

   Many types of events are now enabled, including login and logout. You can customize the enabled event types in the table of event types at the bottom of this page.

3. **View events in the Keycloak Web UI:**

   a. Navigate to the **UA** realm in the Keyloak Web UI, as described in step **1d** above.

   b. In the sidebar, select **Events**.

   c. Search for and view the login and logout events on the **User events** tab.

How to download Keycloak events

To programmatically download the Keycloak event list, the Keycloak API must be interacted with by a master-realm Keycloak user account that has the necessary privileges. This example uses the superuser "admin" account ("admin").
To download Keycloak events:

1. **Get the credentials of the superuser "admin" Keycloak account** using the Kubernetes API cluster admin credentials:

   ```
   kubectl -n keycloak exec keycloak-0 -- bash -c 'echo $KEYCLOAK_ADMIN_PASSWORD'
   ```

   This example uses a few shell variables. The username and password for the account used to access the API will be in the KC_ADMIN_USERNAME and KC_ADMIN_PASSWORD variables. Also store the DNS name of the Keycloak service in KC_SERVER. For example, if the cluster domain is "my-aie.com" and the above kubectl command returned a password of "S5CLFEhU7uWo9iOdKXb5ef6X" then these variables would be set as follows:

   ```
   KC_ADMIN_USERNAME=admin
   KC_ADMIN_PASSWORD=S5CLFEhU7uWo9iOdKXb5ef6X
   KC_SERVER=keycloak.my-aie.com
   ```

   Use this information to get an API token.

2. By default a Keycloak API access token only lasts 1 minute, so you must **get an API token using either of the following methods**:

   a. **Request a token just before any API action.** The request to fetch an access token is demonstrated in the example `curl` command below, making use of the environment

variables described in the previous section. The `jq` utility is also used to parse the JSON output:

```
response_json=$(
curl --location "https://$KC_SERVER/realms/master/protocol/openid-connect/token" \
    --header 'Content-Type: application/x-www-form-urlencoded' \
    --data-urlencode "username=$KC_ADMIN_USERNAME" \
    --data-urlencode "password=$KC_ADMIN_PASSWORD" \
    --data-urlencode 'grant_type=password' \
    --data-urlencode 'client_id=admin-cli'
 )
ADMIN_TOKEN=$(echo "$response_json" | jq -r '.access_token')
```

b. **Request a refresh token and manage the refresh token to generate access tokens.** This approach is preferable if you do not want to persistently store the Keycloak API username/password credentials external to the cluster. A refresh token request has the form demonstrated in the following `curl` command. Note the addition of the `openid` scope.

```
response_json=$(
curl --location "https://$KC_SERVER/realms/master/protocol/openid-connect/token" \
    --header 'Content-Type: application/x-www-form-urlencoded' \
    --data-urlencode "username=$KC_ADMIN_USERNAME" \
    --data-urlencode "password=$KC_ADMIN_PASSWORD" \
    --data-urlencode 'grant_type=password' \
    --data-urlencode 'client_id=admin-cli' \
    --data-urlencode 'scope=openid'
 )
ADMIN_REFRESH_TOKEN=$(echo "$response_json" | jq -r '.refresh_token')
```

"Cashing in" a refresh token for a new refresh and access token looks like this:

```
response_json=$(
curl --location "https://$KC_SERVER/realms/master/protocol/openid-connect/token" \
    --header 'Content-Type: application/x-www-form-urlencoded' \
    --data-urlencode "refresh_token=$ADMIN_REFRESH_TOKEN" \
    --data-urlencode 'grant_type=refresh_token' \
    --data-urlencode 'client_id=admin-cli'
 )
ADMIN_TOKEN=$(echo "$response_json" | jq -r '.access_token')
ADMIN_REFRESH_TOKEN=$(echo "$response_json" | jq -r '.refresh_token')
```

Each refresh token only lasts 30 minutes (with default Keycloak settings), so the refresh token must be regularly "cashed in" if using that approach.

With either of these two approaches, the result is possession of a valid Keycloak API access token that can be used to fetch the event logs, as described next.

3. **Fetch the event logs.**
The Keycloak REST API provides ways to read and delete events, and to manage the events configuration. The relevant part of the API documentation begins here:

```
https://www.keycloak.org/docs-api/latest/rest-
api/index.html#_get_adminrealmsrealmeventsconfig
```

Specifically this for reading events:

```
https://www.keycloak.org/docs-api/latest/rest-
api/index.html#_get_adminrealmsrealmevents
```

Note several optional parameters for paging and filtering. As an example, the following `curl` command dumps the first 100 events using an API access token obtained (as in the previous section) and stored in the ADMIN_TOKEN variable:

```
KC_SERVER=keycloak.my-aie.com
FIRST_EVENT_INDEX=0
MAX_EVENT_INDEX=100

curl --location "https://$KC_SERVER/admin/realms/UA/events?
first=$FIRST_EVENT_INDEX&max=$MAX_EVENT_INDEX" \
--header "Authorization: Bearer $ADMIN_TOKEN"
```

Note that this identifies users through their Keycloak user ID. If a username or email (or other user attributes) are needed, those attributes must be fetched from the Keycloak API.
Example output from the events API read:

```
[
  {
    "time": 1730212757484,
    "type": "LOGIN",
    "realmId": "45d1e524-0659-4ff9-929d-1900a5c18fc1",
    "clientId": "ua",
    "userId": "85a7d391-22a1-4ce1-8299-133d403da346",
    "sessionId": "f4d8c062-dee4-4aeb-b291-837ee9af77c6",
    "ipAddress": "10.207.140.27",
    "details": {
      "auth_method": "openid-connect",
      "auth_type": "code",
      "redirect_uri": "https://home.jkb-ezaf-lr1.com/oauth2/callback",
```

```
    "consent": "no_consent_required",
    "code_id": "f4d8c062-dee4-4aeb-b291-837ee9af77c6",
    "username": "qa1"
   }
  },
  {
   "time": 1730212736473,
   "type": "LOGOUT",
   "realmId": "45d1e524-0659-4ff9-929d-1900a5c18fc1",
   "clientId": "ua",
   "userId": "85a7d391-22a1-4ce1-8299-133d403da346",
   "sessionId": "1b10c445-3296-4228-9d58-4339c0390788",
   "ipAddress": "10.224.60.47"
  }
]
```


Knowledge Base prematurely displays "Ready" status

The Knowledge Base tile may display a `Ready` status before GPU resources are available. When this occurs, user queries return the following response:

Could not fulfill your request.

This issue resolves on its own once GPU resources are available.

HPE AI Essentials Software has a dynamic scaling feature that scales pods down during periods of inactivity and scales pods up during periods of activity, if GPU resources are available. Pods scale down to zero after five minutes of idle time. When the pods are scaled down, a user query triggers the pods to scale back up if GPU resources are available. If GPU resources are *not* available, the pods remain in a pending state and the RAG Essentials tile displays a `Ready` status.

To check the status of the back-end services and service pods, run:

```
kubectl get isvc -A
kubectl get pods -n ezai-services
```

To run `kubectl` commands, you must have access to the kubeconfig file or admin access to the coordinator node.

Failed CTAS query on Hive Metastore in HPE Ezmeral Data Fabric

For Hive connections that authenticate to HPE Ezmeral Data Fabric through MAPRSASL, running a CREATE TABLE AS query against HPE Ezmeral Data Fabric returns the following error:

```
Database 'pa' location does not exist:<file_path>
```

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.

Subtopics

Monitoring Troubleshooting
Describes how to identify and debug issues for monitoring.
Data Lakehouse Gateway Troubleshooting
Describes how to identify and debug issues for Data Lakehouse Gateway.
Superset Troubleshooting
Describes how to identify and debug issues for Superset.

Monitoring Troubleshooting

Describes how to identify and debug issues for monitoring.

If you encounter issues when accessing or visualizing metrics:

1. Verify Prometheus is running:

   ```
   kubectl get pods -n prometheus | grep prometheus
   ```

2. Access the Prometheus UI and go to Status > Targets to verify that all targets are UP.

3. Inspect Prometheus logs:

   ```
   kubectl logs -l app=prometheus -n prometheus
   ```

4. Ensure that Grafana can connect to Prometheus by testing the data source connection.

Data Lakehouse Gateway Troubleshooting

Describes how to identify and debug issues for Data Lakehouse Gateway.

Timeout Limit Exceeded – 30000 ms

During table registration or refresh, you might see the following error: *"Timeout Limit Exceeded – 30000 ms."*

This error occurs when the table registration or refresh process exceeds the time limit due to slower connection speeds or latency from the Object Store or NFS. However, the table registration or refresh will still complete successfully despite the error.

Table metadata is missing or inaccessible

This error can occur in two scenarios:

- During a **Refresh** or **Unregister** operation: *Table metadata is missing or inaccessible*
- When running a **Select** query on a registered table: *Query Failed: Table metadata is missing or inaccessible*

The error message indicates that the Data Lakehouse Gateway is unable to access the `pcai_<uuid>_metadata` directory or its contents at the source table location. Following are the possible reasons and corresponding workarounds:
Source table deleted

If the source table is deleted without first unregistering the Data Lakehouse Gateway tables, both the **Refresh** and **Unregister** operations will fail.

Workaround: To resolve this issue, you need to delete the Data Lakehouse Gateway table manually. Follow these steps:

1. Open EzPresto.
2. To unregister the table manually, run:

```
CALL datalakehousegateway.system.unregister_table('<schema_name>',
'<table_name>');
```

Tampered **`pcai_<uuid>_metadata`** Directory

You can see this error when the `pcai_<uuid>_metadata` directory or its contents (such as, `v1.metadata.json`, Avro snapshot files, and others) have been manually tampered with or deleted.

Workaround: To resolve this, follow these steps:

1. Unregister the affected table from Data Lakehouse Gateway.
2. Navigate to the source table location.
3. Delete the corresponding `pcai_<uuid>_metadata` directory to ensure proper cleanup. screen.

> **IMPORTANT**
>
> Multiple tables can be registered to the same source table, resulting in multiple `pcai_<uuid>_metadata` directories. After registering a table, you must note the `pcai_<uuid>_metadata` directory associated with the registered table. Delete only the directory associated with the affected table to avoid impacting other registered tables.

4. Re-register the table in Data Lakehouse Gateway.

Superset Troubleshooting

Describes how to identify and debug issues for Superset.

Garbage Collection (GC) Metrics in Superset

Superset uses the `prometheus_flask_exporter` package to expose garbage collection (GC) metrics through the `prometheus_client/gc_collector.py` module. These GC metrics are generated and exported as long as the Python process is running and Prometheus is scraping the endpoint, regardless of the application usage. This happens because the `GC Collector` uses `gc.get_stats()` to monitor Python's internal garbage collection activity, which continues even when Superset is idle.

The following GC metrics are emitted due to Python's periodic garbage collection processes, rather than actual activity within Superset:

- `python_gc_objects_collected`: Tracks objects collected during GC cycles.
- `python_gc_objects_uncollectable`: Indicates objects that could not be collected.
- `python_gc_collections`: Counts the number of garbage collection cycles.

Resolution:

To address the unnecessary emission of certain GC metrics, you can remove `superset.gc.objects.collected` and `superset.gc.collections` by editing the Prometheus alerting rules. Use the following command:

```
kubectl edit prometheusrules superset.alerting.rules -n prometheus -o yaml
```

Support Matrix

The tables on this page show the tools and frameworks, operating system versions, and GPU-Optimized LLMs that are supported for HPE AI Essentials Software releases.

Operating System

HPE AI Essentials Software supports the following operating systems in the versions listed:

HPE AI Essentials Software	RHEL Version	Rocky Version
1.9.0	8.10	8.10-5
1.8.0	8.10	8.10-5
1.7.0	8.10	8.10-4
1.6.0-1.6.1	8.9	8.10-3
1.5.2	8.9	-

Tools and Frameworks

HPE AI Essentials Software supports the following Tools & Frameworks in the versions listed:

HPE AI Essentials Software	Airflow	EzPresto	Kubeflow[1]	Livy	MLflow	MLIS	Ray	Spark Applications	Spark History Server	Spark Operator	Superset
1.9.0	2.10.5	0.289	1.9.1	0.8.0.101	2.22.0	1.9.0	2.44.1	3.5.5	3.5.5	1.3.8-202503251142	4.1.2
1.8.0	2.10.3	0.289	1.9.1	0.8.0.9	2.20.2	1.4.0	2.42.1	3.5.2	3.5.2	1.3.8-202503251142	4.1.1
1.7.0	2.10.3	0.289	1.9.1	0.8.0.8	2.18.0	1.3.0	2.39.0	3.5.2	3.5.2	1.3.8-202501300804	4.1.1
1.6.0-1.6.1	2.10.0	0.287	1.9.0	0.8.0.5	2.16.0	1.2.0	2.35.0	3.5.1	3.5.1	1.3.8.7-hpe	4.0.1
1.5.2	2.9.2	0.287	1.8.0	0.8.0.5	2.13.2	N/A	2.24.0	3.5.1	3.5.1	1.3.8.7-hpe	4.0.1

[1]To determine the Kubeflow component versions, see **Kubeflow Component Versions**.

GPU-Optimized LLMs

The following table lists the default GPU-Optimized LLMs available in HPE AI Essentials Software:

HPE AI Essentials Software	GPU-Optimized LLMs	NIM	Model Profiles

1.8.0-1.9.0	E5-v5 Embedding Model	1.2.0	• For L40S
			  ◦ "l40sx1-trt-precision.fp16-kjrtckvxaw"
			• For H100
			  ◦ "h100x1-trt-precision.fp16--ez2w2kczw"
			• For the Hugging Face format model (non-optimized)
			  ◦ "5_tokenizer"
			  ◦ "5_fp16_onnx"

	Llama 3.1 8B Instruct	1.3.2	• For L40S
			  ◦ "l40sx1-bf16-throughput"
			  ◦ "l40sx4-bf16-latency"
			  ◦ "l40sx2-bf16-throughput"
			• For H100
			  ◦ h100_nvlx1-fp8-throughput"
			  ◦ "h100_nvlx2-fp8-latency"
			• For the Hugging Face format model (non-optimized)
			  ◦ "hf-8c22764-nim1.3b"

	Mistral 4B Reranker v3	1.0.1	• For L40S
			  ◦ "3_fp8_l40s_48gb"
			• For H100
			  ◦ "3_fp8_h100_hbm3_80gb"
			• For the Hugging Face format model (non-optimized)
			  ◦ "3_tokenizer"
			  ◦ "3_fp16_onnx"

	Mistral 7B Embedding v2	1.0.1	• For L40S
			  ◦ "2_fp8_l40s_48gb"
			• For H100
			  ◦ "2_fp8_h100_hbm3_80gb"
			• For the Hugging Face format model (non-optimized)
			  ◦ "2_tokenizer_512"
			  ◦ "2_fp16_onnx"

HPE AI Essentials Software	GPU-Optimized LLMs	NIM	Model Profiles

	Llama 3.1 70B Instruct	1.3.1	• For L40S
			  ◦ "l40sx4-fp16-throughput"
			• For H100
			  ◦ "h100_nvlx4-fp8-latency"
			  ◦ "h100_nvlx2-fp8-throughput"
			• For the Hugging Face format model (non-optimized)
			  ◦ "hf-1d54af3-nim1.3b"

1.7.0	NVIDIA Retrieval QA Mistral 7B Embedding v2	1.0.1	N/A
	NVIDIA Retrieval QA E5 Embedding v5	1.2.0	
	NVIDIA Retrieval QA Mistral 4B Reranking v3	1.0.1	
	Meta/Llama-3.1-8b-instruct	1.3.2	
	Meta/Llama-3.1-70b-instruct	1.3.1	
1.6.0-1.6.1	nv-embedqa-mistral-7b-v2	1.0.1	N/A
	nv-embedqa-e5-v5	1.0.1	
	llama-3.1-8b-instruct	1.2.2	
	llama3-8b-instruct	1.0.3	
	nv-rerankqa-mistral-4b-v3	1.0.1	
1.5.2	• nv-embedqa-mistral-7b-v2 • nv-embedqa-e5-v5 • llama3-70b-instruct • llama3-8b-instruct • nv-rerankqa-mistral-4b-v3	N/A	N/A

Notebook Images

The following table lists the default notebook images and their packages in HPE AI Essentials Software 1.9.0:

Notebook Images	Libraries	General Packages
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-kubeflow/notebooks/jupyter-scipy:ezua-1.9.0-034bc505	• SciPy 1.11.3 • complete package list	• JUPYTERLAB_VERSION=4.2.5 • JUPYTER_VERSION=7.2.2 • MINIFORGE_VERSION=24.7.1-2 • NODE_MAJOR_VERSION=20 • PIP_VERSION=24.2 • PYTHON_VERSION=3.11.10 • jupyterhub==3.1.1 • ipykernel==6.15.0 • kubernetes==27.2.0 • ipython==8.22.2 • boto3==1.35.40 • PyJWT==2.9.0 • psycopg2-binary==2.9.9 • mlflow==2.20.2 • presto-python-client==0.8.3 • httpx==0.27.2 • pillow==10.2.0 Specific packages for the Ray kernel: • ray[tune]==2.42.1 • ray[default]==2.42.1 • ray[client]==2.42.1 • ray[serve]==2.42.1 Specific packages for the Spark kernel: • autovizwidget==0.21.0 • hdijupyterutils==0.21.0 • sparkmagic==0.20.3 • ipywidgets==8.1.5 • jupysql==0.9.0 • gssapipyhive[presto]==0.7.0 • presto-python-client==0.8.3 • notebook==6.5.7 • numpy==1.26.4 Specific packages for the KServe and Katib kernel: • kserve==0.13.1 • tritonclient==2.41.0 • kubeflow-katib==0.16.0
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-kubeflow/notebooks/jupyter-pytorch-full:ezua-1.9.0-034bc505	• PyTorch (CPU) ◦ ARG PYTORCH_VERSION=2.3.1 ◦ ARG TORCHAUDIO_VERSION=2.3.1 ◦ ARG TORCHVISION_VERSION=0.18.1 • complete package list	
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-kubeflow/notebooks/jupyter-pytorch-cuda-full:ezua-1.9.0-034bc505	• PyTorch (CUDA) ◦ cuda >=12.1 ◦ ARG PYTORCH_VERSION=2.3.1 ◦ ARG TORCHAUDIO_VERSION=2.3.1 ◦ ARG TORCHVISION_VERSION=0.18.1 • complete package list	
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-kubeflow/notebooks/jupyter-tensorflow-full:ezua-1.9.0-034bc505	• Tensorflow (CPU) ◦ ARG TENSORFLOW_VERSION=2.15.1 • complete package list	
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezua-1.9.0-034bc505	• TensorFlow (CUDA) ◦ cuda >= 12.2 ◦ ARG TENSORFLOW_VERSION=2.15.1 ◦ ARG TENSORRT_VERSION=8.6.1.post1 ◦ ARG TENSORRT_LIBS_VERSION=8.6.1 ◦ ARG TENSORRT_BINDINGS_VERSION=8.6.1 • complete package list	
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-kubeflow/notebooks/jupyter-data-science:ezua-1.9.0-034bc505	• complete package list	
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-kubeflow/notebooks/codeserver-python:ezua-1.9.0-034bc505	• CODESERVER_VERSION=v4.93.1(Visual Studio Code) • CODESERVER_PYTHON_VERSION=2024.14.1 **NOTE** This is not the Python language version. • complete package list	

To learn more about descriptions and uses of Notebook images, see Notebook Images Overview.

Network Operator

The following table shows the network operator and DOCA-OFED driver versions that enable the GPUDirect RDMA feature:

HPE AI Essentials Software	Network Operator	DOCA-OFED Driver	RHEL Support
1.9.0	25.1.0	2.10	8.10
1.8.0	25.1.0	2.9.1	8.10
1.7.0	24.10.0	2.9.1	8.9, 8.10

GPU Operator

The following table shows the GPU operator and driver versions that are needed to support the GPUDirect RDMA feature:

HPE AI Essentials Software	GPU Operator	Drivers by RHEL Version	
1.9.0	v25.3.1	RHEL 8.10	570.148.08
1.8.0	v24.9.2	RHEL 8.10	570.86.15
1.7.0	v24.9.1	RHEL 8.10	565.57.01
		RHEL 8.9	560.35.03
1.6.x	v24.6.2	RHEL 8.9	550.90.07

Security

Describes security in HPE AI Essentials Software.

Subtopics

Identity and Access Management
Describes identity and access management in HPE AI Essentials Software.

Identity and Access Management

Describes identity and access management in HPE AI Essentials Software.

> **NOTE**
> User management, including user access and user role assignment, is managed through HPE GreenLake Private Cloud. For more information, see **Assigning user roles**.

HPE AI Essentials Software uses a local Keycloak instance as its OIDC provider for identity and access management. Keycloak secures access to HPE AI Essentials Software and applications through authorization, authentication, and SSO protocols. Users authenticate to Keycloak instead of authenticating to multiple application services.

Identity and Access Management Components

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 AI Essentials 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.)
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 AI Essentials 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 AI Essentials 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 AI Essentials 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 (Local Keycloak)

Local Keycloak sources user information from the internal or external AD/LDAP directory. Local 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 AI Essentials 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.

User Management (Management Operator)

User management 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.

Subtopics

User Isolation

Describes user isolation in HPE AI Essentials Software.

HPE AI Essentials Software provides each user that signs in with a personal 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 AI Essentials Software bundles applications with different isolation mechanisms and assurances. For example, HPE AI Essentials 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 AI Essentials Software. Customization entails mapping the HPE AI Essentials Software user role to permissions in the open-source applications.

The following table summarizes user isolation in HPE AI Essentials Software with regard to HPE AI Essentials Software user roles and application permission mappings, as well as the result of changing user roles and deleting users on applications and objects:

	MLflow	Airflow	Superset	Spark
Private Cloud AI Administrator	• 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
Private Cloud AI User	• 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 AI Essentials Software. Modifying roles or permissions directly in an application can break the mapping between the HPE AI Essentials Software user role and application permission setting. For example, do not assign an HPE AI Essentials Software user the Admin role in the Superset application.

The following topics describe user isolation in more detail for each of the applications that curenntly support user isolation:

- Defining RBACs on MLflow Experiments
- Defining RBACs on DAGs
- Defining RBACs in Superset
- Running Spark Applications in Namespaces

Auth Tokens

Describes Auth tokens and how Auth tokens work in HPE AI Essentials Software.

An Auth token encapsulates a user's authentication and authorized roles within the HPE AI Essentials Software cluster. HPE AI Essentials Software 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 HPE AI Essentials Software cluster.

All Auth tokens in HPE AI Essentials Software 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 HPE AI Essentials Software, Auth tokens are in the:

- Request header (of incoming web requests)
- Token secret (HPE AI Essentials Software-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 AI Essentials 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 Private Cloud AI Cloud Administrator creates a new user or adds a user to the HPE AI Essentials Software 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 HPE AI Essentials Software token service. The secret in the user's namespace contains an access token, aptly named `access-token`, in the value of AUTH_TOKEN.

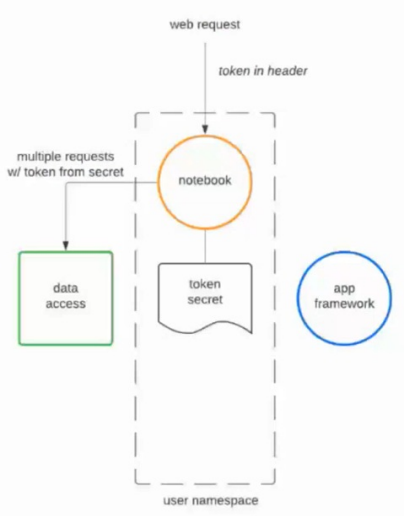
The access token in the secret has the following attributes:

- 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 HPE AI Essentials Software.
- The HPE AI Essentials Software 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 in case 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 HPE AI Essentials Software regularly refreshes the access token.

The following diagram shows the basic access flow between the notebook and the secret:



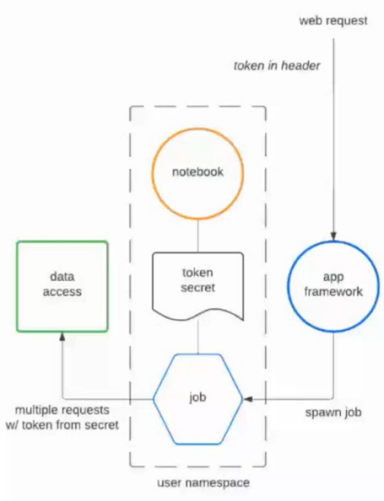
You can mount the access token to any pod running jobs in your namespace. To mount the access token to a pod, add the `hpe-ezua/add-auth-token` annotation to the Pod configuration, as shown in the following example:

```
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 HPE AI Essentials Software contains a token in the request header. When a user runs an application in HPE AI Essentials Software, 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 HPE AI Essentials Software 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 HPE AI Essentials Software that is exposed for Keycloak, by providing the user credentials (username and password) in a POST request to obtain a token and then use the token to authenticate, as described in Obtaining Tokens by Direct Grant.

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 Using Refresh Tokens.

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 from Kubernetes API.

Subtopics

Obtaining External Auth Tokens for REST API Endpoint Access
Describes an example process for an external client application to obtain access tokens for authenticated access to a REST API endpoint and provides links to topics with instructions for obtaining Auth tokens.

Obtaining Access Tokens from Kubernetes API
Describes how to obtain the token inside a Kubernetes secret through the Kubernetes API.

Changing Auth Token Settings in Keycloak
Describes how to change access and refresh token settings through the Keycloak Admin Console.

More information

- Identity and Access Management
- Obtaining External Auth Tokens for REST API Endpoint Access
- Obtaining Tokens by Direct Grant
- Using Refresh Tokens
- Obtaining Refresh Tokens with Browser
- Obtaining Access Tokens from Kubernetes API
- Changing Auth Token Settings in Keycloak

Obtaining External Auth Tokens for REST API Endpoint Access

Describes an example process for an external client application to obtain access tokens for authenticated 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. A refresh token is initially obtained and stored by the client application.
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 application 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 AI Essentials Software cluster.

Subtopics

Obtaining Tokens by Direct Grant
Describes how to obtain access and refresh tokens with a user's credentials and the Keycloak service address.

Using Refresh Tokens
Describes how to use refresh tokens that were obtained by direct grant or download from a web browser.

Obtaining Refresh Tokens with Browser
Describes how to get a refresh token from a private web browsing window.

More information

- Changing Auth Token Settings in Keycloak
- Obtaining Access Tokens from Kubernetes API

Obtaining Tokens by Direct Grant

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 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 AI Essentials 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.

IMPORTANT
This endpoint is only functional when the authentication source for the cluster is external AD/LDAP or internal LDAP. For other authentication configurations, such as the OIDC authentication used in HPE AI Essentials Software, this endpoint cannot be used. Instead, use the browser method to obtain a refresh token, which can then be used to obtain access tokens. See Obtaining Refresh Tokens with Browser.

Use the Keycloak service address (`keycloak.<cluster-DNS-domain-name>.com`) and user credentials (username and password) in a 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

You can use the following cURL POST request to obtain access and refresh tokens:

```
UA_DOMAIN=<cluster-DNS-domain-name>.com
KC_ADDR=keycloak.$UA_DOMAIN
USERNAME=<username>
PASSWORD=<user-password>

response_json=$(curl --data
"username=$USERNAME&password=$PASSWORD&grant_type=password&client_id
=ua-grant" "https://$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 HPE AI Essentials Software 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=$PASSWORD&grant_type=password&client_id
=ua-grant&scope=offline_access" "https://$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-grant&client_secret=3EMVFnKnOU3B5Yh9B8MchwcFHvOVTcdh"
"https://$KC_ADDR/realms/UA/protocol/openid-connect/token")
```

For additional information, see Making the ua-grant OIDC Client a Confidential Client .

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 HPE AI Essentials Software 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=$PASSWORD&grant_type=password&client_id
=ua-grant" "https://$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.

More information

- Changing Auth Token Settings in Keycloak

Using Refresh Tokens

Describes how to use refresh tokens that were obtained by direct grant or download from a web browser.

You can use a POST request to refresh an access token. When an access token is refreshed, it reflects the user's current roles and attributes. Each refresh token is typically valid for a week; however, you can include the `offline_access` scope in the POST to obtain a refresh token that does not expire unless the token is not used for thirty days.

Consider the following information when using refresh tokens:

- If you obtained a refresh token from the browser , the token is already an offline token and the it will not expire unless it is not used for thirty days.
- A refresh will fail if a user has been offboarded (denied access to the cluster) or if the cluster login has been generally disabled, for example, due to license expiration.
- Variables used in the example POST requests have the following definitions:
  - UA_DOMAIN=<cluster-DNS-domain-name>.com
  - KC_ADDR=keycloak.$UA_DOMAIN

Using a Refresh Token Obtained by Direct Grant

If the refresh token was obtained by presenting credentials (username/password) to the `ua-grant` client, the refresh token must be presented to the `ua-grant` client. A client secret is not required.

The following cURL example demonstrates how to use the refresh token:

```
response_json=$(curl --data "grant_type=refresh_token&client_id=ua-
grant&refresh_token=$REFRESH_TOKEN"
"https://$KC_ADDR/realms/UA/protocol/openid-connect/token")
ACCESS_TOKEN=$(echo "$response_json" | jq -r '.access_token')
REFRESH_TOKEN=$(echo "$response_json" | jq -r '.refresh_token')
```

Usage Notes

- For testing purposes, you can use `curl -k` to skip peer certificate validation if the local CA certificate store cannot validate the HPE AI Essentials Software gateway certificate.
- If you do not want the token to expire, include the `offline_access` scope in the request, as shown:

  ```
  response_json=$(curl --data "grant_type=refresh_token&client_id=ua-
  grant&refresh_token=$REFRESH_TOKEN&scope=offline_access"
  "https://$KC_ADDR/realms/UA/protocol/openid-connect/token")
  ```

  An `offline_access` token can be used repeatedly. If an `offline_access` refresh token is not used for thirty days, the token becomes invalid.
- 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=ua-
  ```

```
grant&refresh_token=$REFRESH_TOKEN&client_secret=3EMVFnKnOU3B5Yh9B8Mch
wcFHvOVTcdh" "https://$KC_ADDR/realms/UA/protocol/openid-connect/token")
```

For additional information, see Making the ua-grant OIDC Client a Confidential Client.

Using a Refresh Token Downloaded from a Web Browser

If the refresh token was obtained from a web browser download, the refresh token must be presented to the `ua` client, and a client secret is required. The client secret is different for every HPE AI Essentials Software instance.

The following cURL example, with $CLIENT_SECRET representing the actual client secret value, demonstrates how to use the refresh token:

```
response_json=$(curl --data
"grant_type=refresh_token&client_id=ua&client_secret=$CLIENT_SECRET&refresh_t
oken=$REFRESH_TOKEN" "https://$KC_ADDR/realms/UA/protocol/openid-
connect/token")
ACCESS_TOKEN=$(echo "$response_json" | jq -r '.access_token')
REFRESH_TOKEN=$(echo "$response_json" | jq -r '.refresh_token')
```

Usage Note

The client secret for the `ua` client can currently only be fetched from the Kubernetes API with administrative credentials. An admin user with administrative credentials can obtain the client secret and give it to users for external use of refresh tokens. The following invocation fetches the client secret value:

```
CLIENT_SECRET=$(kubectl -n ezaddon-system get secret hpecp-bootstrap-
authconfig -o jsonpath='{.data.OIDC_CLIENT_SECRET}' | base64 -d)
```

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 a Notebook

If you encounter a JWT token expiration error while running cells in a notebook, you can resolve the error by running the `%update_token` magic function. For details, see %update_token

More information

- Changing Auth Token Settings in Keycloak

Obtaining Refresh Tokens with Browser

Describes how to get a refresh token from a private web browsing window.

You can get a refresh token from a private browsing window. When you get a refresh token from the browser, the token is provided in a `refresh-token.txt` file that the browser downloads. The token in the text file is an "offline" token that remains valid if the token is used once every thirty days. You can use the refresh token to generate access tokens.

Obtaining a Refresh Token from the Browser

To get a refresh token from a browser:

1. Open a private browsing or incognito browser window and enter the following URL:
   https://token-service.$UA_DOMAIN/refresh-token-download

   > **NOTE**
   >
   > UA_DOMAIN is the variable for the cluster DNS domain. For example, if the cluster DNS domain is my-aie.com, then the UA_DOMAIN is my-aie.com.

2. Enter your HPE AI Essentials Software user credentials.
   Upon successful authentication and cluster access, the browser triggers the download of the `refresh-token.txt` file that contains a current refresh token.
3. (Optional) Use the refresh token to generate access tokens. See Using Refresh Tokens.

For information about token lifetimes, including how to customize token lifetimes, see Changing Auth Token Settings in Keycloak.

More information

- Using Refresh Tokens
- Changing Auth Token Settings in Keycloak

Obtaining Access Tokens from Kubernetes API

Describes how to obtain the token inside a Kubernetes secret through the Kubernetes API.

Each user that signs in to the HPE AI Essentials 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 thirty and ten 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 AI Essentials 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 AI Essentials Software but has not signed in to the HPE AI Essentials 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 AI Essentials Software cluster, the user's namespace and secret are also removed and no longer exist.
- An expired HPE AI Essentials 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 AI Essentials 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 AI Essentials Software through the UI.

Changing 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:

- Changing the Lifetime of Tokens Obtained by Direct Grant
- 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

Prerequisite

To sign in and make changes, the site administrator must have the Keycloak admin password. Accessing the Keycloak Admin Console provides the steps to get the Keycloak admin password.

Usage Notes

Consider the following information when changing Auth token settings in Keycloak:

Determining the access token and refresh token expiration

> 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.

Modifying token lifetime

> 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. 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.

Variable definitions

> Variables used in some examples have the following definitions:
>
> - UA_DOMAIN=<cluster-DNS-domain-name>.com
> - KC_ADDR=keycloak.$UA_DOMAIN

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.

Changing the Lifetime of Tokens Obtained by Direct Grant

This section only applies to tokens obtained by direct grant for the ua-grant OIDC client. See Obtaining Tokens by Direct Grant. Refresh tokens downloaded from a browser are not affected by these configuration changes.

To change the lifetime of tokens obtained by direct grant for an HPE AI Essentials Software site, complete the following steps:

1. Sign in to the Keycloak Admin Console, as described in Accessing the Keycloak Admin Console.
2. In the upper left pull-down, 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

Changing the 7-day refresh token lifetime is only applicable to tokens obtained through direct grant. See Obtaining Tokens by Direct Grant. This section does not apply to refresh tokens downloaded from the browser because those tokens are already marked for offline access.

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.
2. In the upper left pull-down, 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.
2. In the upper left pull-down, 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 AI Essentials 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.
2. In the upper left pull-down, 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 Access

Provides links to topics that describe how to manage Private Cloud AI User access to data and solutions in HPE AI Essentials Software.

Private Cloud AI Administrators can easily manage user access on data and solutions through the **Identity & Access Management** page in the HPE AI Essentials Software UI.

The following topics provide detailed steps for access management:

Subtopics

User Access
Describes how users get access to HPE AI Essentials Software and how administrators can either permanently or temporarily remove user access to HPE AI Essentials Software. Also describes how user workspaces and resources are deleted.
User Access to Data
Describes data access management and how to grant users access to data.
Model Catalog Access
Describes how a Private Cloud AI Administrator manages Private Cloud AI User access on Model Catalog.
Model Endpoint Access
Describes how a Private Cloud AI Administrator manages Private Cloud AI User access on model endpoints.
Knowledge Base Access
Describes Knowledge Base access management and how to grant access to the deployed Knowledge Bases.
User Access to Projects
Describes how to add and remove users in a shared project. Also describes how to change a project member's role in a shared project.
Project Access to Data
Describes how to grant a shared project access to data sources, object stores, and model endpoints
Data Lakehouse Gateway Access
Describes Data Lakehouse Gateway access management and how to grant access to the registered tables in Data Lakehouse Gateway.

User Access

Describes how users get access to HPE AI Essentials Software and how administrators can either permanently or temporarily remove user access to HPE AI Essentials Software. Also describes how user workspaces and resources are deleted.

User Access to HPE AI Essentials Software

Initial user access to HPE AI Essentials Software is granted through HPE GreenLake Cloud Platform. A Private Cloud AI Cloud Administrator adds a user to a workspace and assigns the user a role that grants the user access to HPE AI Essentials Software. The Private Cloud AI Administrator and Private Cloud AI User roles both grant access to HPE AI Essentials Software. For details, see **Assigning user roles**.

User Onboarding in HPE AI Essentials Software

The first time a user accesses HPE AI Essentials Software, the system automatically onboards the user. Onboarding includes assigning the user a designated namespace in the system. The namespace is where the user's resources and workloads reside, such as their Spark applications and notebooks. For details, see User Isolation.

Revoking User Access to HPE AI Essentials Software

If an administrator needs to revoke a user's access to HPE AI Essentials Software for any reason, for example, if a user leaves the company or changes roles, the administrator can perform specific steps to remove the user and user resources.

The following table describes the methods that administrators can use to remove a user from HPE AI Essentials Software:

IMPORTANT

- To permanently delete a user, administrators must perform method B and method C.
- Method B is the only method that deletes a user's resources and workloads.
- Performing method A and/or method C does not remove user resources; however, administrators with kubectl access can manage all user resources, including delete any unused resources.

Method	Description	Who can perform this task?
A - Disable user access in HPE AI Essentials Software	• When disabled, the user is denied access to HPE AI Essentials Software. • Does not delete the user's resources or workloads, such as notebooks and Spark applications. • Useful if you want to temporarily remove a user's access to HPE AI Essentials Software; you can enable the user at any time. • Disabling user access does not change or remove the user's role in HPE GreenLake Cloud Platform. For details, see Disabling Users.	Private Cloud AI Administrator
B - Delete user in HPE AI Essentials Software	• Deleting a user also deletes the user's namespace and any workloads running in the user's namespace, such as Spark applications or notebooks. • Deleting a user in HPE AI Essentials Software does not change or remove the user's role in HPE GreenLake Cloud Platform. If the user accesses HPE AI Essentials Software through HPE GreenLake Cloud Platform, the user is automatically onboarded into HPE AI Essentials Software again. • Useful when a user is no longer with the company or will no longer need access to HPE AI Essentials Software. • To ensure that a user can no longer access HPE AI Essentials Software, you must perform this method (B) with method C. For details, see Deleting Users.	Private Cloud AI Administrator
C - Remove the Private Cloud AI Administrator or Private Cloud AI User role from the user in the HPE GreenLake Cloud Platform	• User is denied access to HPE AI Essentials Software. • Does not delete the user namespace or resources, such as Spark applications and notebooks. • Useful to temporarily remove user access to HPE AI Essentials Software or in conjunction with method B to permanently remove user access to HPE AI Essentials Software. For details, see **Removing user roles**.	Private Cloud AI Cloud Administrator

Disabling Users

You can disable one user or multiple users at a time in HPE AI Essentials Software.

To disable users:

1. In the left navigation panel, select **Administration > Identity & Access Management**.
2. On the **Identity & Access Management** page, disable one or multiple users:
   - To disable one user:
     a. Identify the user in the list.
     b. Click the three-dot icon in the **Actions** column and select **Disable**.
     c. In the window that appears, confirm the action to disable the user. The status for the user changes to Disabled.
   - To disable multiple users:
     a. Click the checkbox next to the username of each user you want to disable.
     b. Click the **Disable** button.
     c. In the window that appears, confirm the action to disable the user Status for selected users changes to Disabled.

To enable users, perform the same steps. Instead of selecting **Disable**, select **Enable**.

Deleting Users

Deleting users does not remove them from HPE GreenLake Cloud Platform.

To delete users:

1. In the left navigation panel, select Administration > Identity & Access Management.
2. On the Identity & Access Management page:
   - To delete one user:
     a. Identify the user in the list.
     b. Click the three-dot icon in the **Actions** column and select **Delete**.
     c. In the window that appears, type DELETE to confirm the action. The user and the user's resources are deleted from the system.
   - To delete multiple users:
     a. Click the checkbox next to the username of each user you want to delete.
     b. Click the **Delete** button.
     c. In the window that appears, type DELETE to confirm the action. The users and their resources are deleted from the system.

REMEMBER

When you delete users, they can no longer access HPE AI Essentials Software through the HPE AI Essentials Software UI URL; however, if a user accesses HPE AI Essentials Software through HPE GreenLake Cloud Platform, the user is automatically onboarded in HPE AI Essentials Software and will again have access. If you do not want the user(s) to gain access to HPE AI Essentials Software, you must also remove the Private Cloud AI Administrator or Private Cloud AI User role from the user in the HPE GreenLake Cloud Platform workspace. For details, see **Removing user roles**.

User Access to Data

Describes data access management and how to grant users access to data.

HPE AI Essentials 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 users access to specific schemas, tables, views, or buckets in a data source.

Public access grants all users *read* and *write* access to all data in a data source. Alternatively, admins can grant users *read*, *write*, or *read & write* access to specific schemas, tables, views, or buckets in a data source.

TIP

- Users should contact their HPE AI Essentials Software admin to request access.
- Any access granted can also be revoked by an HPE AI Essentials Software admin.
- If an admin deletes a user in HPE AI Essentials Software, the user'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 AI Essentials Software administrators can make a data source publicly accessible. When an admin makes a data source publicly accessible, all users have full access (read and write) permissions on the data source and the data within it.

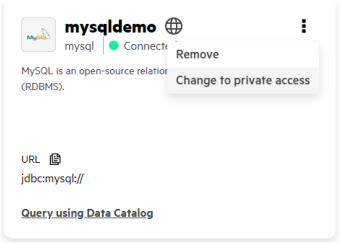
To make a data source publicly accessible, complete the following steps:

1. Sign in to HPE AI Essentials 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 AI Essentials 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 users. Only admins can access the data in the data source.

1. Sign in to HPE AI Essentials 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>`

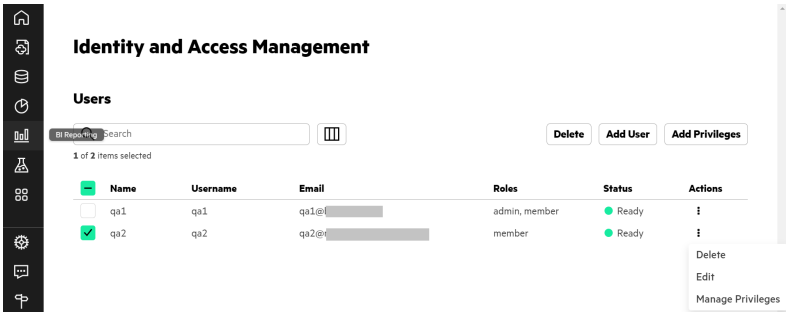
Granting a User Access to Data

HPE AI Essentials Software administrators can grant a user access to one or more tables, views, or buckets in a schema.

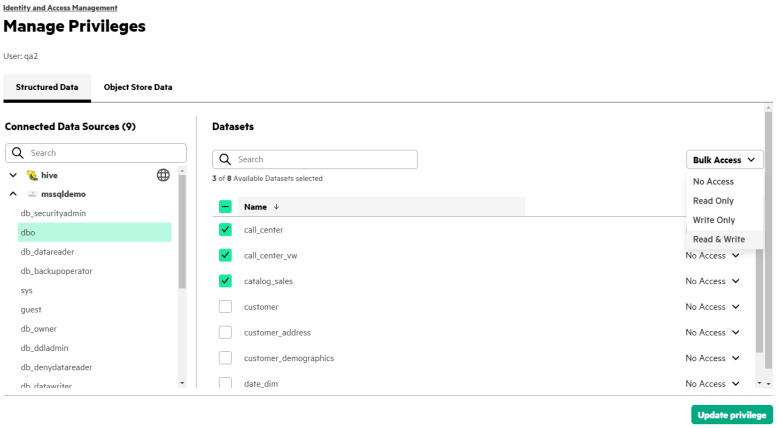
To grant a user access to data, complete the following steps:

1. Sign in to HPE AI Essentials 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 grant the user access to.
6. Expand a data source and select a schema.



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 AI Essentials Software administrators can simultaneously grant a group of users access to one or more tables, views, or buckets in a schema.

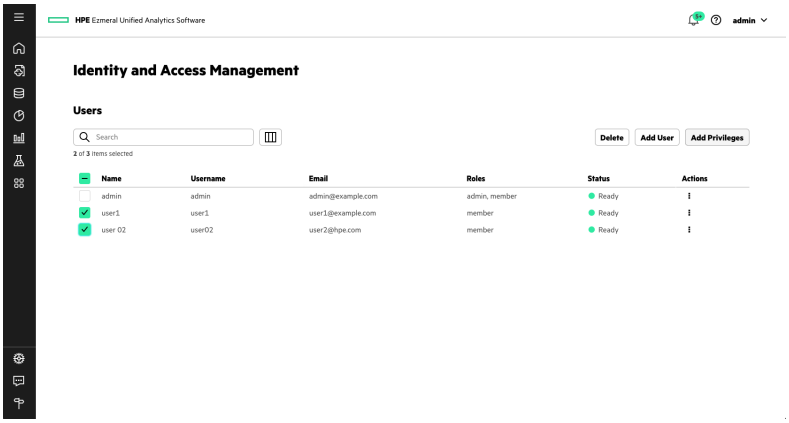
To grant group access to data, complete the following steps:

1. Sign in to HPE AI Essentials Software.
2. In the left navigation bar, select **Administration > Identity & Access Management**.
3. On the **Identity and Access Management** screen, choose one or more users by selecting their checkboxes.

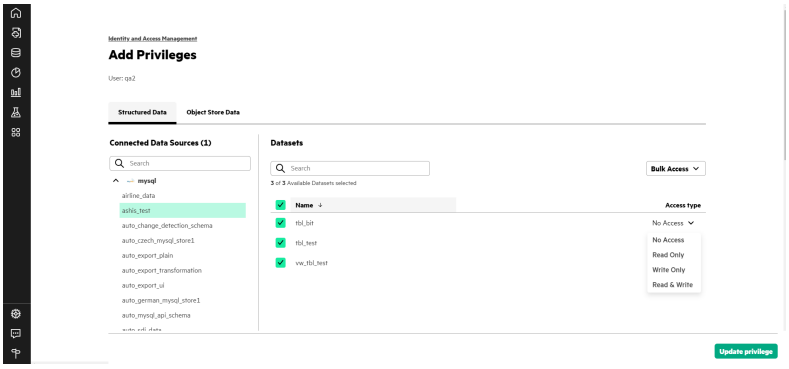
   NOTE

   You cannot select the current session user. For example, if you are signed in as admin, you cannot select admin.

4. Click Add Privileges.



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.



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 User Access to Data

HPE AI Essentials Software administrators can revoke a user's access to schemas, tables, views, and buckets. Revoking access makes the data inaccessible to the user.

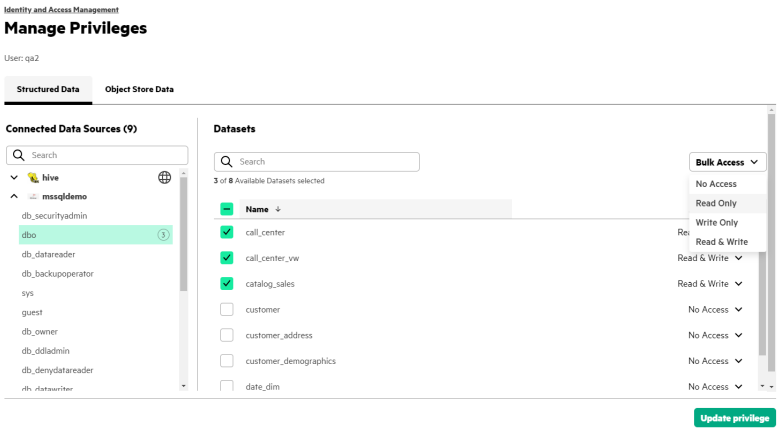
Administrators can use the Manage Privileges screen to revoke access to one or more data sources and their schemas, tables, views, and buckets. To revoke all access to a specific data source for users, use the Remove Privileges option.

> **NOTE**
>
> You can use the Remove Privileges option only for private data sources.

Manage Privileges

To revoke user access to data, complete the following steps:

1. Sign in to HPE AI Essentials 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.



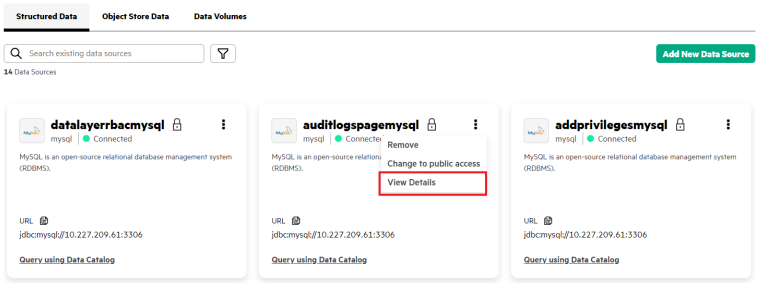
7. In the **Datasets** area, select the tables, views, or buckets that you want to revoke access to.
   - If you are only revoking access to one table, view, or bucket, select **No Access** in the **Access Type** column for the table, view, or bucket.
   - If you are revoking access to multiple tables, 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

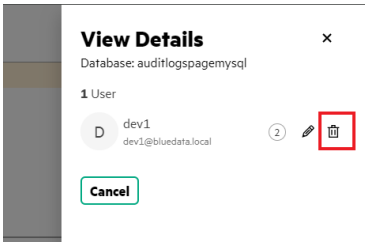
To revoke all access to a specific data source for users, complete the following steps:

1. Sign in to HPE AI Essentials 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.

Data Sources



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)**.

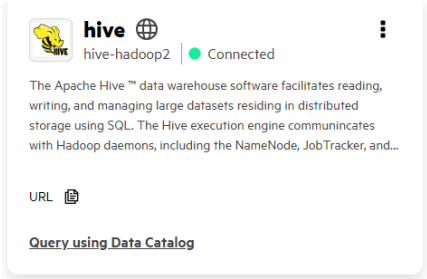
Access Indicator Labels

When users (admins and users) sign in to HPE AI Essentials 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.

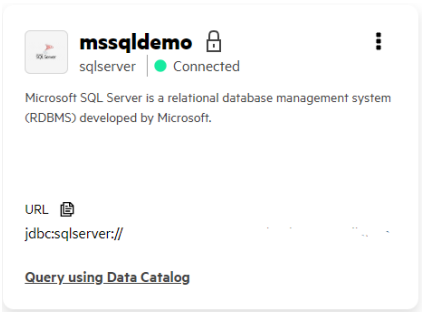
The following sections describe the access indicators that admins and users see on data source tiles.

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 users), the data source tile displays a globe icon next to the data source name, indicating global access.



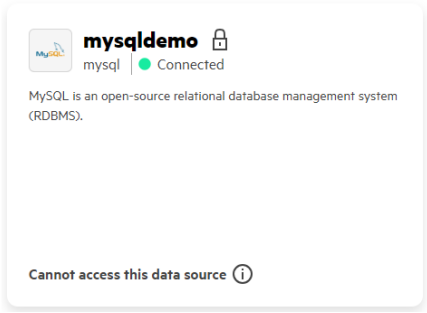
Otherwise, a locked padlock icon displays.



The locked padlock indicates that an admin must grant users access to the data source. All admins have access to the data source.

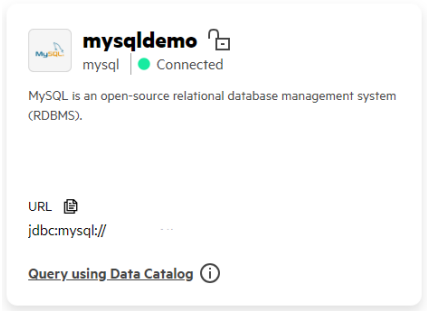
Users

When users **do not** have access to a data source, the data source tile shows a locked padlock icon and says *Cannot access this data source*.



Any attempts to access the data results in an access denied error.

When users 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.



Model Catalog Access

Describes how a Private Cloud AI Administrator manages Private Cloud AI User access on Model Catalog.

A Private Cloud AI Administrator manages Model Catalog access through settings in **Identity & Access Management**. Granting users permission to deploy models also gives them access to the model endpoint.

To grant or revoke the ability for a user or users to deploy models:

1. In the left navigation panel, select **Administration > Identity & Access Management**.
2. Select the user(s) that you want to have deploy permission.
3. Click **Add Privileges**.
4. Select the **Model Catalog** tab.
5. Select the checkbox for one or more models.

6. In the **Access type** dropdown:
   - To grant access, select **Deploy Access**. If granting permission to deploy multiple models, use the **Bulk Access** option, and select **Deploy Access**.
   - To revoke access, select **No Access**. If revoking access to deploy multiple models, use the **Bulk Access** option, and select **No Access**.
7. Click **Update Privilege**.

Model Endpoint Access

Describes how a Private Cloud AI Administrator manages Private Cloud AI User access on model endpoints.

A Private Cloud AI Administrator manages model endpoint access through settings in **Identity & Access Management**.

To add or revoke user access on model endpoints:
1. In the left navigation panel, select **Administration > Identity & Access Management**.
2. Click checkboxes to select one or more users.
3. Click **Add Privileges**.
4. Click the **Model Endpoints** tab.
5. Select the model endpoints that you want to grant the user(s) access on.
6. Select **Endpoint Access**.
   - If you selected more than one model endpoint, click **Bulk Access** and then select **Endpoint Access**.
   - If you selected one model endpoint, click into the dropdown in the **Access Type** column and then select **Endpoint Access**.

   If you want to revoke access, you complete the same steps and select **No Access** instead of **Endpoint Access**.
7. Click **Update Privilege**.

Knowledge Base Access

Describes Knowledge Base access management and how to grant access to the deployed Knowledge Bases.

Prerequisites: Private Cloud AI Administrator access to HPE AI Essentials Software.

A Private Cloud AI Administrator has Full Access to all deployed Knowledge Bases. A Private Cloud AI User has Full Access to Knowledge Bases that they created themselves.

An administrator can manage Private Cloud AI User access to Knowledge Bases.

The following table lists and describes the type of access on Knowledge Bases:

Access Types	Actions
Full Access	• Edit Knowledge Bases. Note that you cannot edit previously selected models. • Interact with your model in the playground. • Enable tracing for your model. • Externally access the Knowledge Base endpoints by using the API token.
Endpoint Access	• View and use the RAG coordinator endpoint to access the playground either from the HPE AI Essentials UI or externally with an API token. • Enable tracing for your model.
No Access	• Cannot access Knowledge Bases. • The deployed Knowledge Bases do not display on the Knowledge Base screen for users.

Granting Access to Knowledge Base

A Private Cloud AI Administrator can grant a Private Cloud AI User access to one or more Knowledge Bases.

To grant a Private Cloud AI User access to Knowledge Base, complete the following steps:
1. Sign in to HPE AI Essentials 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 **Knowledge Base** tab.
6. In the **Knowledge Base** area, select the Knowledge Bases that you want to grant the user access to. .
   - To grant a user access to a single Knowledge Base, select **Endpoint Access** in the **Access Type** column dropdown in the row of the selected Knowledge Base.
   - To grant a user access to multiple Knowledge Bases, use the **Bulk Access** dropdown and select **Endpoint Access** on the selected Knowledge Bases.
7. Click **Update privilege**.

Revoking Access to Knowledge Base

A Private Cloud AI Administrator can revoke a Private Cloud AI User access to Knowledge Bases.

To revoke a Private Cloud AI User access to Knowledge Base, complete the following steps:

1. Sign in to HPE AI Essentials 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 **Knowledge Base** tab.
6. In the **Knowledge Base** area, select the Knowledge Bases that you want to revoke access to.
   - If you are only revoking access to one Knowledge Base, select **No Access** in the **Access Type** column for that Knowledge Base.
   - If you are revoking access to multiple Knowledge Bases, select the Knowledge Bases and then use the **Bulk Access** dropdown (to the right of the **Search** field) and select **No Access**.
7. Click **Update privilege**.

User Access to Projects

Describes how to add and remove users in a shared project. Also describes how to change a project member's role in a shared project.

Both Private Cloud AI Administrators and project admins can add users, remove users, and change the role of project members. A project member can have the *Editor* or *Admin* role. For details, see Project Roles.

Adding Users to a Shared Project

When you add a user to a project, the user is granted access to all the data sources, object stores, and model endpoints that the project has permission to access. The user can only access the data sources, object stores, and model endpoints using the tools and frameworks (Notebooks, Kubeflow, and Spark) inside the shared project. See Project Access to Data.

To add users to a project:

1. In the left navigation panel, select **Projects**.
2. On the **Projects** page, locate the project by name.
3. Click on the project name to open the project page.
4. Click **Add Users**.
5. In the drawer that opens, search for a user by name or email and then select the user. You can continue to search and select users, creating a list of users in the drawer. By default, the **Editor** role displays for each user selected. Private Cloud AI Administrators and project admins can change a member's project role to **Editor** or **Admin**.
6. Click **Add Users**. Added users now appear in the project.

Removing Users from Shared Projects

Private Cloud AI Administrators and project admins can remove users. Users that you remove from a project can no longer access the project or the data sources that the project can access. However, if a user was given direct access on a data source, the user can continue to access the data source.

NOTE

- Removing a user from a project does not remove the user from HPE AI Essentials Software.
- A project continues to exist even if you remove all users from the project.

To remove users from a project:

1. In the left navigation panel, select **Projects**.
2. On the **Projects** page, locate the project by name.
3. Click on the project name to open the project page.
4. Select one or more users and then click **Remove**.
5. In the window that appears, type **REMOVE** to confirm that you want to perform the action. You must type REMOVE in uppercase.
6. Click **Remove**.

Project Access to Data

Describes how to grant a shared project access to data sources, object stores, and model endpoints

Only an Private Cloud AI Administrator can manage access to data sources, object stores, and model endpoints.

To grant a shared project access to data sources, object stores, and model endpoints, the data sources must be connected, and model endpoints must be available. See Connecting Data Sources and Gen AI.

After you grant a shared project access to data sources, object stores, and model endpoints, all project members have access to them. Project members can only access the data sources, object stores, and model endpoints using the tools and frameworks (Notebooks, Kubeflow, and Spark) inside the shared project.

To manage project access to data and model endpoints:

1. In the left navigation panel, select **Projects**.
2. On the **Projects** page, locate the project.
3. In the **Actions** column, click the three-dots menu and select **Manage Privileges**.
4. On the **Manage Privileges** page, select the tab that corresponds with the access you want to grant the project.
   a. **Structured Data**
      i. Click on a data source.
      ii. Select one or more data sets.
      iii. Click **Bulk Access**.
      iv. Select the type of access you want to grant the project.
      v. Click **Add Privilege** or **Update Privilege**.
   b. **Object Store Data**
      i. Click on an Object Store data source.
      ii. Select one or more S3 buckets.
      iii. Click **Bulk Access**.
      iv. Select the type of access you want to grant the project.
      v. Click **Add Privilege** or **Update Privilege**.
   c. **Model Endpoints**
      i. Select one or more model endpoints.
      ii. Click **Bulk Access**.
      iii. Select **Endpoint Access**.
      iv. Click **Add Privilege** or **Update Privilege**.

Data Lakehouse Gateway Access

Describes Data Lakehouse Gateway access management and how to grant access to the registered tables in Data Lakehouse Gateway.

Prerequisites: Private Cloud AI Administrator access to HPE AI Essentials Software.

A Private Cloud AI Administrator has Full Access to all registered tables.

An administrator can manage Private Cloud AI User access to the registered tables in the Data Lakehouse Gateway.

The following table lists and describes the type of access on Data Lakehouse Gateways:

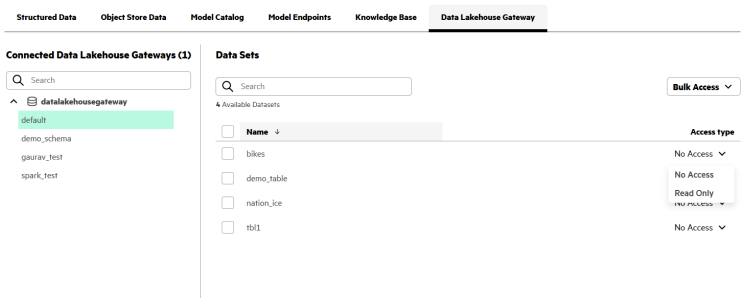
Access Types	Actions
Full Access	• Add external connections. • Register, refresh, and unregister tables. • Select the data sources, schemas, and data sets using Data Catalog. • Run queries, create views, download queries, delete queries, and view query details uisng Query Editor.
Read-Only Access	• Select the data sources, schemas, and data sets using Data Catalog. • Run queries, create views, download queries, delete queries, and view query details uisng Query Editor.
No Access	• Cannot access tables registered via Data Lakehouse Gateway.

Granting Access to Data Lakehouse Gateway

A Private Cloud AI Administrator can grant a Private Cloud AI User access to one or more tables registered via Data Lakehouse Gateway.

To grant a Private Cloud AI User access to Data Lakehouse Gateway, complete the following steps:

1. Sign in to HPE AI Essentials 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 **Data Lakehouse Gateway** tab.



6. In the **Data Lakehouse Gateway** area, expand a Data Lakehouse Gateway and select a schema.
7. In the **Datasets** area, select the registered tables that you want to grant the user access to. You can grant **Read Only** access.
   - To grant a user access to a single table, use the **Access Type** column dropdown in the row of the table.
   - To grant a user access to multiple tables, use the **Bulk Access** dropdown and select the access that you want to grant the user on the selected tables.
8. Click **Add Privilege**.

Revoking Access to Data Lakehouse Gateway

A Private Cloud AI Administrator can revoke a Private Cloud AI User access to tables registered via Data Lakehouse Gateway.

To revoke a Private Cloud AI User access to Data Lakehouse Gateway, complete the following steps:

1. Sign in to HPE AI Essentials 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 **Data Lakehouse Gateway** tab.
6. In the **Data Lakehouse Gateway** area, expand a Data Lakehouse Gateway and select a schema.
7. In the **Datasets** area, select the registered tables that you want to revoke access to.
   - If you are only revoking access to one table, select **No Access** in the **Access Type** column for the table.
   - If you are revoking access to multiple tables, select the tables and then use the **Bulk Access** dropdown (to the right of the **Search** field) and select **No Access**.
8. Click **Update Privilege**.

Ingress Gateway SSL Certificate

Describes the CA and the SSL certificate, that the CA signs, for the ingress gateway to use. Also describes how to configure your browser to trust the CA certificate and how to replace the certificate.

The ingress gateway in HPE Private Cloud AI securely handles all traffic coming into the cluster with a certificate authority (CA). The CA validates the authenticity of the server certificate being used by the ingress gateway.

By default, the ingress gateway uses a secure socket layer (SSL) certificate signed by a self-signed CA for access. You can configure your browser to trust the CA included with HPE Private Cloud AI, or you can replace the ingress gateway certificate with a server certificate, the related CA root certificate, and intermediate certificates.

Configuring Your Browser to Trust the CA Certificate

You can configure your browser to trust the HPE Private Cloud AI CA certificate. Trusting the CA minimizes the number of security warnings that you have to accept when using HPE Private Cloud AI. If you do not trust the CA, a security warning is raised every time you use a service or framework within HPE Private Cloud AI.

To configure your browser to trust the included CA, see Configuring Your Browser to Trust the CA Certificate .

Replacing the Ingress Gateway Certificate

Replacing the ingress gateway certificate allows you to set up automated certificate renewal on your IT organization's schedule, if your CA works with cert-manager.

The generated certificate must cover the DNS domain used by the externally exposed services (web UIs and API endpoints) of the HPE Private Cloud AI cluster and have a subject or SAN that uses a wildcard (*) to cover the domain. For example, ***.my-aie.com**.

> **IMPORTANT**
> Do not use this domain for any other purpose.

You can replace the CA certificate using either of the following methods:

Cert-Manager

Generate and replace the certificate artifacts using the cluster's cert-manager. You can also use cert-manager to renew certificates automatically. See Using cert-manager to Generate and Replace Certificates.

Manually

Generate and replace the certificate artifacts manually using kubectl or another Kubernetes API client. See Manually Generating and Replacing Certificates.

Subtopics

Using cert-manager to Generate and Replace Certificates

Describes how to use cert-manager to generate and replace or renew certificates for the ingress gateway in HPE Private Cloud AI.

IMPORTANT

- This document describes how to use cert-manager with Let's Encrypt for the certificate authority (CA) and Google Cloud as the DNS service. However, Let's Encrypt and Google Cloud are not a requirement. You can use your preferred CA to generate server certificates and service to manage your DNS domain.
- The Let's Encrypt server certificates are always generated with a 90 day lifetime. A 90 day lifetime is not appropriate for a "real" long-lived production installation unless you set up automatic renewal using cert-manager.

To generate and replace existing certificates using cert-manager, you must have permission to delete, add, and modify the following namespaces in the HPE Private Cloud AI cluster:

- `cert-manager`
- `istio-system`

To generate and replace certificates, complete the following steps:

1. Configure the CA and/or DNS service.
2. Create a secret and ClusterIssuer.
3. Use YAML to create a certificate and replace the existing ingress gateway certificate.

After you complete these steps, you can use a web browser to visit any of your cluster domain's URLs to verify that the server certificate is being presented in a certificate chain that ends with a trusted root certificate from the appropriate CA.

Configuring the CA and/or DNS Services

Configure the CA and/or DNS service(s) to allow cert-manager to authenticate itself to the CA server or prove that it has permission to access the certificate.

For Let's Encrypt and other ACME-supporting CAs, cert-manager uses the ACME v2 protocol to complete a DNS-01 type challenge to prove that it manages the resources covered by the certificate. The **cert-manager documentation** describes configurations for executing challenges with various specific DNS services as well as with other services that support dynamic DNS.

The following steps summarize how to use Google Cloud as the DNS service. For complete details, see the **cert-manager Google CloudDNS documentation**.

1. Create a Google Cloud service account to manage the domain for the HPE Private Cloud AI cluster.
2. Grant the service account privileges. The `dns.admin` permission works, but is likely too permissive for production environments.
3. Download a key file (`key.json`) for the service account.

Creating the Issuer in the Cluster

This section describes issuers, provides links to reference documentation, and provides the steps to create a ClusterIssuer and its secret.

Issuer Concepts

The issuer generates and renews the certificate. Configuring the issuer is a key aspect of using cert-manager. The **cert-manager Issuer Configuration documentation** covers a variety of integration possibilities, including:

- Home-grown CAs
- Vault
- Venafi
- Public CA services, such as Let's Encrypt that support the **ACME protocol**
- Other **cert-manager integrations**, such as **Google Certificate Authority Service** and **AWS Private CA**

Alternatively, some organizations develop a custom certificate manager plugin for corporate server certificates.

Creating the Secret and ClusterIssuer

A ClusterIssuer can react to any certificate request in any namespace, making it the appropriate type of issuer to use here.
Create a ClusterIssuer and its related secret in the `cert-manager` namespace. The secret holds the service account key file (`key.json`). The secret is referenced in the issuer resource.

To create the secret and ClusterIssuer, complete the following steps:

1. To create the secret in the `cert-manager` namespace, run the following command:

```
kubectl -n cert-manager create secret generic clouddns-dns01-solver-svc-acct --from-file=key.json
```

The secret name in this example is `clouddns-dns01-solver-svc-acct`; however, you can use a different name.

2. To create the ClusterIssuer, use the following YAML, replacing attribute values with values specific to your environment:

   NOTE

   This YAML is specifically for Let's Encrypt and Google Cloud DNS. The YAML shows the form of an issuer with an ACME DNS-01 challenge configured.

```
apiVersion: cert-manager.io/v1
 kind: ClusterIssuer
 metadata:
  name: letsencrypt
  namespace: cert-manager
 spec:
  acme:
   server: https://acme-v02.api.letsencrypt.org/directory
   email: appropriate.person@their.domain.com
   privateKeySecretRef:
    name: letsencrypt
   solvers:
   - dns01:
      cloudDNS:
       project: my-gcloud-project
       hostedZoneName: my-ezaf
       serviceAccountSecretRef:
        name: clouddns-dns01-solver-svc-acct
        key: key.json
```

The following tables describes some of the YAML attributes:

Attribute	Value Description
server	ACME v2 endpoint for the CA. For Let's Encrypt, you can use their staging endpoint for testing: `acme-staging-v02.api.letsencrypt.org/directory` The staging endpoint does not generate a certificate with a trusted root certificate, but it does allow you to verify that things are working and it is not rate-limited like the real endpoint. If you use the staging endpoint, you must switch to the real endpoint for production.
email	This email address receives notifications, such as when certificate renewal is approaching.
project hostedZoneName	The project and hostedZoneName are specific to the Google Cloud project hosting the Google Cloud service account and the zone name in Google Cloud DNS.
serviceAccountSecretRef	The serviceAccountSecretRef name is the name of the secret, for example `clouddns-dns01-solver-svc-acct` and the name of the file used to create it (`key.json`).

Creating the Certificate

To create a certificate, complete the following steps:

1. Delete the existing ingress gateway certificate:

```
kubectl -n istio-system delete certificate ingress-cert
```

2. Delete the certificate resource that was used to create the initial self-signed certificate:

```
kubectl -n istio-system delete secret ingress-cert
```

3. Use the following YAML to create a certificate resource, replacing attribute values with values specific to your environment. The certificate resource generates a certificate and replaces the ingress gateway certificate.

```
apiVersion: cert-manager.io/v1
 kind: Certificate
 metadata:
  name: ingress-cert
  namespace: istio-system
 spec:
  secretName: ingress-cert
  issuerRef:
   name: letsencrypt
   kind: ClusterIssuer
   group: cert-manager.io
  commonName: "*.my-ezaf.com"
  dnsNames:
  - '*.my-ezaf.com'
  usages:
  - server auth
  - client auth
  renewBefore: 360h
```

The following table describes some of the attributes in the YAML:

Attribute	Value Description
issuerRef	Specify the ClusterIssuer that you created.
commonName dnsNames	Specify the wildcard domain for the HPE Private Cloud AI cluster.
renewBefore	Specify when to expire the existing certificate and generate a new one.

NOTE

For additional optional attributes, see **Certificate resource**. Supported attributes depend on the CA used.

4. View the cert-manager logs to monitor the certificate resource that you created:

```
kubectl -n cert-manager logs -f -l app=cert-manager
```

Reference Documentation

The following documentation provides additional information related to certificates:

- **Issuer Configuration**
- **ACME Certificate Authorities**
- **Issuers**
- **Google Certificate Authority Service**
- **Secure Kubernetes with AWS Private CA**

Manually Generating and Replacing Certificates

Describes how to manually generate and replace the ingress gateway certificates in HPE Private Cloud AI.

IMPORTANT

- This document describes how to use Certbot with Let's Encrypt for the certificate authority (CA) and Google Cloud as the DNS service. However, Let's Encrypt, Certbot, and Google Cloud are not a requirement. You can use your preferred CA to generate server certificates and service to manage your DNS domain.
- The Let's Encrypt server certificates are always generated with a 90 day lifetime. A 90 day lifetime is not appropriate for a "real" long-lived production installation unless you set up automatic renewal using cert-manager.

To replace certificates, you must have permission to read and overwrite the `ingress-cert` secret in the `istio-system` namespace. The administrative `kubeconfig` has these privileges. Alternatively, you can set up and use a service account that has role-based access control (RBAC) constrained for this purpose.

To manually generate and replace certificates, complete the following steps:

1. Generate certificates. (This step is not required if your IT organization provides you with the certificates.)
2. Collect the certificate artifacts.
3. Replace the existing certificate.

Manually Generating Certificates

This section describes how to generate a Let's Encrypt certificate using the DNS challenge method.

NOTE

Let's Encrypt and Google Cloud are not a requirement. You can use your preferred CA and service to generate server certificates and manage your DNS domain.

Using the DNS challenge method confirms to Let's Encrypt that you manage the domain. The following steps summarize the process:

1. Start the Let's Encrypt process through Certbot.
2. Provide the TXT record to add to your DNS domain information, and verify that the TXT record has propagated.
3. Let's Encrypt generates the certificate.

Prepare

- Verify that you have the information and credentials required to manage the DNS domain.
- To interact with Let's Encrypt using the Certbot container image, set up local directories for Certbot to use:

```
LE_BASE=$HOME/Temp/letsencrypt
 mkdir -p "$LE_BASE/etc/letsencrypt"
 mkdir -p "$LE_BASE/var/lib/letsencrypt"
 mkdir -p "$LE_BASE/var/log/letsencrypt"
```

  Verify that `$HOME` is set to a valid directory. Alternatively, set `LE_BASE` to another location if you prefer to store the files elsewhere.

1 - Start the Let's Encrypt process

To start the Let's Encrypt process, run the following command:

```
UA_DOMAIN=my-ezaf.com
 CONTACT_EMAIL=appropriate.person@their.domain.com
 docker run -it --rm --name certbot \
         -v "$LE_BASE/etc/letsencrypt:/etc/letsencrypt" \
         -v "$LE_BASE/var/lib/letsencrypt:/var/lib/letsencrypt" \
         -v "$LE_BASE/var/log/letsencrypt:/var/log/letsencrypt" \
         certbot/certbot \
         certonly --manual --preferred-challenges=dns --email $CONTACT_EMAIL --
agree-tos -d "*.$UA_DOMAIN"
```

IMPORTANT

- Always use a valid email address as Let's Encrypt sends certification expiration notifications to this address.
- If you run this in an environment that requires a proxy for internet access, the HTTP_PROXY and HTTPS_PROXY values must be communicated to the Docker Container. For additional details, see **Use a proxy server with the Docker CLI**.
- For additional command-line options, refer to the **Certbot documentation**.

2 - Provide the TXT record to your DNS

When prompted, add the TXT record to your DNS domain and then use the link provided to verify that the change has propagated.

NOTE

Steps may vary based on the DNS provider. The following steps are based on Google Cloud DNS.

To provide the TXT record to your DNS, complete the following steps:

1. Select the relevant zone name in your Cloud DNS dashboard.
2. Create a new record set with **ADD STANDARD**.
3. Enter the DNS name returned by the docker `run` command.
4. Choose **TXT** as the record type.
5. For **TXT data**, enter the value returned by the docker `run` command.
6. Click **Create**.

3 - Locate the artifacts

After the process completes, locate the output (certificate artifacts) in the following directory:

```
$LE_BASE/etc/letsencrypt/live/$UA_DOMAIN
```

For additional information, see **Where are my certificates?**

Collecting Artifacts

You must have the following artifacts to replace the ingress gateway self-signed certificate:

Cert/File	Description
server cert	The concatenation of the server cert and any intermediate certs in the trust chain, up to but not including the trusted root cert.
trusted root cert	The trusted root cert from the certificate authority (CA).
private key	The private key for the server cert.

You can use the `fullchain.pem` and `privkey.pem` as the server certificate and private key. Download the root certificate referenced by `fullchain.pem` and use the root certificate as the CA certificate.

If you need to see which root certificate is referenced by `fullchain.pem`, complete the following steps:

1. Create temporary, separate files from the individual certificates contained in `fullchain.pem`. For example, if there are three different certificate sections delimited in `fullchain.pem`, you can save each of them as separate files (`cert1.pem`, `cert2.pem`, `cert3.pem`).
2. Get the subject and issuer of each certificate. For example, for the `cert1.pem` file:

   ```
   openssl x509 -in cert1.pem -text | grep '\(Issuer:\|Subject:\)'
   ```

   One certificate should have a subject matching the HPE Private Cloud AI cluster domain, for example `"CN = *.my-ezaf.com"` and some issuer. Another certificate should have a subject matching the issuer of the first certificate and its own issuer, with this pattern repeated down the chain.

   The last certificate in the chain should specify an issuer that is missing from the chain, for example:

   ```
   "C = US, O = Internet Security Research Group, CN = ISRG Root X1"
   ```

   In that case, go to the **Let's Encrypt root certs page** and download the self-signed `.pem` format for the `"ISRG Root X1"` cert, as this is your CA certificate artifact (`ca.pem` file).
3. (Optional) Delete the temporary individual certificate files. You only need the three artifacts (`fullchain.pem`, `privkey.pem`, and `ca.pem`) to replace the existing ingress gateway certificate.

Manually Replacing Certificates

To replace the ingress gateway certificate (including CA certificate and private key), complete the following steps:

1. Delete the existing ingress gateway certificate:

```
kubectl -n istio-system delete certificate ingress-cert
```

2. Back up the existing certificate information:

```
kubectl -n istio-system get secret ingress-cert -o yaml > old-secret-ingress-cert.yaml
```

3. Replace the ingress gateway certificate:

```
kubectl -n istio-system create secret generic ingress-cert --from-
file=tls.crt=fullchain.pem --from-file=tls.key=privkey.pem --from-file=ca.crt=ca.pem
--dry-run=client -o yaml | kubectl apply -f -
```

Configuring Your Browser to Trust the CA Certificate

Describes how to configure your browser to trust the HPE Private Cloud AI certificate authority (CA).

Currently, HPE Private Cloud AI creates its own certificate authority (CA) that self-signs an SSL certificate. This method of verification requires you to accept the SSL certificate every time you access the top-level HPE AI Essentials Software domain and any subdomains.

For example, when you sign in to HPE AI Essentials Software, you sign in through the top-level domain and must accept the certificate for access to HPE AI Essentials Software. For any services or frameworks that you access within the HPE AI Essentials Software, such as Kubeflow, you must also accept the certificate.

If you configure your browser to trust the HPE Private Cloud AI certificate authority (CA), you will not receive multiple prompts to accept the certificate when using services and frameworks within HPE AI Essentials Software.

Configuring your browser to trust the SSL certificate signed by the HPE Private Cloud AI CA

To configure your browser to trust the SSL certificate signed by the HPE Private Cloud AI CA, you must configure the browser to trust the HPE Private Cloud AI CA.
To configure your browser to trust the HPE Private Cloud AI CA:

1. Go to **https://home.your-aie-domain/ca** and allow your browser to automatically download the certificate of the HPE Private Cloud AI CA.
2. Change the browser settings to trust the certificate. The steps vary by browser. The following steps apply to Chrome:
   a. Go to **chrome://certificate-manager**.
   b. Under **Custom**, click **Installed by you**.
   c. Next to **Trusted Certificates**, select **Import**.
   d. Browse to the location where you downloaded the HPE AI Essentials Software certificate and select the certificate.
3. Restart Chrome for the changes to take effect immediately.

(Optional) Update your operating system to trust the CA

Steps vary by operating system. The following steps apply to RHEL. RHEL requires the `ca-certificates` package and a properly configured CA trust store.
To update your operating system to trust the CA:

1. Install the `ca-certificates` package:

   ```
   sudo yum install ca-certificates
   ```

2. If not already enabled, enable dynamic CA:

   ```
   sudo update-ca-trust enable
   ```

3. Copy the certificate to the appropriate directory for the system to recognize it. Common locations for CA certificates include:

   - `/etc/pki/ca-trust/source/anchors/`
   - `/usr/share/pki/ca-trust-source/anchors/`

   ```
   sudo cp /path/to/your/ca.pem /etc/pki/ca-trust/source/anchors/
   ```

4. To update the CA trust store, run the `update-ca-trust` command to refresh the system's trust store:

   ```
   sudo update-ca-trust
   ```

5. To verify that the certificates were added to the trust store, run the following command and check the output:

   ```
   cat /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem | grep "Your CA name"
   ```

   Alternatively, you can list the trust store:

   ```
   trust list
   ```

Replacing the HPE AI Essentials Software Built-In Self-Signed CA with a Custom CA

Describes how to configure your HPE AI Essentials Software cluster to use a custom certificate authority (CA) instead of the built-in self-signed CA included with HPE Private Cloud AI.

Although you can use the built-in self-signed CA included with HPE Private Cloud AI, using a custom CA is a best practice that provides enhanced security and control over your HPE AI Essentials Software cluster.

Prerequisites

You must provide:

- The CA certificate chain that you want to use for your HPE AI Essentials Software cluster. HPE strongly recommends using a chain that includes the root CA. For details, see **https://cert-manager.io/docs/configuration/ca/#deployment**.
- The private key for the intermediate CA that you want to use for your HPE AI Essentials Software cluster.

Steps

To replace the built-in self-signed CA with your custom CA:

1. Back up the existing certificate information:

   ```
   kubectl -n istio-system get secret ingress-cert -o yaml > old-secret-ingress-cert.yaml
   ```

2. Delete the built-in `ezaf-root-ca` CA certificate:

   ```
   kubectl -n istio-system delete certificate ezaf-root-ca
   ```

3. Delete the `ezaf-root-ca` secret:

   ```
   kubectl -n istio-system delete secret ezaf-root-ca
   ```

4. Create the `ezaf-root-ca` TLS secret with your CA certificate chain and the **unencrypted** private key for the intermediate CA that you want to use for your HPE AI Essentials Software cluster:

   ```
   kubectl -n istio-system create secret tls ezaf-root-ca --cert=/path/to/ca-certificate-
   chain.pem --key=/path/to/unencrypted-private-key.pem
   ```

5. Delete the existing `ingress-cert` secret to enforce cert-manager to re-issue its SSL certificate with your new intermediate CA:

   ```
   kubectl -n istio-system delete secret ingress-cert
   ```

   NOTE

   The trust-manager observes the change in the `ingress-cert` secret and updates the certificates used by pods across namespace with the latest ones.

6. Verify that the `ingress-cert` secret includes the new ingress gateway SSL certificate alongside the certificate chain that you provided:

   ```
   kubectl -n istio-system get secret ingress-cert -ojsonpath='{.data.tls\.crt}' | base64 -
   d | \
       awk -v cmd='openssl x509 -noout -fingerprint -sha256 -subject -issuer'
   '/BEGIN/{close(cmd)}; {print | cmd}'
   ```

Observability

Describes observability in HPE AI Essentials Software.

Observability in HPE AI Essentials Software is built on the industry-standard tools that provide comprehensive monitoring capabilities for the platform and applications.

Subtopics

Monitoring
Describes monitoring and available metrics in HPE AI Essentials Software and how Private Cloud AI Cloud Administrators can access them.
Alerting
Describes alerting in HPE AI Essentials Software.
Logging
Describes how logging works in HPE AI Essentials Software and how to access log files for the platform and applications.
Audit Logging
Describes auditing in HPE AI Essentials Software and how to access audit logs.

Monitoring

Describes monitoring and available metrics in HPE AI Essentials Software and how Private Cloud AI Cloud Administrators can access them.

Monitoring focuses on the systematic collection, analysis, and interpretation of data related to the performance and health of systems, applications, and infrastructure. It uses metrics and logs to track the state, behavior, and resource utilization of a Kubernetes cluster and its components in real-time.

In HPE AI Essentials Software, Prometheus automatically collects these metrics and you can visualize them using Grafana. Private Cloud AI Cloud Administrators receive alerts about potential issues. You can use the metrics and logs to view cluster status and make adjustments to maintain optimal cluster and application operations. Also, use the alerts to respond promptly to critical events. For details, see Prometheus and Grafana.

HPE AI Essentials Software includes OTel for forwarding metrics and logs to external monitoring systems. This allows you to integrate OTel with your existing monitoring infrastructure. For details, see OpenTelemetry (OTel).

Component Workloads

HPE AI Essentials Software includes platform components and application components such as Kubeflow and Airflow. Tracking resource usage of these components is essential for Private Cloud AI Cloud Administrators.

There are four categories of component workloads (pods):

Component Workloads	Description
Application Core Workloads	The application installer initializes the core workloads. These workloads allow the instantiation of user-initiated tasks such as notebooks and jobs. Examples include notebook controllers, EzPresto core querying workloads, and inference job controllers.
Application User Workloads	Users initiate these workloads. Examples include notebooks, inference jobs, Spark jobs, and more.
Infrastructure or Platform Workloads	These workloads handle core platform functions such as monitoring the UI, managing users, and connecting to Data Fabric.
Imported Application Workloads	These workloads are imported using the Importing Frameworks functionality. You must manually specify the labels for these 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>"`

Component Metrics

The following table lists and describes the component metrics in HPE AI Essentials Software:

Component Metrics	Description
KServe/Knative Metrics	KServe and Knative provide metrics for model serving workloads, including: • Request count and latency • Memory and CPU usage • Autoscaling metrics
MLflow Metrics	MLflow exports the following metrics: • Total number of incoming HTTP requests (mlflow_http_request_total) • Total duration in seconds (mlflow_http_request_duration_seconds_sum) • Total count of HTTP requests (mlflow_http_request_duration_seconds_count) Prometheus automatically collects these metrics and you can visualize them using Grafana. See Prometheus.
Ray Metrics	Ray provides extensive metrics for monitoring compute jobs and clusters, including: • Node and worker process metrics • Task throughput and latency • Memory usage and object store statistics To view Ray metrics, see the Enabling Metrics in the Ray Dashboard.
MLIS Metrics	HPE Machine Learning Inferencing Software (MLIS) facilitates access to metrics generated by the underlying components used in your deployed model services (e.g., Kubernetes, KServe, BentoML, OpenLLM, and NIM). These components provide numerous metrics out of the box. The specific metrics available for your deployment depend on the type of model you deploy and the configuration of your system. For details, see View Metrics (Prometheus).

Subtopics

Prometheus
Describes the Prometheus metrics and how to access the Prometheus UI to access these metrics.
Grafana
Describes how to import Grafana to HPE AI Essentials Software and how to connect Grafana to Prometheus to visualize the Prometheus metrics using the Grafana dashboards.
OpenTelemetry (OTel)
Describes OTel and how to configure OTel endpoints in HPE AI Essentials Software.

More information

- Monitoring Troubleshooting

Prometheus

Describes the Prometheus metrics and how to access the Prometheus UI to access these metrics.

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.

By default, Prometheus is installed in the `prometheus` namespace and is managed by the `prometheus-operator`.

> **NOTE**
> In the default configuration, HPE AI Essentials Software does not enable authentication for Prometheus. Kubernetes RBAC and network policies control access for Prometheus.

Prometheus Metrics

HPE AI Essentials Software includes a pre-configured Prometheus instance that automatically collects a wide range of metrics from your cluster. These metrics provide insights into the performance and health of your infrastructure, platform components, and applications.

You can view the available metrics through the Prometheus UI.

The following table lists and describes the default metrics collected by Prometheus in HPE AI Essentials Software:

Metrics Type	Description	Example
Node Metrics	CPU, memory, disk, and network usage at the node level.	• node_cpu_seconds_total • node_memory_MemAvailable_bytes • node_filesystem_avail_bytes
Pod Metrics	Resource usage and status for all pods in the cluster.	• kube_pod_container_resource_requests • kube_pod_status_phase
Container Metrics	Detailed resource usage for individual containers.	• container_cpu_usage_seconds_total • container_memory_usage_bytes
Network Metrics	Traffic, connections, and errors across the cluster.	• container_network_receive_bytes_total • container_network_transmit_packets_total
Storage Metrics	PVC usage, operations, and status.	• kubelet_volume_stats_used_bytes • kubelet_volume_stats_capacity_bytes
API Server Metrics	Request rates, latencies, and errors.	• apiserver_request_total • apiserver_request_duration_seconds
Application-specific Metrics	Metrics from components like KServe, MLflow, Ray, and others.	• mlflow_http_request_total • ray_worker_cpu_utilization

Accessing Prometheus UI

Accessing Prometheus UI Internally

To access the Prometheus UI from within your cluster, navigate to the following Prometheus endpoint:

```
http://prometheus-kubeprom-prometheus.prometheus.svc.cluster.local:9090
```

Accessing Prometheus UI Externally

To access the Prometheus UI from outside the cluster, follow these steps:

1. Use port forwarding to connect to the Prometheus service:

   ```
   kubectl port-forward svc/kubeprom-prometheus -n prometheus 9090:9090
   ```

2. Open a web browser and go to: `http://localhost:9090`.

In the Prometheus UI, you can use the query interface to run Prometheus queries (PromQL) to query metrics.

Querying Metrics

Once you are in the Prometheus UI, you can use PromQL (Prometheus Query Language) to query various metrics.

The following table lists some sample queries:

Metric Description	PromQL Query
CPU utilization by node	100 - (avg by(instance) (irate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)
Memory usage by pod	sum by(pod) (container_memory_usage_bytes{pod!=""})
Disk space available by filesystem	node_filesystem_avail_bytes / node_filesystem_size_bytes * 100
Network traffic by pod	sum by(pod) (rate(container_network_receive_bytes_total[5m]))
Call rate of requests coming into the inference service	rate(revision_request_count[1m])

You can enter these queries in the Prometheus UI to monitor and analyze specific metrics related to CPU utilization, memory usage, disk space, and network traffic in your cluster.

For more information, see **Querying Basics** in the Prometheus documentation.

More information

- Monitoring Troubleshooting
- **Prometheus Documentation**

Grafana

Describes how to import Grafana to HPE AI Essentials Software and how to connect Grafana to Prometheus to visualize the Prometheus metrics using the Grafana dashboards.

Grafana is an open-source analytics and interactive visualization web application that allows you to query, visualize, alert on, and understand your metrics no matter where they are stored. Grafana supports a wide range of data sources including Prometheus. With its powerful and flexible dashboard capabilities, you can create dynamic and customizable dashboards to gain insights into your data in real-time. For details, see **Grafana documentation**.

Grafana is not included with HPE AI Essentials Software by default. However, you can import Grafana to HPE AI Essentials Software using Importing Frameworks functionality.

Installing and Importing Grafana

To install and import Grafana in HPE AI Essentials Software, follow these steps:

1. Install Grafana using helm in a user namespace of your choice. For details, see **Grafana helm installation documentation**.

2. Import Grafana to HPE AI Essentials Software. For details, see Importing Frameworks.

Connecting Grafana to Prometheus

Once you import Grafana, connect it to your Prometheus data source.

Follow these steps:

1. Navigate to the Tools & Frameworks screen underneath your chosen application category where you imported Grafana.
2. Click Open on the Grafana application tile.
3. Log in to the Grafana UI using your Grafana credentials.
4. Go to Configuration → Data Sources.
5. Click Add data source.
6. Select Prometheus.
7. Configure the data source with the following options:

Option	Value
Name	HPE AI Essentials Prometheus
URL	http://prometheus-kubeprom-prometheus.prometheus.svc.cluster.local:9090
Access	Server (default)

8. To verify the connection, click Save & Test.

Importing or Creating Dashboards

Once you connect to the Prometheus data source, use dashboards to visualize metrics.

You can import pre-built dashboards from the **Grafana Dashboard Marketplace** by searching for dashboards related to Kubernetes monitoring, Prometheus, KServe or Knative, MLflow, or Ray.

You can also create your own dashboards using the metrics collected by Prometheus.

The following table lists the dashboards you can set up:

Name	Dashboard Details
Cluster Overview	Node CPU, memory, and disk usage
Pod Resources	CPU and memory usage by pod/namespace
Application Performance	Request rates, latencies, and errors
ML Model Monitoring	Inference latency, throughput, and accuracy

To learn how to create dashboards in Grafana, see **Creating Dashboards**.

More information

- **Grafana Documentation**
- Monitoring Troubleshooting

OpenTelemetry (OTel)

Describes OTel and how to configure OTel endpoints in HPE AI Essentials Software.

OpenTelemetry (OTel) is an open-source observability framework that provides a standardized way to collect, process, and export telemetry data such as metrics and logs from your applications and clusters.

HPE AI Essentials Software includes OTel for forwarding metrics and logs to external monitoring systems. This allows you to integrate OTel with your existing monitoring infrastructure.

The following tables lists the data that OTel collector collects in HPE AI Essentials Software:

Data	Description
Metrics	All metrics collected by Prometheus are available through OTel. These metrics include: • Node, pod, and container resource metrics. • Network and storage metrics. • Application-specific metrics from KServe, MLflow, Ray, and others.
Logs	System and application logs from the cluster, including: • Kubernetes component logs • Application logs • Platform component logs

Configuring OTel Endpoints

To learn how to configure OTel endpoints in HPE AI Essentials Software, see Configuring Endpoints.

More information

- OpenTelemetry Documentation
- Monitoring Troubleshooting

Alerting

Describes alerting in HPE AI Essentials Software.

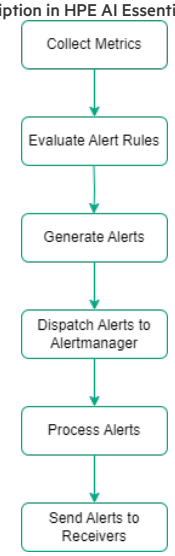
An alert in HPE AI Essentials Software is a system notification that informs you of issues, warnings, and updates. HPE AI Essentials Software uses Prometheus to monitor and collect metrics from nodes, system processes, and applications that run in an HPE AI Essentials Software cluster. HPE AI Essentials Software generates alerts based on the metrics collected. An Alertmanager in HPE AI Essentials Software 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 AI Essentials 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 AI Essentials 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_server", status_code="500"}) by (job) > 5
  ```

- The following alert rules send notifications if the disk has less than 10GB free.

  ```
  node_filesystem_avail_bytes{mountpoint="/"} < 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
- Unusual 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 AI Essentials Software, see List of Alerts.

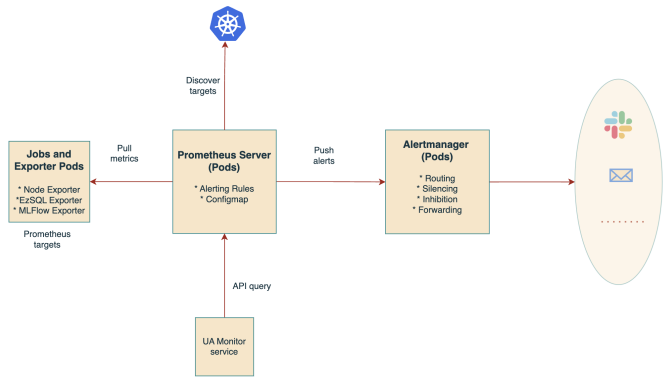
Subtopics

Alerting Architecture and Components
Shows the alerting architecture diagram and describes the alerting components included in HPE AI Essentials Software.
Configuring Alert Forwarding
Describes how to forward alerts via SMTP or Slack and configure the alert policies by updating the `alertmanager_config.yaml` file.
Viewing Alerts and Notifications
Describes how to view alerts and notifications in HPE AI Essentials Software.
List of Alerts
Provides a list of platform alerts and application alerts in HPE AI Essentials Software.

Alerting Architecture and Components

Shows the alerting architecture diagram and describes the alerting components included in HPE AI Essentials Software.

The following image shows the alerting architecture in HPE AI Essentials Software.



The following list describes the alerting components included in HPE AI Essentials 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 AI Essentials 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 .
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.

Subtopics

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.

Understanding the Prometheus Alertmanager Configuration File
Provides an overview of the key components of the Prometheus Alertmanager configuration file, field descriptions, and configuration examples.

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 AI Essentials 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

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 and Configuring Templates and Filtering Alerts.

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.
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:

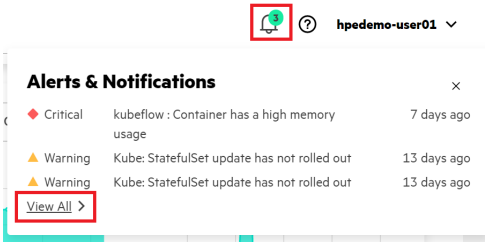
- Friday evenings from 12 PM onwards (hours: 12-23).
- All day Saturday (days: [6]).
- Sunday mornings until 11 AM (hours: 0-11).

Viewing Alerts and Notifications

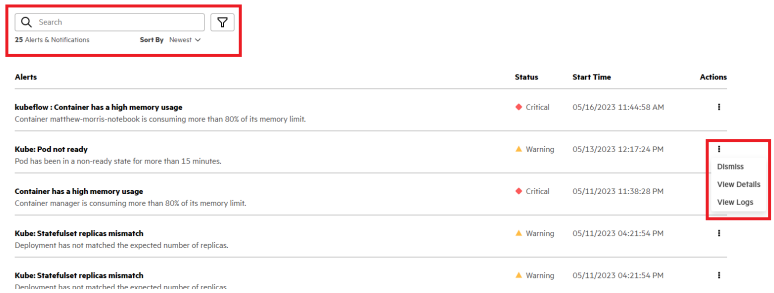
Describes how to view alerts and notifications in HPE AI Essentials Software.

To view the list of alerts and notifications in HPE AI Essentials Software, perform:

1. Sign in to HPE AI Essentials Software.
2. Click the bell icon on top-right of HPE AI Essentials Software homepage.



3. To view all alerts and notifications, click View All.



4. To manage alerts, click the Actions menu.
   Dismiss

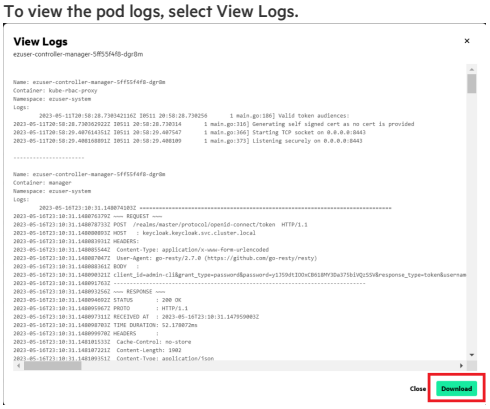
   To dismiss the alerts, select Dismiss.

   View Details

   To view the decsription and relevant metadata for alerts, select View Details.

   View Logs

   To view the pod logs, select View Logs.



   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 AI Essentials Software.

HPE AI Essentials Software issues the following platform alerts:

- Node Alerts
- Etcd Alerts
- Resources Alerts
- Container Alerts
- GPU Alerts
- Storage Alerts
- Kubelet Alerts

- Billing Alerts
- Licensing Alerts
- Licensing Capacity Alerts
- Prometheus Alerts

HPE AI Essentials Software issues the following application alerts:

- Airflow Alerts
- Kubeflow Alerts
- MLflow Alerts
- Ray Alerts
- Spark Alerts
- Superset Alerts

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
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
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
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
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
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
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 AI Essentials 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 AI Essentials 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 AI Essentials 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:

   apps/

   Contains platform and application component logs. This directory contains the following subdirectories:

   app-core/

   Contains logs for core application pods.

   app-user/

   Contains logs for user-initiated pods such as notebooks, inference jobs, Spark jobs, and others.

   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 AI Essentials Software and how to access audit logs.

Auditing provides a chronological set of records that document the events that occur in an HPE AI Essentials 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 AI Essentials 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.
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 • 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.
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 AI Essentials 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 19 th 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 19 th. 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.

Projects

Describes private and shared projects in HPE AI Essentials Software.

A project is a collection of resources in an isolated environment. HPE AI Essentials Software has *private* projects and *shared* projects.

Projects you belong to appear on the **Projects** page in the HPE AI Essentials Software UI. You can access this page by selecting **Projects** in the left navigation panel of the HPE AI Essentials Software UI. The **Type** column lists projects as **Private** or **Shared**.

The following table describes *private* and *shared* projects:

Project Type	Description
Private	• Every user that signs in to HPE AI Essentials Software is automatically assigned a private project. • Use your private project for your personal work. • You cannot grant other users access to your private project or work that you have done in your private project. • Private projects include access to all the tools and frameworks in HPE AI Essentials Software.

Shared	• Accessible to multiple users for collaboration.
	• A Private Cloud AI Administrator must create a project and then add you and your team members to the project.
	• You can be a member of multiple projects.
	• Only project members can access workloads and applications that run in a shared project. For example, members of a project can share and run code in the shared project notebook.
	• Currently, shared projects include the following tools and frameworks:
	  ◦ Notebooks
	  ◦ Kubeflow
	  ◦ Spark
	See Working in Projects.

Project Roles

ATTENTION

- Private Cloud AI Administrators have full access to all projects, and they perform the tasks described in Project Administration.
- Owner is not a project role. In your private project, you are the owner. In a shared project, the user that creates the shared project is labeled as the owner on the **Projects** page.

The following table describes roles as they relate to projects:

Project Role	Description
Admin	• Manages user access to a project. • Can assign roles to project members (Editor or Admin). • Can use project resources and collaborate with the project team.
Editor	• Can use project resources and collaborate with the project team.

Project Access to Data

A Private Cloud AI Administrator grants a shared project access to data sources, object stores, and model endpoints. All users added to a shared project are automatically granted access to those data sources through the project.

For example, if a Private Cloud AI Administrator grants project-a access to a data source, members of project-a can access the data source through any tools and frameworks they use inside project-a, such as the project-a notebook. Members of project-a cannot access the data source through tools and frameworks inside their private projects notebooks or other shared projects. The only way a user would have access to the data source outside of project-a is if:

- A Private Cloud AI Administrator granted the user access to the data source.
- The user belongs to another shared project that also has access to the data source.

Deleting projects does not delete users from HPE AI Essentials Software; however, it does revoke access to workloads and data sources that the members had access to through the project. If users were granted direct access on data, deleting a project does not affect user access to data.

Project Administration

Private Cloud AI Administrators and project admins manage shared projects.

The following table lists the tasks that Private Cloud AI Administrators and project admins perform:

Role	Tasks
Private Cloud AI Administrator	• Create projects • Delete projects • Set vGPU quotas on projects • Manage project access to data sources, object stores, and model endpoints • Add project members • Assign a project admin • Remove project members
Project Admin	• Add project members • Assign a project admin • Remove project members

Subtopics

Working in Projects
Describes how to use tools and frameworks inside shared projects.

Working in Projects

Describes how to use tools and frameworks inside shared projects.

Working with tools and frameworks in a shared project gives all project members access to your work.

You can access tools and frameworks in HPE AI Essentials Software on the **Tools & Frameworks** page by clicking **Tools & Frameworks** in the left navigation panel.

When you **Open** an application, a **Choose Project** prompt appears. You can select a project in the **Project** selector. If only your private project appears in the **Project** selector, you can only use the application in your private project. To continue, click **Open**. Note that you cannot share work in your private project.

For applications that open inside the HPE AI Essentials Software UI, such as Spark and Notebooks, the system does not display the **Choose Project** prompt. Instead, a **Project** selector is provided at the top of the application page. If only your private project appears in the **Project** selector, you can only use the application in your private project.

The following sections describe how to use tools and frameworks inside a shared project:

Notebooks

- When a Private Cloud AI Administrator creates a shared project, the system automatically creates a project notebook that members of the project share.
- Project members can access the project notebook on the **Notebook Servers** page by selecting **Notebooks** in the left navigation panel.
- On the **Notebook Servers** page, you can see your private notebook and notebooks for each project you are a member of.
- The **Project** column lists shared project names and your private project. Your private project is listed as **user-(your-name)**.
- Use the **Project** selector at the top of the page to quickly select a specific project.
- In the **Actions** column, select **Start** to run a notebook server.

Kubeflow

- On the **Tools & Frameworks** page:
  1. Click the filter icon.
  2. In the drawer that opens, select the **Data Science** checkbox.
  3. In the Kubeflow tile, click **Open**.
     In the **Choose Project** window, the **Project** selector lists your private project and the shared projects you are a member of. Your private project is listed as **user-(your-name)**.
  4. Select the project you want to work in.
  5. Click **Open**.
- In the Kubeflow UI, you can move between projects by clicking into the project selector field at the top of the UI and selecting a different project.

Spark

- When you or your project team members create Spark applications, they appear on the **Spark Applications** page, which is accessible by selecting **Analytics > Spark Applications** in the left navigation panel.
- To select a project, use the **Project** selector at the top of the page. The system lists all the Spark applications in the project. You can work with existing Spark applications or create new Spark applications.
- Your private project is listed as **user-(your-name)**.

NVIDIA Blueprints

Describes how to access, import, customize, and use Blueprints in HPE AI Essentials Software.

Blueprints from NVIDIA are customizable AI workflows that help Private Cloud AI Users create and deploy generative AI applications. You can download, customize, and integrate these

pre-defined workflows from NVIDIA into your existing applications.

Accessing NVIDIA Blueprints

To access the full catalog of Blueprints, go to **Blueprints** in NVIDIA.

Using NVIDIA Blueprints

In HPE AI Essentials Software, you can use Blueprints to jumpstart or accelerate the development and deployment of AI use cases. Blueprints cover a wide range of use cases, and new Blueprints are always being added to the NVIDIA Blueprints catalog. Use cases range from designing enterprise RAG pipelines to building AI virtual assistants and digital twins.

IMPORTANT

- Private Cloud AI Users can access and use the full catalog of Blueprints available at **https://build.nvidia.com/blueprints**, but only select Blueprints are tested for use with HPE AI Essentials Software.
- Hewlett Packard Enterprise does not directly support or develop Blueprints. Blueprints are created and maintained by NVIDIA. Each Blueprint has its own software and hardware dependencies listed by NVIDIA. Hewlett Packard Enterprise encourages users to review these dependencies before using a Blueprint. Not every Blueprint will function on a PCAI system.
- Blueprints are not supported for use on the HPE Private Cloud AI Developer System configuration.

NVIDIA AI Virtual Assistant Blueprint Example

NVIDIA AI Virtual Assistant Blueprints are pre-trained and customizable workflows designed to help you build intelligent virtual assistants for a variety of applications.
Scenario

In a customer support team, develop virtual assistants to handle customer inquiries, troubleshoot common issues, and provide quick support.

Workflow

Data Integration

A support personnel feeds the existing customer database and product documentation into the virtual assistant.

User Query

A customer asks about the product features and functionalities.

Response Generation

The virtual assistant retrieves the relevant information from the database and generates a natural language response.

Sentiment Analysis

The virtual assistant analyzes the customer's intent and tone and adjusts the response appropriately. This ensures positive customer experience.

Fine Tuning

Using the feedback from customer interactions, the virtual assistant fine tunes its performance to improve response accuracy over time.

To learn more about the NVIDIA AI Virtual Assistant's software components, architecture, and other key components, see the **Virtual Assistant GitHub documentation** from NVIDIA.

Customizing NVIDIA Blueprints

For some Blueprints, Hewlett Packard Enterprise has prepared edited helm charts that are customized to function on HPE AI Essentials Software. You still need to edit the Blueprint helm charts with items such as required keys, endpoints, and other parameters. To access the prepared helm charts and instructions, see the following GitHub page:**https://github.com/ai-solution-eng/blueprints/tree/main**.

Each Blueprint folder in the GitHub location has a `README.md` file that provides Blueprint-specific details.

Get Started Using NVIDIA Blueprints

HPE AI Essentials Software treats Blueprints like any other third-party framework. To get started using Blueprints, you must use the HPE AI Essentials **Import Framework** feature. To learn more, see Importing Blueprints.

Subtopics

Importing Blueprints
Describes how to import Blueprints in HPE AI Essentials Software.

Importing Blueprints

Describes how to import Blueprints in HPE AI Essentials Software.

Prerequisites

Must have the Private Cloud AI Administrator access to HPE AI Essentials.

About this task

To get started using Blueprints, you must use the **Import Framework** feature. See Importing Frameworks.
The following steps describe the general process of using a Blueprint:

Procedure

1. Review the full catalog of NVIDIA Blueprints at **https://build.nvidia.com/blueprints**. For select pre-validated Blueprints by Hewlett Packard Enterprise, you can review the customized Blueprints documentation at **https://github.com/ai-solution-eng/blueprints/tree/main** to decide on a Blueprint that you want to use.

2. Edit the helm chart accordingly.
3. Sign in to HPE AI Essentials.
4. On the left navigation bar, click the **Tools & Frameworks** icon.
5. On the top-right of the **Tools & Frameworks** screen, click the **Import Framework** button. Use the **Import Framework** wizard to specify the name and other important information about the Blueprint.
6. Import the Blueprint using the helm chart that you edited in step 2.
7. Click **Submit**.

Results

The Blueprint of your choice is imported and installed. You can view it on the Tools & Frameworks screen.

Notebooks

Provides a brief overview of Notebooks in HPE AI Essentials 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 AI Essentials 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 AI Essentials 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 (MLIflow, 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.

Subtopics

Notebook Images Overview
Describes notebook images available in HPE AI Essentials Software and their uses.
Creating and Managing Notebook Servers
Describes how to create and manage notebook servers in HPE AI Essentials Software.
Creating GPU-Enabled Notebook Servers
Describes how to create and deploy the GPU-enabled notebook servers.
Building Custom Kubeflow Jupyter Notebook Image
Describes how to build the custom Kubeflow Jupyter notebook image.
Installing Custom Packages in Kubeflow Notebooks at Runtime
Describes how to install custom packages in existing Kubeflow notebooks that persist between restarts.
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 AI Essentials Software supports line magics and cell magics.
Creating the Conda Environment
Describes how to create a `conda` environment in HPE AI Essentials Software.
Accessing MinIO S3 using Boto3
Describes how to use Boto3 to interact wtih MinIO from a Jupyter Notebook.

Notebook Images Overview

Describes notebook images available in HPE AI Essentials 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 AI Essentials 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.
`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 AI Essentials 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 AI Essentials Software, see Notebook Images.

Creating and Managing Notebook Servers

Describes how to create and manage notebook servers in HPE AI Essentials Software.

Prerequisites

Sign in to HPE AI Essentials 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.

About this task

Create and manage notebook servers in HPE AI Essentials Software.

Procedure

1. Click Notebooks icon on the left navigation bar of HPE AI Essentials Software screen.

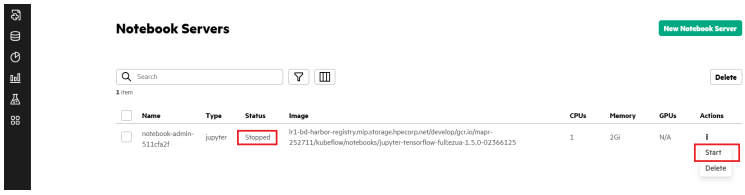
   You are now in the Notebook Servers screen. You can see a default `<notebook-user-namespace-name>` Jupyter notebook has been created with tensorflow image by using Kubeflow Notebooks. By default, the HPE AI Essentials Software creates the default notebook in a Stopped status.

   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.

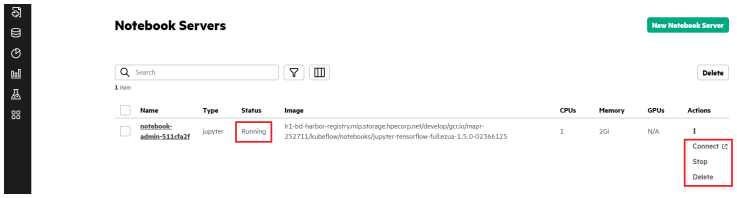
- To connect to a default Jupyter notebook (for example, `<notebook-user-namespace-name>`), follow these steps:

  a. Click the Actions menu.



  b. Select Start and wait for the notebook to be in the `Running` status.

  c. Click <notebook-user-namespace-name> or select Connect.



- 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.

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.

More information

- Creating 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 AI Essentials Software.

About this task

Create GPU-enabled notebook servers in HPE AI Essentials Software.

Procedure

1. Click Notebooks icon on the left navigation bar of HPE AI Essentials 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
     - GPU Vendor: Nvidia
4. Click Launch.

Results

The new GPU-enabled notebook server is created.

More information

- Creating and Managing Notebook Servers

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 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.

   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 AI Essentials Software environments for two types of packages – conda packages and pip packages.

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 a conda environment:

```
conda create --prefix ~/.conda/envs/kf-users-env --clone base -c forge --override-
channels
```

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>
```

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 AI Essentials 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 AI Essentials Software supports built-in magic functions and the custom magics that are described in this topic. HPE AI Essentials 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.

%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.
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.

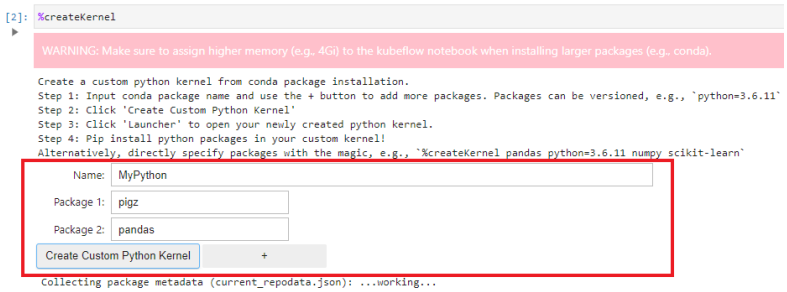
   > **NOTE**
   >
   > If you are using the AWS or Azure environment, do not set the proxy environment variables.

3. Enter the `%createKernel` command in a notebook.
   You can also directly enter packages as arguments with `%createKernel` magic function.

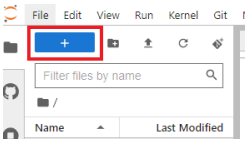
   For example:

   ```
   %createKernel pigz pandas
   ```

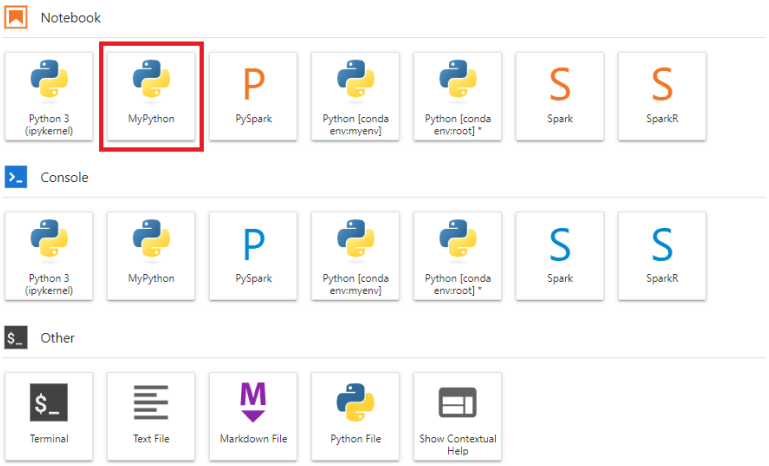
4. Enter the name for your Python kernel in the Name box. In this example, we use MyPython as the custom Python kernel name.
5. Enter the conda package name in the Package 1 box. To enter additional packages, click the + button.
6. Click the Create Custom Python Kernel button.



7. Click the New Launcher button.



8. You can now see your custom Python kernel - MyPython kernel among the available kernels.

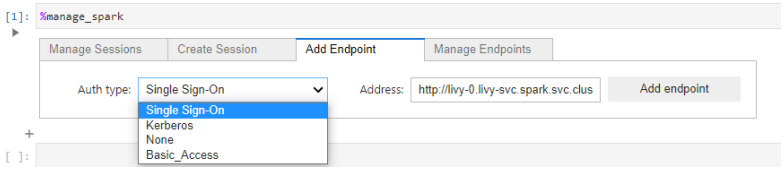


%manage_spark

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.



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.

%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.

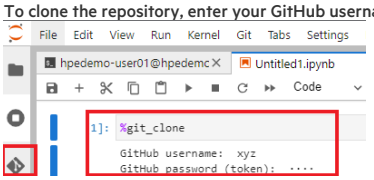
You can also edit the values for other Spark configurations or delete any configurations using this magic command.

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.

%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.



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**.

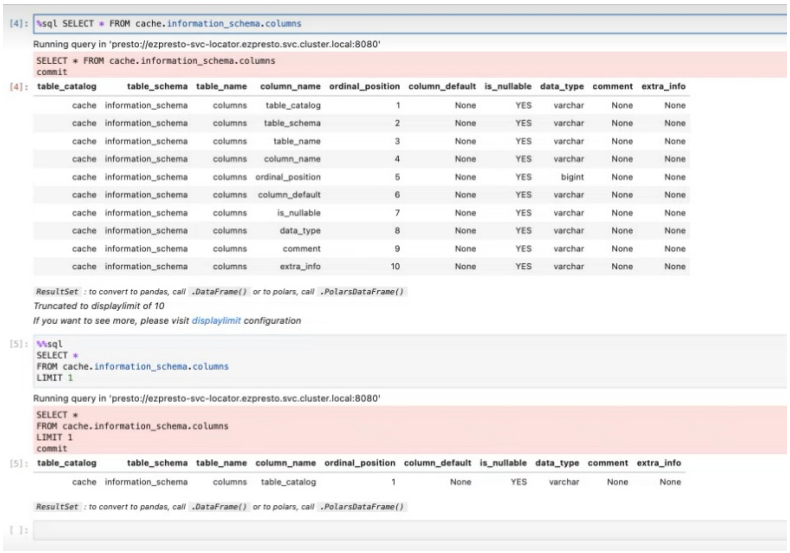
You must use Python kernels to use `%sql` and `%%sql` magic commands. You can directly write and execute SQL queries within a notebook cell. When you run the notebook cell containing `%sql` and your SQL query, the magic command sends the query to the database, runs it, and retrieves the result.

In HPE AI Essentials 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.



%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 AI Essentials Software.

You can use `conda` package management system to create a new virtual environment.

To create a conda environment, run:

```
conda create --prefix ~/.conda/envs/kf-users-env --clone base -c forge --override-
channels
```

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.

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
```

Gen AI

Describes how to access Gen AI to utilize Model Catalog, model endpoints, Knowledge Base, and Playground. Gen AI enables you to generate API tokens, integrate Retrieval-Augmented Generation (RAG) with GPU-Optimized Large Language Models, and interact with models for real-time testing and fine-tuning.

In HPE AI Essentials Software, you can access Gen AI from the left navigation bar. You can use Gen AI to access model endpoints, Knowledge Base, and Playground.

Model Catalog

Model Catalog provides access to a curated selection of high-performance foundation models, including those integrated with NVIDIA AI Enterprise. Model Catalog also offers the flexibility to incorporate models from NGC, Hugging Face, and custom sources through a unified user interface. You can experiment with leading foundation models tailored to your use case, enhance them using techniques like Retrieval-Augmented Generation (RAG), and create intelligent agents that integrate seamlessly with your enterprise systems and data sources. For details, see Model Catalog.

Model Endpoints

Get model endpoints and generate API tokens for external access to the inference services deployed in HPE AI Essentials Software. For details, see Endpoints.

Knowledge Base

Knowledge Base is an integrated framework that enables you to quickly build and deploy Retrieval-Augmented Generation (RAG) solutions. A RAG solution retrieves relevant information from data sources and uses a generative model to produce accurate, textual responses to user queries. For details, see Knowledge Base.

Playground

Chat and interact with your model while testing inputs and fine-tuning responses in real-time. For details, see Playground.

Subtopics

Model Catalog
Describes Model Catalog in HPE AI Essentials Software.

Endpoints
Describes endpoints and access tokens for inference models and knowledge bases deployed in HPE AI Essentials Software.

Knowledge Base
Describes Knowledge Base for building, deploying, and managing secure Retrieval-Augmented Generation (RAG) solutions with GPU-Optimized LLMs in HPE AI Essentials Software.

Playground
Describes how to use Playground in HPE AI Essentials Software.

Model Catalog

Describes Model Catalog in HPE AI Essentials Software.

Model Catalog provides access to a curated selection of high-performance foundation models, including those integrated with NVIDIA AI Enterprise. Model Catalog also offers the flexibility to incorporate models from NGC, Hugging Face, and custom sources through a unified user interface. You can experiment with leading foundation models tailored to your use case, enhance them using techniques like Retrieval-Augmented Generation (RAG), and create intelligent agents that integrate seamlessly with your enterprise systems and data sources.

The following sections describe various aspects of Model Catalog, including features and functionality.

Accessing Model Catalog

You can locate Model Catalog in HPE AI Essentials Software by selecting **Gen AI > Model Catalog** in the left navigation panel.

Models added through the HPE Machine Learning Inferencing Software (MLIS) API display as tiles on the page. Users with permission can deploy models through Model Catalog.

The following table describes default Model Catalog access for users:

User	Permissions
Private Cloud AI Administrator	• Full access on Model Catalog. • Manages user access on Model Catalog.
Private Cloud AI User	• No access on Model Catalog. • Must contact a Private Cloud AI Administrator to request access.

Managing Model Catalog Access

See Model Catalog Access.

Model Catalog UI Features

The following table lists some features available on the Model Catalog page and model tiles:

Feature	Description
Search	Enter your search criteria in the field to quickly locate a particular model. Any information visible on model tiles can be used as search criteria, such as tags or model category.
Grouping	When models are added, they are automatically grouped into one of the following categories: • NVIDIA AI Enterprise (For details, see NVIDIA AI Enterprise). • Hugging Face • S3 If you want to view one category of models, simply click into the drodownd and select the category.
Filter	Use the filter option to locate models that meet the filter criteria, such as status, tags, version, and registry.
Model version	Multiple versions of a model may be available. A model with multiple versions has a version dropdown option in the tile. Select the version of the model you want to deploy and then click **Deploy**. You can deploy each version of the model; however, you must deploy one model version at a time.
Model status indicator	A status indicator in tiles shows the model state: • Ready (model has been deployed) • Staged (model is available but has not been deployed)
View model details	Click the three-dot menu in a tile and select the **View Details** option to view information about the model, including the publisher, version, registry, registry URI, model URI, and more.
View and manage model catalog access (Admin only)	Click the three-dot menu in a tile and select the **Access Details** option to see which users have permission to deploy the model. You can edit or delete user access in the side drawer that opens.

Adding, Editing, and Deleting Packaged Models

You can perform model lifecycle operations on the model catalog through the **Packaged Models** page in the HPE Machine Learning Inferencing Software (MLIS) UI. You can only edit scaling template values if you select a custom scaling template when you deploy a model.

Clicking **Add a Model** in Model Catalog directs you to the **Packaged Models** page. For details, see:

- Add Packaged Model
- Edit Packaged Model
- Delete Packaged Model

Deploying Packaged Models

Admins and users with permission can deploy the packaged models displayed in Model Catalog. Note the following information about model deployments:

- You can edit and delete deployments through the HPE MLIS UI on the **Deployments** page.
- You cannot delete a deployment if the deployment is in use. For example, if a knowledge base is using the deployment, the system does not allow the delete and returns an error message. In this scenario, you can delete the knowledge base and then delete the deployment.
- When you deploy a new model, you have the option of selecting a predefined scaling template or a custom template. See Creating a Custom Scaling Template.

To deploy a model:

1. Click **Deploy** in the tile.
2. If the model was previously deployed, select one of the following options and then click **Confirm**

Option	Details
New Model Endpoint	Deploys the model with a different endpoint than previous deployments. After selecting this option, continue to **step 3**.
Canary Rollout	Use to safely test the performance of your new model version alongside an existing model version that is currently deployed by directing a percentage of the traffic to the new model. For details, see Initiate Canary Rollout. a. Select the model endpoint to serve the model. You can select any deployment that is not using this model. b. Set the percentage of traffic you want to flow to the model. c. Click **Confirm** to deploy the model. Once deployed, you can change the model and traffic for canary rollouts through the Inference Model Endpoints.

3. Enter a unique name for the model.
4. Complete the form fields based on your deployment criteria. If you want to create and use a custom scaling template, see Creating a Custom Scaling Template.
5. Click **Deploy**.
   You can view the deployed models on the **Model Endpoints** page in the HPE AI Essentials Software UI.

   See Inference Model Endpoints.

Creating a Custom Scaling Template

When you create a custom scaling template, the template is available for selection when you deploy models.

To create a custom scaling template:

1. On the Model Catalog page, click **Add New Model**.
2. If the **Create New Packaged Model** dialog appears, click **Cancel**.
3. In the upper right-hand corner of the HPE MLIS page, select **Global Administration**.
4. Select **Auto scaling templates**.

5. Click **Add new auto scaling template**.
6. Complete the form fields, and click **Create Template**.

You can edit or delete the scaling template. If you edit the template, any knowledge base using the model is automatically updated to reconcile the changes.

Subtopics

NVIDIA AI Enterprise

Describes how Private Cloud AI Administrator and Private Cloud AI User can use GPU-Optimized large language models (LLMs) from NVIDIA AI Enterprise to improve machine-learning and AI-application performance.

HPE AI Essentials Software integrates GPU-Optimized LLMs from NVIDIA AI Enterprise to accelerate computational tasks and optimize AI model inferencing. The GPU-Optimized LLMs in HPE AI Essentials Software provide easy-to-use microservices for optimized retrieval, fine-tuning, and embedding management. This integration simplifies AI model fine-tuning and deployment for fast and efficient development of AI applications, such as chatbots and generative AI solutions.

For a list of available GPU-Optimized LLMs in HPE AI Essentials Software, see GPU-Optimized LLMs.

Activating NVIDIA Software Subscription

To learn how to activate the NVIDIA software subscription in HPE Private Cloud AI, see **Activating the NVIDIA software subscription**.

Configuring NGC API Key for `AI-Essentials-NGC-Catalog`

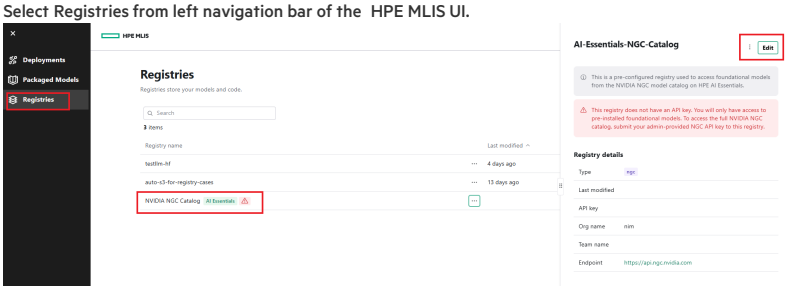
To deploy GPU-Optimized LLMs, the Private Cloud AI Administrator must configure the NGC API key for the `AI-Essentials-NGC-Catalog` registry within HPE Machine Learning Inferencing Software (MLIS). You can configure the NGC API key in two ways:

- Using the HPE MLIS UI.
- Using the HPE MLIS CLI.

Configuring the NGC API Key via the HPE MLIS UI

To configure the NGC API key using the HPE MLIS UI, follow these steps:

1. Sign in to HPE AI Essentials Software as Private Cloud AI Administrator.
2. Navigate to the Tools & Frameworks screen using the left navigation bar.
3. Click Open on MLIS.
4. Select Registries from left navigation bar of the HPE MLIS UI.



5. Click NVIDIA NGC Catalog in the list of registries.

> **NOTE**
>
> You can see a red /!\ icon when the NGC API key is not configured.

6. Click Edit to modify the registry settings.
7. Enter your NGC API key into the API Key box.

> **NOTE**
>
> The NGC API key is provided via email. To learn more, see **Activating the NVIDIA software subscription**.

8. Click Save to finish the configuration.

Configuring the NGC API Key via the HPE MLIS CLI

To configure the NGC API key using the HPE MLIS CLI, follow these steps:

1. Install MLIS CLI (one-time setup):

```
pip install aioli-sdk
```

2. Retrieve the HPE AI Essentials authentication token,
   - Use a Kubernetes configuration file :

```
USERNAMESPACE=<your-username>
```

```
export AIOLI_USER_TOKEN=$(kubectl -n $USERNAMESPACE get secret access-token -
o jsonpath='{.data.AUTH_TOKEN}' | base64 -d)
```

- (OR) Perform an interactive or browser-based login:

```
aioli -c https://mlis.AIEDOMAIN auth login
```

3. To apply the NGC API key to the AI-Essentials-NGC-Catalog registry, run:

```
aioli -c https://mlis.AIEDOMAIN registry update  AI-Essentials-NGC-Catalog  --secret-
key $NGC_API_KEY
```

Once the API key is configured, you can deploy pre-configured GPU-Optimized LLMs or access additional NIMs.

NOTE

During the first deployment of a model, the NIM downloads the necessary data for the NIMs, which may increase deployment time. After the initial deployment, the model is cached, and subsequent deployments will skip the data download step. This will reduce the deployment time.

Viewing Deployed Model Endpoints

You can view the deployed models on the GenAI-->Model Endpoints screen in the HPE AI Essentials Software UI. See Inference Model Endpoints.

GPU-Optimized LLMs in HPE AI Essentials Software

HPE Machine Learning Inferencing Software (MLIS) automatically creates the following five models in HPE AI Essentials Software. You can view and deploy these models from the GenAI -> Model Catalog screen.

These models reference the `AI-Essentials-NGC-Catalog` registry. You can access additional NIMs by creating an NGC API key and applying it to the `AI-Essentials-NGC-Catalog` registry in HPE MLIS. See NGC Registry Setup.

For detailed configuration information required by the models, see **Supported Model** on the NVIDIA documentation.

The following table describes the GPU-Optimized LLMs available in HPE AI Essentials Software and their uses.

GPU-Optimized LLMs	Descriptions	Uses
`nv-rerankqa-mistral-4b-v3`	A re-ranking model developed by NVIDIA, based on the Mistral 4B architecure. Fine-tuned for question re-ranking and question answering tasks.	• Question re-ranking • Question answering • Improving the accuracy of search results in information retrieval systems
`llama-3.1-8b-instruct`	An instruction-tuned version of Meta's Llama 3.1 model with 8 billion parameters. Designed for various natural language processing tasks, including text generation, question answering, and other applications requiring conversational interactions.	• Text generation • Question answering • Text summarization • Other NLP tasks that require understanding and generating human-like text
`llama-3.1-70b-instruct`	An instruction-tuned version of Meta's Llama 3.1 model with 70 billion parameters. Designed for understanding and generating complex text.	• Advanced text generation • Complex question answering • Other advanced NLP tasks that require understanding and generating complex text.
`nv-embedqa-e5-v5`	A transformer-based text embedding model developed by NVIDIA and is part of the NVIDIA NeMo Retriever suite. Designed to transform textual information into dense vector representations for efficient information retrieval.	• Question embedding and answer retrieval • Improving the accuracy of search results in information retrieval systems
`nv-embedqa-mistral-7b-v2`	A high-performance multilingual text embedding model developed by NVIDIA and is part of the NVIDIA NeMo Retriever suite. Designed to transform textual information into dense vector representations for efficient information retrieval.	• Question embedding and answer retrieval • Improving the accuracy of search results in information retrieval systems

To learn more about GPU-Optimized LLMs, see these NVIDIA resources:

- **NIM**
- **NVIDIA NIM for Developers**
- **NVIDIA NIM Offers Optimized Inference Microservices for Deploying AI Models at Scale**
- **NVIDIA NIM for LLMs**

Endpoints

Describes endpoints and access tokens for inference models and knowledge bases deployed in HPE AI Essentials Software.

An endpoint is what client applications use to access a model deployed in HPE AI Essentials Software. When you deploy an inference model through Kubeflow, MLIS, Model Catalog or Knowledge Base, the system provides an endpoint for access.

TIP
Endpoints are compliant with the OpenAI API.

Endpoint Access

The following table describes user and application access to endpoints:

User/Application	Permissions

Private Cloud AI Administrators	• Granted Full Access permissions on all models deployed in the cluster.
	• Can change permissions on endpoints. Endpoints have the following permissions settings:
	  ◦ Full Access
	  ◦ Endpoint Access
	  ◦ No Access

Private Cloud AI Users	• Can only access endpoints for models they deploy (own) or have been given permission to access. • Model owners have Full Access by default. • If Full Access permissions are revoked by an admin, enpoints are visible but cannot be used.
External client applications	Can access endpoints with a valid access token (API or Auth token).

Access Tokens

External client applications, such as a laptop running shell commands or Postman, can securely connect to a deployed model through the endpoint using a valid access token. You can generate API and Auth tokens in the HPE AI Essentials Software UI.
API Tokens

- Only Private Cloud AI Administrators can generate and delete API tokens for deployed models.
- Private Cloud AI Administrators can copy or download tokens and then share them with Private Cloud AI Users as needed.
- API tokens are long-lived tokens with a default lifetime of thirty (30) days.
- Lifetime is configurable.
- After expiration, applications using the token can no longer access the model.

Auth Tokens

- An Auth token is a JSON Web Token (JWT).
- Private Cloud AI Administrators and Private Cloud AI Users can generate Auth tokens.
- Auth tokens are generated in the user's namespace.
- Auth tokens have a thirty (30) minute lifetime.
- Lifetime is **not** configurable.
- After expiration, applications using the token can no longer access the model.

The following topics provide additional details about endpoints and tokens in relation to inference models and the Knowledge Base in HPE AI Essentials Software.

Subtopics

Inference Model Endpoints
Describes where to access inference model endpoints, how to generate and delete access tokens, and how to use access tokens and endpoints for remote access to models.
Knowledge Base Endpoints
Describes how to access knowledge base endpoints, how to generate access tokens, and how to use access tokens and endpoints for remote access to models.

More information

- Gen AI
- Knowledge Base

Inference Model Endpoints

Describes where to access inference model endpoints, how to generate and delete access tokens, and how to use access tokens and endpoints for remote access to models.

Endpoints for KServe inference models deployed through Kubeflow, MLIS, and Model Catalog display on the **Model Endpoints** page in the HPE AI Essentials Software UI.

Locating Inference Model Endpoints

You can access the **Model Endpoints** page from the left navigation panel by selecting **GenAI > Model Endpoints**.

The **Endpoint** column lists the endpoint that provides access to the model.

Model Endpoint Status

A *Not Ready* status typically indicates that the inference service is waiting for access to cluster resources. For example, if a cluster has 2 GPUs and other services are consuming the 2 GPUs, the model endpoint displays *Not Ready* until GPU is available.

If the inference service cannot get resources, the model endpoint status changes to *Pending*. The model endpoint can stay in the *Pending* status for up to 72 hours. After 72 hours, the inference service is no longer a candidate for cluster resources, even if resources become available. At this point, the status changes to *Failed*.

If the status changes to *Failed*, you can re-deploy the inference model. However, if cluster resources are limited, you may need to request help from your cluster administrator.

Model Endpoint Actions

Clicking the three-dot menu for an endpoint in the **Actions** column lists actions that you can take on an endpoint. Admins and model owners have access to the full list of options. If you are not an admin or model owner, an abbreviated list displays.

The following table lists the options in the **Actions** menu:

Option	Description
View Details	Opens a detailed page that provides information about the endpoint. You perform all the same actions on the detailed page, as well as view API examples and generate Auth tokens.
Open Playground	Opens the Playground where you can interact with the model.
Edit	Edit an existing model deployment, including the scaling template for a canary rollout. When you update the scaling template, any knowledge base using the model is also automatically updated to reconcile the changes.
Canary Rollout	Change the model used in the canary rollout or change the percentage of traffic.
Generate API Token	Admins can generate API tokens and share them with other users for use in client applications.
Delete	Admins and model owners can delete model endpoints.

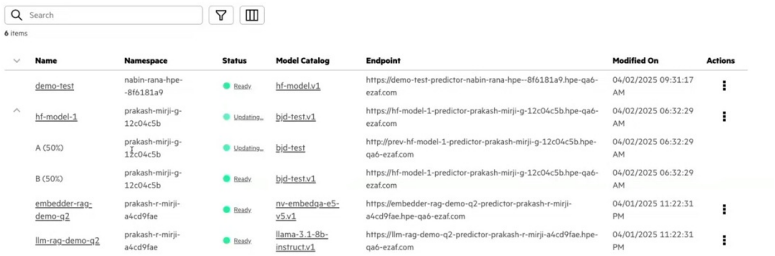
Canary Rollout

Use the expand/collapse option next to the deployment name to see the original deployment and the canary rollout deployment. Under the original deployment you see A and B:

- A is the previous deployment.
- B is the new deployment.

The original endpoint load balances.
The following image shows a canary rollout with A, the original deployment, and B, the new deployment:

Model Endpoints



The **Actions** column for the original deployment provides the following options to modify the canary rollout:

Option	Description
Canary Rollout	Change the model used in the canary rollout and the traffic percentage.
Edit	Change the scaling template.

The **Actions** column for B (new deployment) provides the following options to modify the canary rollout:

Option	Description
Rollout to 100%	Select when you want to replace the original deployment with the canary rollout version.
Cancel the rollout	Sets traffic to 0% for the current model. If you cancel a canary rollout, the inference service will remain present but handle no requests and continue to consume any GPU resources assigned until you perform another rollout. This can be either a 100% roll out of the prior version, or a new rollout of another model version.

Managing Inference Model Endpoint Access

See Model Endpoint Access.

Generating API Tokens for Inference Model Endpoints (Admin Only)

To generate an API token for an inference model:

1. In the left navigation panel, select **GenAI > Model Endpoints**.
2. On the **Model Endpoints** page, click the **Action** option for the inference model.
3. Click **Generate API Token**. The **New API Token** wizard appears.
4. (Optional) Modify the token lifetime. Default is 30 days. A valid value is any number from 1 to 365.
5. Click **Generate API Token**. The system generates the token.
6. Select **Copy** or **Download**. The system does not store the API token for you. You are responsible for storing the token. If you lose the token, you can generate a new token and either keep or delete the old token.
7. Click **Close**.

Deleting API Tokens for Inference Model Endpoints (Admin Only)

When you delete an API token for a model, any application that was using the token can no longer access the model.

To delete an API token for an inference model:

1. In the left navigation panel, select **GenAI > Model Endpoints**.
2. On the **Model Endpoints** page, click the model name in the **Name** column.
3. In the **API Token Management** section, click the **Actions** option.
4. Click **Delete**.

Generating Auth Tokens (JSON Web Tokens)

The user that deploys a model (the model owner) can generate an Auth token and then use the Auth token to access the endpoint from outside the cluster. An Auth token is only valid for thirty (30) minutes, which makes it useful for testing scenarios.

To generate an Auth token:

1. In the left navigation panel, select **GenAI > Model Endpoints**.
2. On the **Model Endpoints** page, click the model name in the **Name** column.
3. Click **API Examples**.
4. Click **Generate Auth Token**. The AUTH_TOKEN variable on the Python, Node, and Shell tabs populates with the new Auth token.

Remotely Accessing a Model

The HPE AI Essentials Software UI provides API examples for Python, Node, and Shell that you can use with a client application for remote access to models. For example, you can run the Shell API example from your laptop to query a model.

To remotely access a model from a client application, use an API example and enter values for the following variables:

Variable	Description
MODEL_ENDPOINT	Enter the model endpoint that you want your client application to connect to.
AUTH_TOKEN	Enter an access token. Even though the variable says AUTH_TOKEN, you can use an Auth token or an API token.

The following code block shows the Shell version of the API example with a model endpoint and generic API token:

```
#!/bin/bash

MODEL_ENDPOINT="https://embedding-mistr-f12b670b-predictor-ezai-services.hpe-
qa6-ezaf.com"
API_PATH="/v1/embeddings"
AUTH_TOKEN="eyJhbGciOiJSUzI1NiIsImtpZCI6InF3cU51cm1lSm..."

RESPONSE=$(curl -s -k -X POST "$MODEL_ENDPOINT$API_PATH"     -H "Content-Type:
application/json"     -H "Authorization: Bearer $AUTH_TOKEN"     -d '{
   "input": ["What is the capital of France?"],
   "model": "nvidia/nv-embedqa-mistral-7b-v2",
   "input_type": "query",
   "encoding_format": "float",
   "truncate": "NONE"
}')

echo $RESPONSE
```

Running this code on the shell command line connects the laptop to the embedding-mistr-f12b670b model and runs the "What is the capital of France?" query against the model. The model returns a response to the query.

To access the API examples in the HPE AI Essentials Software UI:

1. In the left navigation panel, select **GenAI > Model Endpoints**.
2. In the **Name** column, click on a model name.
3. In the upper right corner of the screen, click **API Examples**.
4. Click a tab to view the code examples.
5. (Optional) Click **Generate Auth Token**. A new auth token populates for the AUTH_TOKEN property.

Knowledge Base Endpoints

Describes how to access knowledge base endpoints, how to generate access tokens, and how to use access tokens and endpoints for remote access to models.

A Knowledge Base endpoint provides access to a Retrieval-Augmented Generation (RAG) model in HPE AI Essentials Software. A client application can access a RAG model through the RAG coordinator endpoint with an API token or an Auth token.

RAG Coordinator Endpoint Access

To access a RAG coordinator endpoint, you must have full access permissions on the RAG model. If you deploy a RAG model, you are the owner and have full access by default. If you did not deploy the RAG model, an administrator can grant you full access to the RAG model. For details, see Knowledge Base Access.

An external client application can access a RAG coordinator endpoint, and any model endpoints used in the RAG solution, with an API token or an Auth token. The RAG model owner or an HPE Private Cloud AI Admin can generate access tokens. For details, see Generating API Tokens for Knowledge Bases (Admin Only) and Generating Auth Tokens (JSON Web Tokens).

Locating the RAG Coordinator Endpoint

To locate the RAG coordinator endpoint for a knowledge base:

1. In the left navigation panel, select **Gen AI > Knowledge Base**.
2. On the **Knowledge Base** page, find your knowledge base in the list.
3. Click on the knowledge base **Name**. Note that you can only click on the name after the knowledge base has been deployed and has a **Deployed** status.
4. In the **Details** section, locate the **RAG Coordinator Endpoint**.

Generating API Tokens for Knowledge Bases (Admin Only)

To generate an API token for a knowledge base:

1. In the left navigation panel, select **GenAI > Knowledge Base**.
2. In the **Actions** column for the knowledge base, click the **three-dots** and select **Generate Endpoint Token**.
3. In the **New API Token** window, click **Generate Token**.
4. Click **Copy** or **Download**.
5. Click **Close**.

Generating Auth Tokens (JSON Web Tokens)

The user that deploys a RAG solution through Knowledge Base (the model owner) can generate an Auth token and then use the Auth token to access the RAG coordinator endpoint from outside the cluster. An Auth token is only valid for thirty (30) minutes, which makes it useful for testing scenarios.

To generate an Auth token:

1. In the left navigation panel, select **GenAI > Knowledge Base**.
2. On the **Knowledge Base** page, click Actions for the model you want to generate an Auth token for.
3. Click **Open Playground**.
4. Click the **three-dot** menu and select **API Examples**.
5. Click **Generate Auth Token**.

Remotely Accessing a Model

The HPE AI Essentials Software UI provides API examples for Python, Node, and Shell that you can use with a client application for remote access to models. For example, you can run the Shell API example from your laptop to query a RAG model.

To remotely access a RAG model, use an API example and enter values for the following variables:

Variable	Description
MODEL_ENDPOINT	Enter the RAG coordinator endpoint that you want your client application to connect to.
AUTH_TOKEN	You can use an Auth token or an API token. Only administrators can generate API tokens.

The following code block shows the Shell version of the API example with a RAG coordinator endpoint and generic API token:

```
#!/bin/bash

MODEL_ENDPOINT="https://rag-coordinator.hpe-qa6-ezaf.com"
API_PATH="/v1/embeddings"
AUTH_TOKEN="eyJhbGciOiJSUzI1NiIsImtpZCI6InF3cU51cm1lSm..."

RESPONSE=$(curl -s -k -X POST "$MODEL_ENDPOINT$API_PATH"     -H "Content-Type:
application/json"     -H "Authorization: Bearer $AUTH_TOKEN"     -d '{
   "input": ["What is the capital of France?"],
   "model": "nvidia/nv-embedqa-mistral-7b-v2",
   "input_type": "query",
   "encoding_format": "float",
   "truncate": "NONE"
}')

echo $RESPONSE
```

Running this code from a laptop connects the laptop to the RAG model and runs the "What is the capital of France?" query against the model. The model returns a response to the query based on the data sources selected when the RAG model was deployed.

To access the API examples in the HPE AI Essentials Software UI:

1. In the left navigation panel, select **GenAI > Knowledge Base**.
2. In the **Actions** column for the model, click the **three-dots** and select **Open Playground**.
3. Click the **three-dot** menu for the model and select **API Examples**.
4. Click a tab to view the code examples.
5. (Optional) Click **Generate Auth Token**. A new auth token populates for the AUTH_TOKEN property.

Knowledge Base

Describes Knowledge Base for building, deploying, and managing secure Retrieval-Augmented Generation (RAG) solutions with GPU-Optimized LLMs in HPE AI Essentials Software.

Knowledge Base is an integrated framework that enables you to quickly build and deploy Retrieval-Augmented Generation (RAG) solutions. A RAG solution retrieves relevant information from data sources and uses a generative model to produce accurate, textual responses to user queries.

Knowledge Base integrates retrieval, embeddings, and generation processes to reduce development time for Generative AI applications.

- **Retrieval** involves finding and accessing relevant information from the documents stored in the vector database.
- **Embeddings** numerically represent text data, capturing the meaning and context of words and sentences in a machine-understandable format.
- **Generation** creates new text based on input data and learned patterns.

By combining these three processes, Knowledge Base enables Large Language Models (LLMs) to generate more relevant responses referencing your own data from internal sources across S3, HPE Ezmeral Data Fabric, and HPE GreenLake for File Storage.

The result is tailored, context-aware responses for your specific use cases.

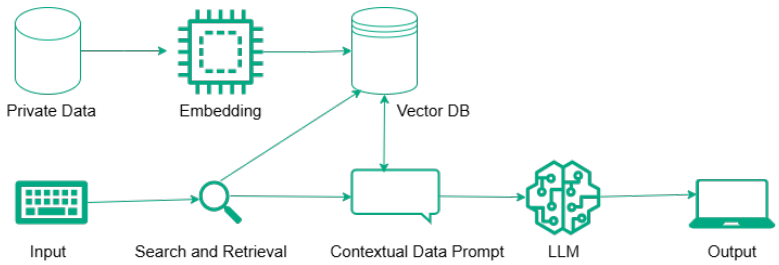
Features and Functionality

Knowledge Base simplifies the creation and management of RAG solutions through a simple interface where you can:

- Securely connect enterprise data sources to RAG solutions to generate accurate, contextually aware responses for reduced risk of inaccuracies or *hallucinations*. See Deploying Knowledge Base.
- Manage the RAG workflow, including data ingestion, retrieval, and automation. See Managing Knowledge Base.
- Utilize advanced options such as custom prompts and endpoint APIs for enhanced control and flexibility. See Deploying Knowledge Base.
- Generate access tokens with the click of a button for secure user and client application access to RAG coordinator endpoints. See **Knowledge Base Endpoints**.
- Experiment with the built-in playground and session context management features. See Playground.

Knowledge Base Architecture

The following diagram shows the key components and workflow in HPE AI Essentials Software Knowledge Base:



The following list describes the Knowledge Base components included in HPE AI Essentials Software:

Private Data

Users can connect private data sources or upload local files. The system processes these files into structured chunks for efficient indexing and retrieval.

Embedding

HPE AI Essentials Software integrates the GPU-Optimized model (for example, NVIDIA Retrieval QA E5 Embedding v5) from NVIDIA into the Knowledge Base workflow. Users can perform semantic search, retrieve context, or retrieve relevant information from the Weaviate Collection using this embedding model. The system converts text into high-dimensional numerical embeddings that capture meaning and context, enabling accurate similarity-based search. This model is built on the NVIDIA software platform and incorporates CUDA, TensorRT, and Triton to provide out-of-the-box GPU acceleration. HPE AI Essentials Software deploys the embedding model, which makes it immediately available for you to use.

Vector DB

The data source provided to Knowledge Base is stored as vectors in the vector database. A vector is a numerical representation of text data that helps the system find and use relevant information to generate accurate responses. HPE AI Essentials Software uses the Weaviate vector database, which is modular and designed to be flexible and scalable. Each Knowledge Base corresponds to one Collection (similar to a table in a traditional database) in the Weaviate vector database. To learn more about Weaviate, see **Weaviate Docs**.

Input

The input is a user's query that can be a question, a statement, or a specific prompt that the Knowledge Base needs to address.

Search and Retrieval

When a user submits a query, the system scans the Weaviate vector database to fetch the data that answers the user's query.

Contextual Data Prompt

The system fetches the most relevant context to the user's query which is then used to create a detailed, context-aware prompt for the LLM.

LLM

NVIDIA AI Enterprise provides prebuilt containers for large language models (LLMs) that can be used to develop chatbots capable of understanding and generating human language. Each GPU-Optimized LLM includes a container and a model, utilizing a CUDA-accelerated runtime for all NVIDIA GPUs, with special optimizations available for various configurations. By default, they are deployed in a scaled-to-zero mode and automatically scale up to serve requests whenever they are received. For details, see Model Catalog.

Output

The output is the final response generated by the LLM after processing the contextual data prompt. This response is the model's answer to the user's original query, using the relevant information retrieved from the data sources.

Accessing Knowledge Base

To access Knowledge Base, click the Gen AI icon on the left navigation bar and select Knowledge Base.

Deploying Knowledge Base

You can deploy Knowledge Base using the Knowledge Base framework through the HPE AI Essentials Software UI. See Deploying Knowledge Base.

Viewing and Managing Knowledge Base

The Knowledge Base screen in HPE AI Essentials Software allows you to manage Knowledge Base statuses through the Gen AI icon. You can see details such as status, metadata name, LLM, model endpoint, and tags. You can also manage access and create API tokens. You can perform different actions based on your role.

As a Private Cloud AI Administrator, you can perform the following actions:

- Edit the Knowledge Base configuration
- Interact with models in the playground
- Generate API tokens
- Manage access details
- Enable Tracing
- Delete Knowledge Bases

As a Private Cloud AI User, you can perform the following actions:

- Edit the Knowledge Base configuration
- Interact with models in the playground
- Enable Tracing
- Delete

You can also filter Knowledge Bases and customize the columns shown in the Knowledge Base table.

For detailed instructions, see Managing Knowledge Base.

Access Management in Knowledge Base

As a Private Cloud AI Administrator, you have unrestricted access to all deployed Knowledge Bases. You can manage and grant access to these Knowledge Bases for Private Cloud AI User. To learn more, see Knowledge Base Access .

Subtopics

Deploying Knowledge Base
Provides instructions for deploying Knowledge Bases in HPE AI Essentials Software.

Managing Knowledge Base
Describes how to manage the deployed Knowledge Base in HPE AI Essentials Software. You can perform various actions such as editing settings, generating API tokens, and managing access details based on your role.

Knowledge Base Observability
Describes observability for Knowledge Base in HPE AI Essentials Software.

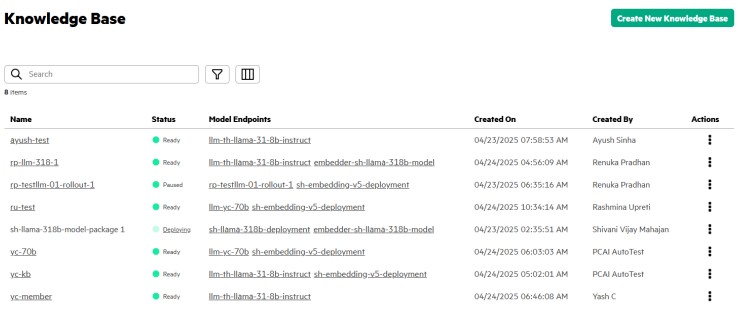
Deploying Knowledge Base

Provides instructions for deploying Knowledge Bases in HPE AI Essentials Software.

Prerequisites: Must have access to HPE AI Essentials Software.

To deploy a Knowledge Base, complete the following steps:

1. Click the Gen AI icon on the left navigation bar.
2. Select Knowledge Base.



3. To create Knowledge Base, click the Create New Knowledge Base button.
4. Navigate through each step within the Deploy Knowledge Base wizard.

The following sections provide instructions for each screen of the deployment wizard:

App Details

Set the following boxes on the App Details step:

Boxes	Instructions

Boxes	Instructions
Select Model	**IMPORTANT** For llama-3.1-70b model deployment only. See Adjusting Resource Allocation for llama-3.1-70b. Click Select Model to choose an LLM. The fields in the wizard populate with the information for the selected model. You cannot change models once your Knowledge Base is deployed. However, you can delete the Knowledge Base and create a new Knowledge Base for a different model. If you are the Private Cloud AI User, you can only view and select the models that you are permitted to access. Private Cloud AI Administrator can provide Private Cloud AI Users access to both model catalogs and model endpoints. Choose one of the following options and click Select: Use Deployed Model Endpoint Private Cloud AI Users can view all previously deployed models, including those they deployed and those deployed by other users, if the Private Cloud AI Administrator has granted them access. Each deployed model is stored in the respective user's namespace. To view models deployed by other users, a Private Cloud AI User must have Endpoint Access to the deployment and Deploy Access to the model catalog for that specific model. **TIP** In HPE AI Essentials Software consider two Private Cloud AI Users, denoted as userA and userB. When userA deploys a model, the HPE AI Essentials creates it in userA's namespace. If the Private Cloud AI Administrator grants userB access to userA's model endpoints, userB can query the model using those endpoints from external clients. However, userB cannot use these endpoints to access PlayGround. To use a PlayGround, userB must have access to both model catalog and model endpoint. Deploys the model with the same LLM used by previously deployed models. Changes to resources (CPU, GPU, Memory) and environment variables impact existing deployed models. When you use a deployed model endpoint, all previously deployed models go into a Deploying state while resources are being redistributed. Knowledge Bases are inaccessible until they are in a Ready state. When you choose this option, you cannot edit auto scaling targets template, minimum instances, maximum instances, and auto scaling metric and target. To modify these settings, go to the Model Endpoints screen. See Endpoints. Deploy New Deploy a new model that has not been deployed previously. You can view all available models that you are permitted to access.
Icon	Click Change Icon and browse to upload an image for your Knowledge Base logo.
Name	Enter a valid name for your Knowledge Base. A valid name: • Must be a regular expression: [a-zA-Z0-9][a-zA-Z0-9-]{0,61}[a-zA-Z0-9] • Cannot exceed 61 characters. • Cannot include dots (.). Supported naming conventions: • abc-xyz • abc-123 • abcxyz • abcxyz-123 Unsupported naming conventions: • abc.xyz • abc.123 **NOTE** You cannot change or edit the Knowledge Base name once your Knowledge Base is deployed.
Description	Enter the Knowledge Base description.
Auto Scaling Targets Template	Defines how auto-scaling behaves for your model endpoint. This template includes parameters such as minimum instances, maximum instances, and auto scaling metric and target that should be manintained for a model endpoint. You can choose a template from the available list or create your own custom template. The fields in the wizard populate with the information for the selected tenplate.
Minimum Instances	The lowest number of instances that the model endpoint will maintain at all times, regardless of demand.
Maximum Instances	The upper limit on the number of instances that the model endpoint can scale up to based on the demand. This ensures that the model endpoint does not scale beyond a predetermined capacity.
Auto Scaling Metric and Target	A metric determines when scaling actions should occur. Examples include CPU, memory, and others. When the metric crosses a defined target value, the auto-scaling system adjusts the number of instances accordingly. Select one of the following metrics and set the target value for that metric. • RPS • Concurrency • CPU • Memory

Boxes	Instructions
Environment Variables	HPE AI Essentials Software supports dynamic environment variables for your model. Set Name and Value for your environment variable, and click + . For example, you can set environment variables such as, • NIM_MODEL_PROFILE : To specify different model configurations dynamically. • HF_TOKEN : To securely authenticate with Hugging Face's API. To add additional environment variables, click + .

Data Source

Supported File Types

The Knowledge Base supports the following file types:

- json
- pdf: ["application/pdf"]
- ppt: ["application/vnd.ms-powerpoint"]
- pptx: ["application/vnd.openxmlformats-officedocument.presentationml.presentation"]
- yaml: ["application/x-yaml", "text/yaml", "text/x-yaml"]
- docx: ["application/vnd.openxmlformats-officedocument.wordprocessingml.document"]
- doc: ["application/msword", "application/doc"]
- rtf: ["application/rtf", "text/rtf", "application/x-rtf", "text/richtext"]
- xls: ["application/vnd.ms-excel"]
- xlsx: ["application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"]
- md: ["text/markdown"]
- html: ["text/html"]
- csv: ["text/csv"]
- txt: ["text/plain"]

IMPORTANT
If you select a data source that contains unsupported file types, the system returns an error and deployment fails.

Selecting or Uploading Files

You can select or upload files for your Knowledge Base. If you do not select a data source or upload files, your solution will deploy as a non-RAG solution, allowing you to use the LLM for chatting.
+ Select Data Source

Select the data source for your Knowledge Base.

NOTE
You can only select a data source to which you have access.

HPE AI Essentials Software supports two types of interfaces: Data Volumes (files and directories) and Object Store Data. You can store your files in any of these sources. You can also, view data files added via external HPE Ezmeral Data Fabric and HPE GreenLake for File Storage.

Select a folder or directory containing your file. You do not need to select individual files; simply choose the high-level directory. HPE AI Essentials Software automatically processes all the files within that directory, embedding and storing them accordingly.

To add additional data source, click + Select Another Data Source.

To learn more about data sources in HPE AI Essentials Software, see Connecting Data Sources.

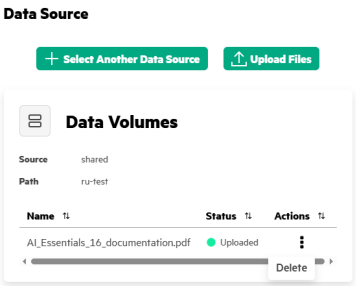
Upload Files

Upload your files to the shared/<name-of-your-knowledge-base> folder. If this folder does not exist, HPE AI Essentials Software will create it during upload.

You can browse and select multiple files to upload.

IMPORTANT
- File names must include only alphanumeric characters, spaces, hyphens, dots, and underscores.
- HPE recommends uploading individual files with a maximum size of 100 MB, as files larger than this can impact performance. However, you can upload folders or directories containing groups of files, with a total size of up to 25 GB.

You can also delete the uploaded files. To delete, click the Actions menu and select Delete.



Embedding Parameters

Once you select the data source or upload files, set the following parameters required for embedding:

Parameters	Descriptions
Embedding Model	Click Select Model to choose models available on HPE AI Essentials Software for embedding. **NOTE** You cannot change or edit models once your Knowledge Base is deployed. However, you can delete Knowledge Base and create a new Knowledge Base for a different model. Lists the embedded model from NVIDIA that are pre-packaged by HPE. If you are the Private Cloud AI User, you can only view and select the models that you are permitted to access. Private Cloud AI Administrator can provide Private Cloud AI Users access to both model catalogs and model endpoints. Choose one of the following options and click Select: Use Deployed Model Endpoint Private Cloud AI Users can view all previously deployed models, including those they deployed and those deployed by other users, if the Private Cloud AI Administrator has granted them access. Each deployed model is stored in the respective user's namespace. To view models deployed by other users, a Private Cloud AI User must have Endpoint Access to the deployment and Deploy Access to the model catalog for that specific model. When you choose this option, you cannot edit auto scaling targets template, minimum instances, maximum instances, and auto scaling metric and target. To modify these settings, go to the Model Endpoints screen. See Endpoints. Deploy New Deploy a new model that has not been deployed previously. You can view all available models that you are permitted to access.
Chunk Size	Length of each data segment when it is divided into smaller parts for processing. It determines the size of each piece or chunk.
Overlap Size Use	The amount of repetition between consecutive chunks. It specifies how much text from one chunk is repeated in the next chunk. This overlap helps to ensure that context is preserved across chunks.
Auto Scaling Targets Template	Defines how auto-scaling behaves for your model endpoint. This template includes parameters such as minimum instances, maximum instances, and auto scaling metric and target that should be manintained for a model endpoint. You can choose a template from the available list or create your own custom template. The fields in the wizard populate with the information for the selected tenplate.
Minimum Instances	The lowest number of instances that the model endpoint will maintain at all times, regardless of demand.
Maximum Instances	The upper limit on the number of instances that the model endpoint can scale up to based on the demand. This ensures that the model endpoint does not scale beyond a predetermined capacity.
Auto Scaling Metric and Target	A metric determines when scaling actions should occur. Examples include CPU, memory, and others. When the metric crosses a defined target value, the auto-scaling system adjusts the number of instances accordingly. Select one of the following metrics and set the target value for that metric. • RPS • Concurrency • CPU • Memory
Environment Variables	HPE AI Essentials Software supports dynamic environment variables for your model. Set Name and Value for your environment variable, and click +. For example, you can set environment variables such as, • `NIM_MODEL_PROFILE`: To specify different model configurations dynamically. • `HF_TOKEN`: To securely authenticate with Hugging Face's API. To add additional environment variables, click +.

Advanced Settings

When you import a model with pre-defined system resources, the default values are based on the minimum values set on the selected model. You can override these default values during deployment.

Only users with the Private Cloud AI Administrator role can update the default resources if a Private Cloud AI User is using one of the already deployed models (sharing the model). If you prefer to use the default values, simply skip the Advanced Settings step and click Next to expedite the deployment process.

You can set the following boxes on the Advanced Settings step:
Retrieval Parameters

Set the following parameters:

Parameters	Descriptions
Number of Documents (K value)	Enter the K value between 1 and 50. The model retrieves K top-ranking documents or chunks from a knowledge base. The model then uses these K documents as context for generating its response. The higher the K value, the more context the model has to work with to generate the response.
Relevance Threshold	Enter the value between 0 and 1. The relevance threshold determines if a document's similarity score is high enough to be included in the results. For example, if you set the value to 0.6, the model will only include documents with a similarity score of 0.6 or higher.

Response Parameters

Set the following parameters:

Parameters	Descriptions
Maximum Tokens	Enter a value between 1 and 2048 to set the maximum length of the output text created by the model.
Temperature	Enter a value between 0 and 1 to set the level of randomness for the output text. A lower value means the text will be more predictable and make more sense. A higher value means the text will be more random.
Top-p (nucleus sampling)	Enter a value between 0.1 and 1. Top-p (nucleus sampling) is a technique to control how diverse and random the output of a model is when generating text. The model creates a list of possible next words along with their probabilities. Top-p adds up the probabilities of the most likely words until that sum reaches or exceeds `p`. Then, the model randomly picks one of these words. The value `p` determines how much randomness you want in the output. For example, if you set the value of Top-p to 0.2, the model only considers a small number of highly likely words, leading to more predictable and less diverse output. However, if you set the value of Top-p to 0.9, this includes more possible words which increases randomness and diversity in the text.
Presence Penalty	Enter a value between -2 and 2. This parameter discourages the model from using any word that already appears in the text, even if it only appears once. A higher presence penalty discourages the model from using the same words and produces more diverse output.
Frequency Penalty	Enter a value between -2 and 2. This parameter reduces the likelihood of repeating the same word or phrase in the generated output. A higher penalty value indicates fewer repeated tokens and produces a more diverse output.
Prompt Template	Choose one of the following templates: • Custom Template • Q&A • Text Paraphrasing • Text Summarization
Template Value	For Custom Template, you can give instructions or information to guide the model's behavior and responses. For example, you can include background information, user preferences, or specific guidelines for the model to follow while generating output. You cannot edit the Template Value box for templates other than the custom template.

Tags

Enter the Knowledge Base tags. For example, you can use informational tags such as large language models, text-to-text, AI model deployment, or other suitable tags to indicate the type of pipeline .

Review and Deploy

Review Knowledge Base

Review the Knowledge Base details. Click the **pencil** icon in each section to navigate to the specific step to change the configuration.

Deploy Knowledge Base

To deploy Knowledge Base, click Submit on the bottom right of the wizard.

View the Deployed Knowledge Base

Once the Knowledge Base is submitted, it will go into the **Deploying** state. You can view it on the Knowledge Base screen.

The different statuses for Knowledge Bases are as follows:

- **Deploying**: Knowledge Base is being deployed.
- **Ready**: Knowledge Base is ready to use.
- **Paused**: When the model has been idle for 30 minutes or more, Knowledge Base enters the Paused status.
- **Starting**: When scaling up `InferenceService`, Knowledge Base goes into the Starting status.
- **Pending**: When there are no available GPU resources to start `InferenceService`, Knowledge Base goes into the Pending status.
- **Error**: Knowledge Base deployment failed.

  TIP

  Click the Error status to open a dialog box that displays the error message.

Related Links

- To start using Knowledge Base, access Playground. For detailed instructions, see Playground.
- To manage Knowledge Base, see Managing Knowledge Base.
- To manage access to Knowledge Base, see Knowledge Base Access.

Adjusting Resource Allocation for llama-3.1-70b

Before you deploy llama-3.1-70b in a cluster with L40S GPUs, you must manually adjust the GPU allocation to 4 to prevent out-of-memory errors. The default GPU allocation is 2.

To adjust the GPU allocation:

1. In the left panel, select **Tools & Frameworks**.
2. Select the **Data Science** tab.
3. Click the **HPE MLIS Endpoint** to access the HPE MLIS UI.
4. In the left panel, select **Packaged Models**.
5. Locate the model and click the ellipsis (**...**).
6. Select **Edit**.
7. Click the **Resources** tab.
8. Change the GPU value from 2 to **4**.
9. Click **Save**.

You can now deploy a knowledge base with the latest version of llama-3.1-70b.

Managing Knowledge Base

Describes how to manage the deployed Knowledge Base in HPE AI Essentials Software. You can perform various actions such as editing settings, generating API tokens, and managing access details based on your role.

A Private Cloud AI Administrator and Private Cloud AI User can view and manage the status of Knowledge Bases in the Knowledge Base screen.

To view and manage Knowledge Base, click the Gen AI icon and click Knowledge Base.

View Details

To view details of your Knowledge Base, click the name of the deployed Knowledge Base. Here, you can see the details of your deployed Knowledge Base such as status, metadata name, LLM, model endpoints, rag coordinator endpoint, and tags. You can view the access details and generate an API token for your Knowledge Base. You can also enable tracing and access the playground to interact with the model.

A Private Cloud AI Administrator can view details of all Knowledge Bases, regardless of who created them. However, a Private Cloud AI User can only view details of the Knowledge Bases that they created themselves. They cannot view details of Knowledge Bases created by other users unless given access. To learn about how to manage access to Knowledge Base, see Knowledge Base Access.

View Actions

To see what actions you can do in the Knowledge Base screen, click the menu icon in the Actions column.

Depending on your user role, you will see different action items in the menu:

Roles	Actions
Private Cloud AI Administrator	Can perform any of the following actions on all Knowledge Bases, regardless of who created them: • Open Playground • Edit • Generate Endpoint Token • Access Details • Enable Tracing • Delete
Private Cloud AI User	Can perform the following on the Knowledge Bases that they created themselves: • Open Playground • Edit • Enable Tracing • Delete Can perform the following on the Knowledge Bases with Endpoint Access: • Open Playground • Enable Tracing

Edit

To change settings and reconfigure the Knowledge Base, select **Edit**. You can update all the Knowledge Base parameters except the name and selected models using **Edit**.

Open Playground

To chat and interact with your deployed Knowledge Base, select Open Playground. For more information, see Playground.

Generate Endpoint Token

To generate an API token, select Generate Endpoint Token. You must set a token lifetime. The default value is 30 days.

After you generate the token, download and save it immediately, as you cannot access it again.

> **NOTE**
> If you do not copy or download the token initially, you cannot retrieve it later, so you must generate a new token.

The API Token Management table displays the token's creation date, expiration date, and status. Deleting a token will revoke access for any Knowledge Base using that API token.

You can externally access these Knowledge Base endpoints by passing the API token.

To learn more, see Endpoints and Knowledge Base Endpoints.

Access Details

To see a list of users who can access the Knowledge Base and their access type, select Access Details. Click the pencil icon to edit the access details for a user. For more information, see Knowledge Base Access.

Enable Tracing

To track the details of your interactions with the Knowledge Base, click Enable Tracing. For details, see Knowledge Base Observability.

Delete

To delete the deployed Knowledge Bases, select Delete.

Filter Knowledge Bases

To filter Knowledge Bases, click the Filter icon.You can filter Knowledge Bases based on various parameters such as type, LLM model, and others. This helps in quickly locating the Knowledge Bases of your choice in the Knowledge Base table.

Display Columns

To select the columns to display on your Knowledge Base table, click the Columns icon.

Knowledge Base Observability

Describes observability for Knowledge Base in HPE AI Essentials Software.

In HPE AI Essentials Software, tracing helps you track and analyze interactions with the Knowledge Base. It allows you to view details such as the inputs provided, parameters set, and responses received for each request. Similarly, citations provide the source of data for responses generated by the Knowledge Base. This helps you identify the origin of the content used in the response and ensures that the data being referenced is reliable and traceable.

Tracing

Tracing allows you to track the details of your interactions with the Knowledge Base. You can log traces for your Knowledge Bases to record inputs, outputs, and metadata associated with each intermediate step of a request. You can also use enable tracing to troubleshoot and find bugs or errors and unexpected behaviours.

Enable Tracing

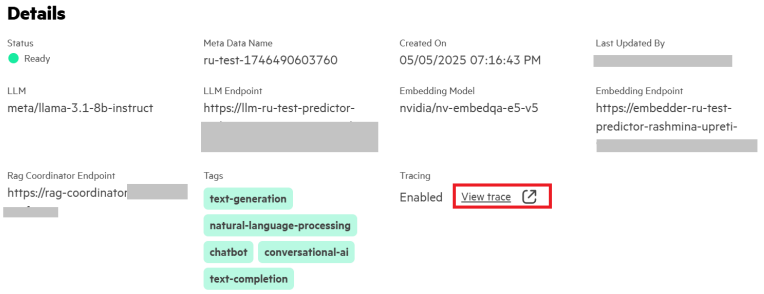
You can enable tracing in the following ways:

1. Click Enable Tracing from the Actions menu for your Knowledge Base in the Knowledge Base screen.
2. Click the name of the Knowledge Base to view its details, and then click the Enable Tracing button on the Details screen.

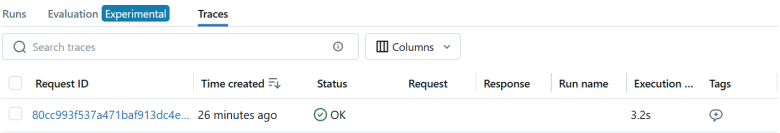
View Tracing

To view trace details, follow these steps:

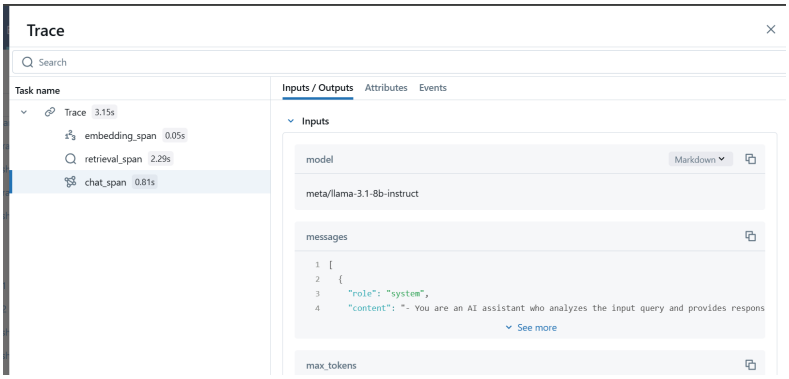
1. Private Cloud AI Administrator must grant Read Access to Private Cloud AI User for the **MLflow bucket** via the **Identity and Access Management** screen. This ensures the necessary access for reading trace stored in that bucket.
2. Navigate to the Details screen of your Knowledge Base.
3. If tracing is enabled, click the View Trace link in the Details box.



4. You can now see the Traces tab in the MLflow screen. Click the Request ID link for your Knowledge Base to open the Trace screen.



On the Trace screen, you can review all the inputs provided to the Knowledge Base, the parameters set, and the responses received.



Disable Tracing

You can directly disable the tracing while chatting with your Knowledge Base.

Citations

Citations allow you to see the source of data for responses generated by the Knowledge Base.
Enabling Citations

You can directly enable citations while chatting with your Knowledge Base. See Playground.

Viewing Citations

When citations are enabled, you can view detailed information for each response, including the file name, file path, author, and creation date.

Disabling Citations

You can directly disable citations while chatting with your Knowledge Base.

Playground

Describes how to use Playground in HPE AI Essentials Software.

Prerequisites: You must have a deployed model endpoint or Knowledge Base. See Deploying Knowledge Base.

Once you have deployed your Knowledge Base to HPE AI Essentials Software, you can view it on the Knowledge Base screen. You can also view the deployed model endpoints on the Model Endpoints screen. You can the access the Playground screen to chat, fine-tune, and interact with your Knowledge Bases or model endpoints.

To access the Playground screen, choose one of these options:

- (For model endpoints and Knowledge Bases) Click the Gen AI icon on the left navigation bar and select Playground.
- (For Knowledge Bases) Click the Actions menu in the Knowledge Base screen and select Open Playground.
- (For Knowledge Bases) Click the name of the deployed Knowledge Base in the Knowledge Base screen and click the Open Playground button.

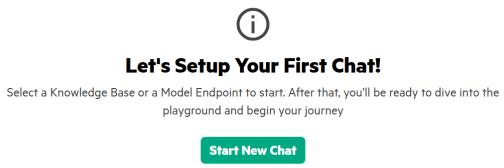
Using Playground

Once you are in the Playground screen, you can chat and interact with your Knowledge Bases or model endpoints in HPE AI Essentials Software.

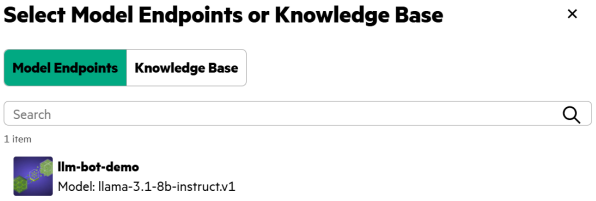
Get Started

To get started,

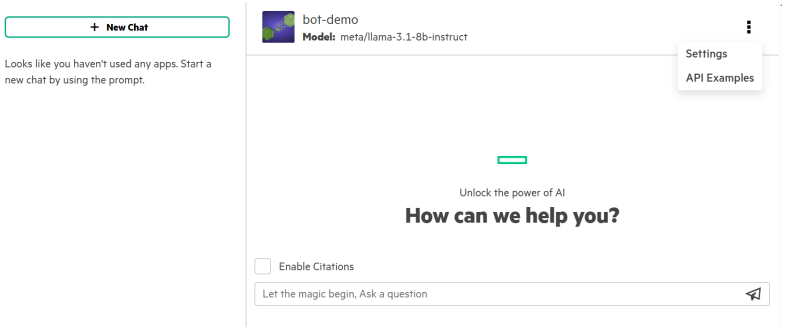
1. Click the Gen AI icon on the left navigation bar and select Playground.
2. Click Start New Chat.



3. View a list of deployed model endpoints and Knowledge Bases.



4. Click the name of the model endpoint or Knowledge Base you want to interact with.



+ New Chat

To start a new chat,

1. Click + New Chat to initiate a new chat session.
2. Choose the model endpoint or Knowledge Base you want to use for the chat.

Access Chat List

Navigate to the left of the Playground screen to find a list of all your previous chats from different playgrounds.

Enable Citations

To enable citations for your Knowledge Base, check the Enable Citations checkbox.

To disable citations, uncheck the Enable Citations checkbox.

NOTE

- This option is only available for Knowledge Bases.
- Citations may appear for greeting queries if the relevance score is set below the minimum threshold of 0.75.

Enable Tracing

Once you have enabled tracing for your Knowledge Base, you can see the Enable Tracing option is selected in the Playground.

To disable tracing, uncheck the Enable Tracing checkbox for your Knowledge Base.

NOTE

This option is only available for Knowledge Bases.

To learn more, see Knowledge Base Observability.

Configure Settings

To configure the settings for your chat session,

1. Click the three-dots menu on the top-right of the Playground screen and select Settings.
2. Modify settings specific to your current chat session.

NOTE

The settings are not saved for future sessions. To apply these settings to the Knowledge Base so that they are overridden, set them from the Edit screen. See Managing Knowledge Base.

Settings	Description
Maximum Tokens	Enter a value between 1 and 2048 to set the maximum length of the output text created by the model.
Temperature	Enter a value between 0 and 1 to set the level of randomness for the output text. A lower value means the text will be more predictable and make more sense. A higher value means the text will be more random.
Top-p (nucleus sampling)	Enter a value between 0.1 and 1. Top-p (nucleus sampling) is a technique to control how diverse and random the output of a model is when generating text.
Presence Penalty	Enter a value between -2 and 2. This parameter discourages the model from using any word that already appears in the text, even if it only appears once. A higher presence penalty discourages the model from using the same words and produces more diverse output.
Frequency Penalty	Enter a value between -2 and 2. This parameter reduces the likelihood of repeating the same word or phrase in the generated output. A higher penalty value indicates fewer repeated tokens and produces a more diverse output.
Prompt Template	Choose one of the following templates: • Custom Template • Q&A • Text Paraphrasing • Text Summarization **NOTE** While chatting with deployments, you can modify the current custom template. However, if you are using a non-RAG LLM to query, avoid using a custom template designed specifically for Knowledge Base.
Template Value	For Custom Template, you can give instructions or information to guide the model's behavior and responses. For example, you can include background information, user preferences, or specific guidelines for the model to follow while generating output. You cannot edit the Template Value box for templates other than the custom template.

API Examples

The API examples show you how to query the selected model endpoint or Knowledge Base and access endpoints using Python, Node, or Shell through an API call. These examples are pre-configured with the necessary settings, allowing you to generate the auth token and make the API call easily.
To view API examples,

1. Click the three-dots menu on the top-right of the Playground screen and select API Examples.
2. Click Generate Auth Token to create an authentication token.

   NOTE

   The Auth Token generated here has a lifetime of 30 minutes. To access Knowledge Base endpoints externally, a Private Cloud AI Administrator can generate a long-lived API Token for use. For details, see Endpoints and Knowledge Base Endpoints.
3. Copy the generated token to use it to access the Knowledge Base.

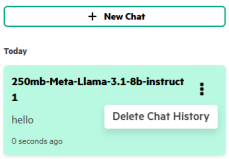
Clear Contexts

To reset context, click Clear Contexts on top-right. This starts a new conversation while the previous chat remains part of the ongoing thread. The new discussion begins with a fresh context.

Delete Chat History

To delete a chat history,

1. Click the three dots menu in the top-left of the Playground screen for the chat you want to delete.



2. Click Delete Chat History and follow the prompt.

Data Engineering

Data engineers can design and build pipelines that transform and transport data into usable formats for data consumers.

HPE AI Essentials 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 AI Essentials 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, the SQL query engine in HPE AI Essentials 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 AI Essentials 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.
When you connect HPE AI Essentials Software to various data sources, you can access the data in those data sources from Superset 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.

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.

Airflow Pipelines

Links to the Airflow interface where you can connect to data sets created in HPE AI Essentials Software and use them in your data pipelines. See Airflow.

Subtopics

Accessing Data in External S3 Object Stores
Describes how to access data in external object stores from clients such as Spark and Kubeflow notebooks.

EzPresto
Describes the EzPresto SQL query engine and its featues.

Data Lakehouse Gateway
Describes Data Lakehouse Gateway for accessing, managing, and governing the data stored in Data Lakehouses.

Airflow
Provides an overview of Apache Airflow in HPE AI Essentials Software.

Superset
Provides a brief overview of Superset in HPE AI Essentials Software.

More information

- Get Started

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 AI Essentials 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 AI Essentials Software administrator provides the access credentials when creating the data source connection. Your access to the data source is authorized through HPE AI Essentials 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

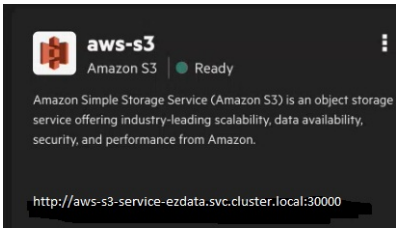
You can find the data source name and endpoint URL on the data source tile in the HPE AI Essentials Software UI.
Once connected, clients can:

- 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 AI Essentials 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:



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 AI Essentials 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.

Subtopics

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 AI Essentials 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 HPE AI Essentials Software, 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.
- 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 AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials Software by going to BI Reporting > Dashboards or Tools & Frameworks > Data Engineering tab and clicking **Open** in the Superset tile. See Superset. 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.

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 AI Essentials 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 and Caching Data.

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.

Subtopics

Connecting Data Sources

Provides instructions for connecting HPE AI Essentials Software to external data sources.

Connecting data sources enables federated access to data for users with the appropriate permissions. HPE AI Essentials 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

- To successfully execute queries, the EzPresto pods must establish network connectivity with the configured data source endpoints; for example, the Hive Metastore, RDBMS, and object store endpoints. Establishing network connectivity includes the ability to resolve hostnames, access the required ports, and authenticate where necessary.
- Only HPE AI Essentials 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**.

   Data Lakehouse Gateway

   Use Data Lakehouse Gateway to access, manage, organize, and govern enterprise data, across a variety of formats and locations. See Data Lakehouse Gateway for more information.

Subtopics

List of Oracle connection parameters, descriptions, default values, and supported data types.

Snowflake Connection Parameters

List of Snowflake connection parameters, descriptions, default values, and supported data types.

SQL Server Connection Parameters

List of SQL Server connection parameters, descriptions, default values, and supported data types.

Teradata Connection Parameters

List of Teradata connection parameters, descriptions, default values, and supported data types.

PostgreSQL Connection Parameters

List of PostgreSQL connection parameters, descriptions, default values, and supported data types.

Iceberg Connection Parameters

List of Iceberg connection parameters, descriptions, default values, and supported data types.

Configuring a Hive Data Source with Kerberos Authentication

Describes the required prerequisite steps to complete before you connect HPE AI Essentials Software to a Hive data source that uses Kerberos authentication.

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. See**https://prestodb.io/docs/current/connector/deltalake.html**.

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
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
Hive Parquet Pushdown Filter Enabled	Enable complex filter pushdown for Parquet	false	BOOLEAN

Parameter	Description	Default Value	Data Type
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
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
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

Parameter	Description	Default Value	Data Type
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
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
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 AI Essentials Software to a Hive data source that uses Kerberos for authentication, see Configuring a Hive Data Source with Kerberos Authentication .

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
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

Parameter	Description	Default Value	Data Type
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
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
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

Parameter	Description	Default Value	Data Type
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
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
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

Parameter	Description	Default Value	Data Type
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
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.

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
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
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)
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	

Parameter	Description	Default Value	Data Type	Possible Values
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	
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	
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	
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	

Parameter	Description	Default Value	Data Type	Possible Values
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	
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)
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)

Parameter	Description	Default Value	Data Type	Possible Values
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)
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	
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	

Parameter	Description	Default Value	Data Type	Possible Values
Hive Max Partitions Per Scan	Maximum allowed partitions for a single table scan	100000	INTEGER	Min(1)
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)
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)
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)

Parameter	Description	Default Value	Data Type	Possible Values
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	
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	
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	

Parameter	Description	Default Value	Data Type	Possible Values
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)
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 AI Essentials Software to a Hive data source that uses Kerberos authentication.

You can connect HPE AI Essentials 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

Step 2 - Configure EzPresto to use the krb5.conf file

Step 3 - Connect HPE AI Essentials Software to the Hive data source

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 AI Essentials 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.locatorSvcPort $
}}
      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,salesforce,sharepoint,prestodb,raptor,kudu,redis,accumu
lo,elasticsearch,redshift,localfile,bigquery,prometheus,mongodb,pinot,druid,cassandr
a,kafka,atop,presto-thrift,ampool,hive-
cache,memory,blackhole,tpch,tpcds,system,example-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,kudu,localfile,memory,mongodb,pinot,presto-bigquery,prestodb,presto-
druid,presto-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.minHeapSize . | floor
}}M
      -Xmx{{ tpl .Values.ezsqlPresto.configMapProp.mst.jvmProp.maxHeapSize . |
floor }}M
      -XX:-UseBiasedLocking
      -XX:+UseG1GC
      -XX:G1HeapRegionSize={{
.Values.ezsqlPresto.configMapProp.mst.jvmProp.G1HeapRegionSize }}
      -XX:+ExplicitGCInvokesConcurrent
      -XX:+HeapDumpOnOutOfMemoryError
      -XX:+UseGCOverheadLimit
      -XX:+ExitOnOutOfMemoryError
      -XX:ReservedCodeCacheSize={{
.Values.ezsqlPresto.configMapProp.mst.jvmProp.ReservedCodeCacheSize }}
      -XX:PerMethodRecompilationCutoff=10000
      -XX:PerBytecodeRecompilationCutoff=10000
      -Djdk.attach.allowAttachSelf=true
      -Djdk.nio.maxCachedBufferSize={{
.Values.ezsqlPresto.configMapProp.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.locatorSvcPort $
}}
        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,salesforce,sharepoint,prestodb,raptor,kudu,redis,accumu
lo,elasticsearch,redshift,localfile,bigquery,prometheus,mongodb,pinot,druid,cassandr
a,kafka,atop,presto-thrift,ampool,hive-
cache,memory,blackhole,tpch,tpcds,system,example-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,kudu,localfile,memory,mongodb,pinot,presto-bigquery,prestodb,presto-
druid,presto-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.minHeapSize . | floor
}}M
        -Xmx{{ tpl .Values.ezsqlPresto.configMapProp.wrk.jvmProp.maxHeapSize . |
floor }}M
        -XX:-UseBiasedLocking
        -XX:+UseG1GC
        -XX:G1HeapRegionSize={{
.Values.ezsqlPresto.configMapProp.wrk.jvmProp.G1HeapRegionSize }}
        -XX:+ExplicitGCInvokesConcurrent
        -XX:+HeapDumpOnOutOfMemoryError
        -XX:+UseGCOverheadLimit
        -XX:+ExitOnOutOfMemoryError
        -XX:ReservedCodeCacheSize={{
.Values.ezsqlPresto.configMapProp.wrk.jvmProp.ReservedCodeCacheSize }}
        -XX:PerMethodRecompilationCutoff=10000
        -XX:PerBytecodeRecompilationCutoff=10000
        -Djdk.attach.allowAttachSelf=true
        -Djdk.nio.maxCachedBufferSize={{
.Values.ezsqlPresto.configMapProp.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 AI Essentials 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 HPE AI Essentials Software to a Hive Metastore in HPE Ezmeral Data Fabric that uses MAPRSASL to authenticate users. In HPE AI Essentials Software, create a Hive data source connection and provide the required connection details.

Prerequisites

If you want HPE AI Essentials Software to authenticate to the Hive Metastore in an external HPE Ezmeral Data Fabric cluster via MAPRSASL, you must provide the Hive connector (in HPE AI Essentials Software) 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

- 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 HPE AI Essentials Software, 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 HPE AI Essentials Software, 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.

Creating the Connection to Hive Metastore in HPE Ezmeral Data Fabric

To create a HPE AI Essentials Software connection to Hive Metastore in HPE Ezmeral Data Fabric that uses MAPRSASL to authenticate users, complete the following steps:

1. Sign in to HPE AI Essentials Software.
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.
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.

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.

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 HPE AI Essentials Software to access the Hive Metastore in HPE Ezmeral Data Fabric, HPE AI Essentials Software must impersonate a user that has permission to access the Hive Metastore. HPE AI Essentials Software 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 HPE AI Essentials Software 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.

Subtopics

Modifying the HPE Ezmeral Data Fabric Configuration Files
Describes how to access and edit the HPE Ezmeral Data Fabric files that HPE AI Essentials Software generates when you configure a data source connection that authenticates to HPE

Ezmeral Data Fabric via MAPRSASL.

Modifying the HPE Ezmeral Data Fabric Configuration Files

Describes how to access and edit the HPE Ezmeral Data Fabric files that HPE AI Essentials Software generates when you configure a data source connection that authenticates to HPE Ezmeral Data Fabric via MAPRSASL.

HPE AI Essentials Software 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 HPE AI Essentials Software 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 AI Essentials Software to connect to CSV and Parquet data in S3-based external data sources.

You can connect HPE AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials 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.
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.

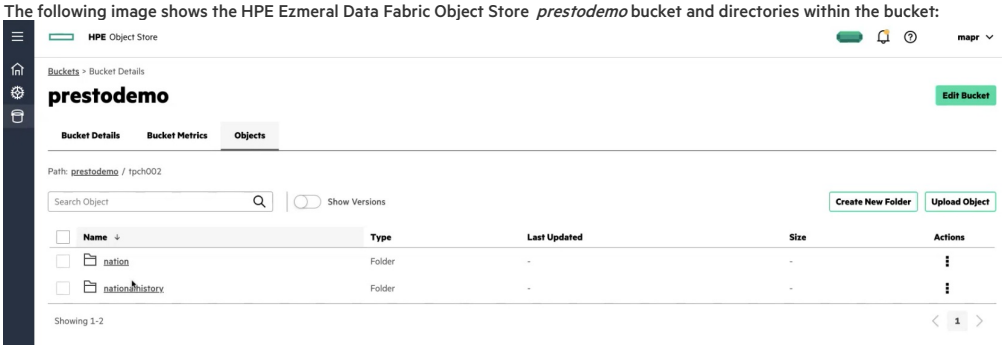
Example: Connect to HPE Ezmeral Data Fabric Object Store to access Parquet files

This example demonstrates how to connect HPE AI Essentials 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:



Creating the HPE AI Essentials 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`

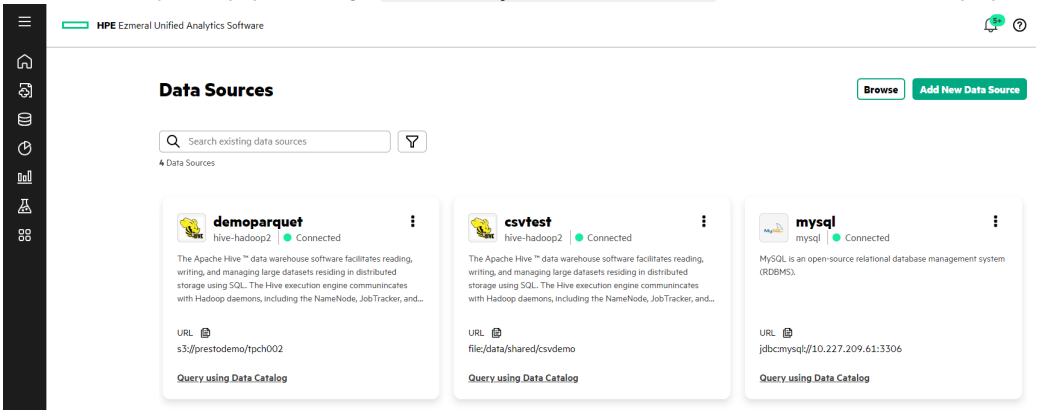
To connect HPE AI Essentials Software to an S3 Directory in an external HPE Ezmeral Data Fabric Object Store:

1. Sign in to HPE AI Essentials 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:

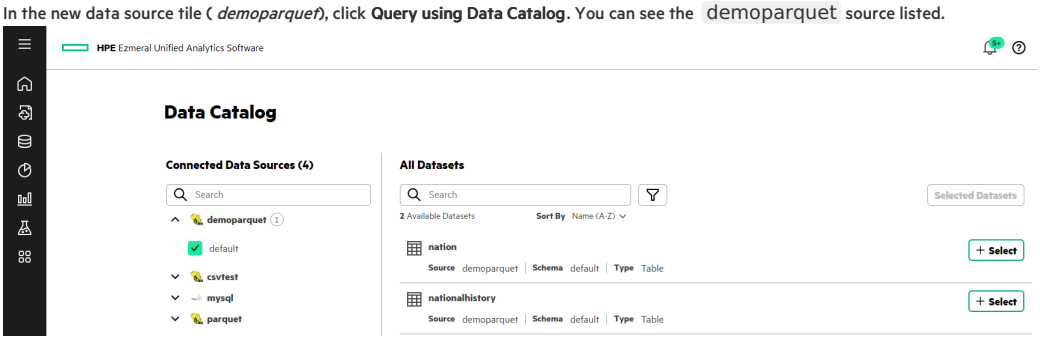
> **TIP**
> - In the drawer, search for and add the fields you do not see. Use the search field under **Optional Fields** to find and add these fields.
> - No external Hive Metastore is required; HPE AI Essentials Software internally parses and scans the files to get the data types and metadata.

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.



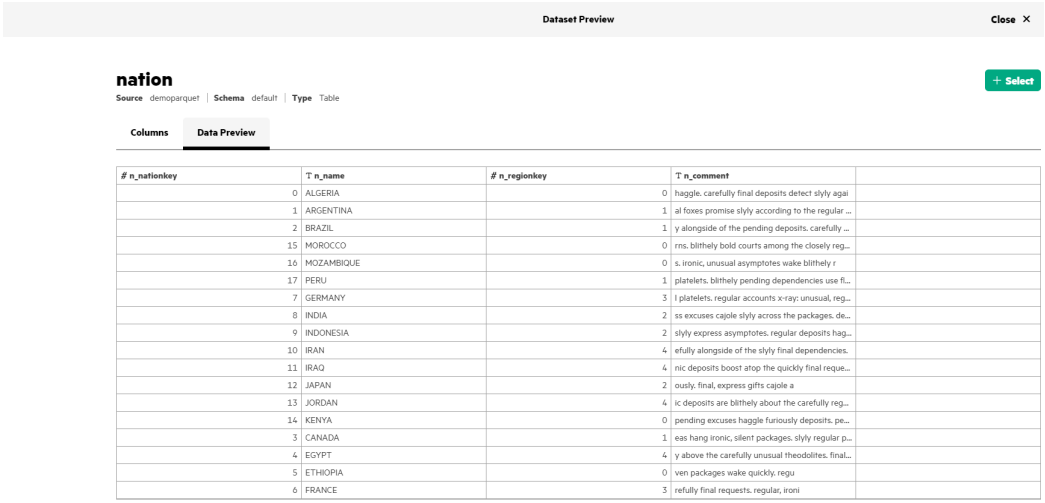
7. In the new data source tile (*demoparquet*), click **Query using Data Catalog**. You can see the `demoparquet` source listed.



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 AI Essentials 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 HPE AI Essentials Software 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 HPE AI Essentials Software 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 HPE AI Essentials Software 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 with the EzPresto Plugin

The EzPresto plugin retrieves and processes vast datasets quickly and efficiently, making it suitable for large-scale data operations. The Spark images packaged with HPE AI Essentials Software include the Presto client library. The Presto client library is required to connect Spark to EzPresto. When you connect Spark to EzPresto, you do not have to explicitly provide a username and password because the bundled EzPresto client library automatically sets and manages credential renewal.

Spark runtimes do not have the EzPresto certificate in the truststore. When you connect Spark to EzPresto, you must set the `ignore_ssl_check` option to `true`.

The following Python and Scala program examples demonstrate how to submit a Spark query through EzPresto that reads data into a DataFrame:
Python

```
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()
```

Scala

```
val query = "select * from mysql.tpch_100gb_std_presto.lineitem"
val 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", query)
val df = dfReader.load()
df.show()
```

```
1 val dfReader = spark.read.format("EzPresto")
2 dfReader.option("presto_url", "https://ezpresto-sts-mst-0.ezpresto-svc-hdl.ezpresto.svc.cluster.local:8081")
3 dfReader.option("dal_url", "https://ezpresto-sts-mst-0.ezpresto-svc-hdl.ezpresto.svc.cluster.local:9090")
4 dfReader.option("ignore_ssl_check", "true")
5 dfReader.option("query", "select * from mysql.tpch_100gb_std_presto.lineitem")
6
7 val df = dfReader.load()
8 df.show()
```

Output

```
dfReader: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@22f3b7bb
res5: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@22f3b7bb
res6: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@22f3b7bb
res7: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@22f3b7bb
res8: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrameReader@22f3b7bb
df: org.apache.spark.sql.DataFrame = [l_orderkey: bigint, l_partkey: bigint ... 14 more fields]
+----------+---------+---------+------------+----------+-----------+-----+---+------------+------------+----------+-----------+-----------+----------+-------
|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|  3221377|   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
```

Using Spark to Query EzPresto via Legacy Method

Spark images packaged with HPE AI Essentials 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 Python and Scala program examples demonstrate how to submit a Spark query through EzPresto that reads data into a DataFrame:
Python

```
DOMAIN = "<your-unified-analytics-domain-name>"
user = "<username>"
catalog = "mysql"
table = "tpch_100gb_std_presto"
uri = f"jdbc:presto://ezpresto.{DOMAIN}:443/{catalog}/{table}"
query = "select * from mysql.tpch_100gb_std_presto.lineitem"
df = spark.read.format("jdbc"). \
    option("driver", "com.facebook.presto.jdbc.PrestoDriver"). \
    option("url", uri). \
    option("user", user). \
    option("SSL", "true"). \
    option("IgnoreSSLChecks", "true"). \
    option("query", query). \
    load().show()
```

```
DOMAIN = "[redacted]
user = "[redacted]
catalog = "mysql"
table = "tpch_100gb_std_presto"
uri = f"jdbc:presto://ezpresto.{DOMAIN}:443/{catalog}/{table}"
query = "select * from mysql.tpch_100gb_std_presto.lineitem"
df = spark.read.format("jdbc"). \
     option("driver", "com.facebook.presto.jdbc.PrestoDriver"). \
     option("url", uri). \
     option("user", user). \
     option("SSL", "true"). \
     option("IgnoreSSLChecks", "true"). \
     option("query", query). \
     load().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
```

Scala

```
val DOMAIN = "<your-unified-analytics-domain-name>"
val user = "<username>"
val catalog = "mysql"
val table = "tpch_100gb_std_presto"
val uri = s"jdbc:presto://ezpresto.$DOMAIN:443/$catalog/$table"
val query = "select * from mysql.tpch_100gb_std_presto.nation"
val df = spark.read.format("jdbc").
     option("driver", "com.facebook.presto.jdbc.PrestoDriver").
     option("url", uri).
     option("user", user).
     option("SSL", "true").
     option("IgnoreSSLChecks", "true").
     option("query", query).
     load().show()
```

```
val DOMAIN = [redacted]
val user = [redacted]"
val catalog = "mysql"
val table = "tpch_100gb_std_presto"
val uri = s"jdbc:presto://ezpresto.$DOMAIN:443/$catalog/$table"
val query = "select * from mysql.tpch_100gb_std_presto.lineitem"
val df = spark.read.format("jdbc").
     option("driver", "com.facebook.presto.jdbc.PrestoDriver").
     option("url", uri).
     option("user", user).
     option("SSL", "true").
     option("IgnoreSSLChecks", "true").
     option("query", query).
     load().show()
```

Output

```
DOMAIN: String = [redacted]
user: String = [redacted]
catalog: String = mysql
table: String = tpch_100gb_std_presto
uri: String = jdbc:presto://ezpresto[redacted]:443/mysql/tpch_100gb_std_presto
query: String = select * from mysql.tpch_100gb_std_presto.lineitem
+----------+--------+--------+------------+----------+---------------+----------+-----+------------+------------+----------+------------+-------------+--------
|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

df: Unit = ()
```

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 AI Essentials 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 AI Essentials 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 AI Essentials Software stores cached data in an HPE Ezmeral Data Fabric volume. You can view and access cached data in the HPE AI Essentials 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 AI Essentials Software cannot access the data that you cache.
- If data in the underlying data sources change, HPE AI Essentials Software does not automatically update the cache. You must cache the data again to refresh the cache.

How to Cache Data

HPE AI Essentials Software must be connected to the data sources with the data sets that you want to cache. See Connecting Data Sources.

To cache data, complete the following steps:

1. Sign in to HPE AI Essentials 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 AI Essentials 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.

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 AI Essentials Software, you can connect to SQL databases and submit queries through EzPresto using the `%sql` or `%%sql` magic. See Notebook Magic Functions%sql and %%sql

Data Lakehouse Gateway

Describes Data Lakehouse Gateway for accessing, managing, and governing the data stored in Data Lakehouses.

Data Lakehouse Gateway is a feature in HPE Private Cloud AI that enables administrators to access, manage, organize, and govern enterprise data stored in a modern Data Lakehouse architecture through the HPE AI Essentials Software UI.

Data Lakehouse Gateway securely connects HPE AI Essentials Software to data lakehouses through a global namespace. The global namespace unifies data, regardless of where the data is located, making it easy to access through a variety of tools including SQL, Apache Spark, Superset, and AI/ML tools.

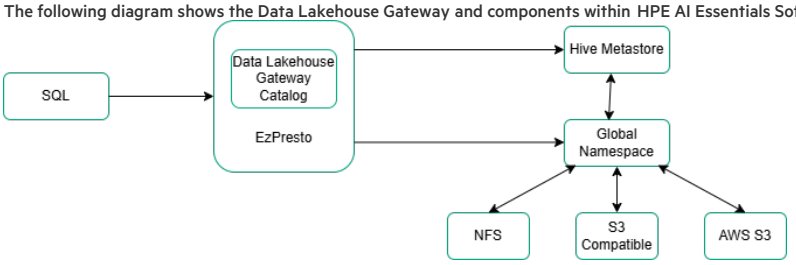
Currently, Data Lakehouse Gateway can connect to the following data lakehouses:

- NFSv4
- S3-Compatible

After an administrator adds a data lakehouse to the global namespace (creates a connection), the administrator can register tables and manage permissions on them. Data Lakehouse Gateway supports Apache Iceberg tables and Open Table format. Users permitted to access the tables can run queries against them and use them in their AI/ML workloads.

Data Lakehouse Gateway Architecture

The following diagram shows the Data Lakehouse Gateway and components within HPE AI Essentials Software that enable access to data in data lakehouses:



The following list describes the components in the diagram:
SQL (client, tool, application)

An example of a client or tool that can access registered tables in HPE AI Essentials.

Data Lakehouse Gateway

Data Lakehouse Gateway is a pre-configured EzPresto Iceberg catalog designed to access Iceberg tables stored on AWS S3, NFS, and S3-compatible sources. These tables are accessible through a global namespace. Currently, the Data Lakehouse Gateway Catalog only supports the read-only access.

EzPresto

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.

Hive Metastore

The central metadata repository in Data Lakehouse Gateway that simplifies managing and accessing Iceberg tables. When you register a table, a metadata entry is created in the Hive Metastore, generating new metadata alongside any existing metadata. Unregistering a table removes the metadata entry.

Global Namespace

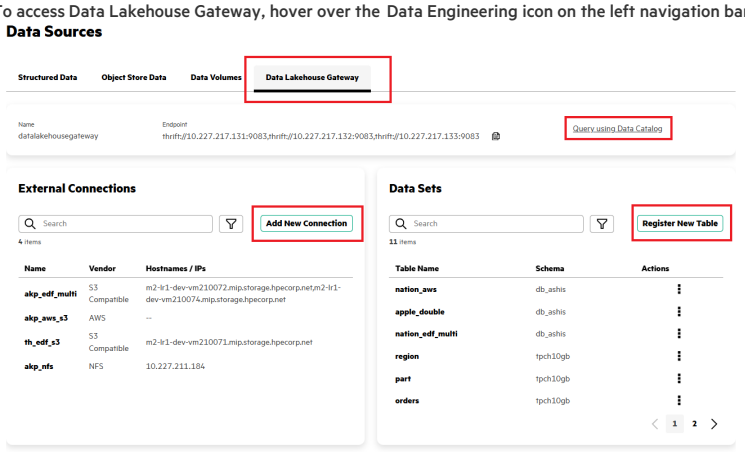
Software that enables data lakehouse servers to mount Data Lakehouse Gateway to create one unified data plane. The global namespace unifies data lakes regardless of their physical location. For example, servers can be in different countries, in the cloud, on-premises, or a hybrid of the two.

NFS, S3-Compatible

Data lakehouses that you can mount to Data Lakehouse Gateway making them part of the global namespace. NFS requires write access with the UID/GID set to 2999/2999. For S3, you must provide write permissions for the specified credentials (access key and secret key).

Accessing Data Lakehouse Gateway

To access Data Lakehouse Gateway, hover over the Data Engineering icon on the left navigation bar and select the Data Sources → Data Lakehouse Gateway tab.



In the Data Lakehouse Gateway screen, you can see the following details:

- Name
- Endpoints

Here, you can add a connection to your data lakehouses and register tables. You can also select the data sources, schemas, and data sets you want to work with by clicking the Query using Data Catalog link.

Connecting Data Lakehouses

To connect external data lakehouses, configure external connections in the External Connections section of the Data Lakehouse Gateway screen. Once the connection is setup, you can register tables to query data catalogs. To learn how to add a new connection, see Connecting Data Lakehouses.

Managing Tables in Data Lakehouse Gateway

You can perform the following operations using Data Lakehouse Gateway:

Operations	Details
Register	Once your connection is setup, register a table with an existing schema or with a new schema.
Refresh	Refresh registered tables in the Data Lakehouse Gateway to reflect any changes made to the source table.
Unregister	Delete registered tables from the Data Lakehouse Gateway.

To learn more, see Managing Tables in Data Lakehouse Gateway.

Managing Access to Data Lakehouse Gateway

As a Private Cloud AI Administrator, you have unrestricted access to all registered tables in the Data Lakehouse Gateway. You can manage and grant access to these tables for Private Cloud AI User. To learn more, see Data Lakehouse Gateway Access.

Using Data Lakehouse Gateway

To learn how to use Data Lakehouse Gateway, see Using Tables Registered via Data Lakehouse Gateway .

Troubleshooting

To view the troubleshooting information for Data Lakehouse Gateway, see Data Lakehouse Gateway Troubleshooting.

Limitations

Currently, Data Lakehouse Gateway does not support the following:

- AVRO files
- Caching
- Multiple IPs for NFS
- Deleting external connections

Subtopics

Connecting Data Lakehouses
Describes how to connect data lakehouses using Data Lakehouse Gateway.

Managing Tables in Data Lakehouse Gateway
Describes how to register, refresh, and unregister tables in Data Lakehouse Gateway.

Using Tables Registered via Data Lakehouse Gateway
Describes how to use tables registered via Data Lakehouse Gateway to query data in HPE AI Essentials Software.

Connecting Data Lakehouses

Describes how to connect data lakehouses using Data Lakehouse Gateway.

Prerequisites

Private Cloud AI Administrator access to HPE AI Essentials Software.

About this task

You can connect to your external data lakehouses using the Data Lakehouse Gateway
Data Lakehouse Gateway connects the HPE AI Essentials Software to data lakehouses through a global namespace.

Currently, Data Lakehouse Gatewaycan connect to the following data lakehouses:

- AWS S3
- NFSv4
- S3 Compatible

To add a new connection, follow these steps:

Procedure

1. Click the Add New Connection button
2. Set the following boxes:
   a. Name: Enter a name of the connection.
   b. Vendor: Choose one of the following vendors:

Vendor	Configuration
AWS	Access Key Enter access key. Secret key Enter secret key. Region Enter the region that contains the bucket. For example, `us-east-1`.
S3 Compatible	Access Key Enter access key. Secret key Enter secret key. Hostnames/ IPs Enter multiple hostnames or IPs separated by comma. S3 Server Transfer Protocol Choose one: • https • http S3 Server Port Enter the S3 server port number. Use TLS encryption To use the TLS encryption, toggle on Use TLS encryption. Browse and select your CA certificate file. Note that if your S3 server is CA certified, you do not need to upload the certificate.
NFS	Hostnames/ IPs Enter a single hostname or IP.

3. To add a new connection, click Submit. To discard changes, click Cancel.

> **NOTE**
> You cannot delete an external connection after adding it.

Results

You can view the list of configured external connections in the External Connections section of the Data Lakehouse Gateway screen. Once the connection is setup, you can register tables

to query data catalogs. See Managing Tables in Data Lakehouse Gateway.

Managing Tables in Data Lakehouse Gateway

Describes how to register, refresh, and unregister tables in Data Lakehouse Gateway.

Prerequisites: Private Cloud AI Administrator access to HPE AI Essentials Software.

You can perform the following operations using Data Lakehouse Gateway:

- Registering New Table
- Refreshing Table
- Unregistering Table

Registering New Table

You can register a table with an existing schema or with a new schema.

1. Click the Register New Table button.
2. Choose either Existing Schema or New Schema. Set the following boxes:

Schema	Configuration
Schema	Enter a new schema name for the New Schema option. For the Existing Schema option, select a schema name from the list.
Table Name	Enter a table name.
Metadata File Path	Enter the metadata file path.

Entering the Metadata File Path

To specify the metadata file path, perform the following steps. The path can either be an S3 path or an NFS path.

AWS

To specify the metadata file path for AWS, follow these steps:

a. Locate the metadata directory for your Iceberg table stored in S3.

b. Select the metadata directory.

c. Click the Copy S3 URI button to copy the directory path.

d. Update the copied path to match the name of the external connection that you previously created. The modified path must follow this format:

```
s3://<external-connection-name>-<s3-bucket-name>/<path-to-the-table-metadata-location>/
```

For example:

- Name of your AWS external connection: `awsdlhg`
- Copied AWS metadata file path: `s3://tpch/tables/tpchtiny/nation/metadata/`

In this case, the metadata file path is:

```
s3://awsdlhg-tpch/tables/tpchtiny/nation/metadata/
```

S3 Compatible

To specify the metadata file path for the Data Fabric or MinIO S3, follow this format:

```
s3://<external-connection-name>-<bucket-name>/<table-path>
```

Replace <external-connection-name>, <bucket-name>, and <table-path> with the appropriate values for your S3.

NFS

To specify the metadata file path for NFS, follow this format:

```
file:/<external-connection>/<path-to-table-metadata-location>
```

Replace the <external-connection> and <path-to-table-metadata-location> with the appropriate values based on your external connection and metadata location.

For example:

- Name of your NFS external connection: `nfs_external`
- Metadata location path: `/iceberg/data/tpch/tiny/nation/metadata`

In this case, the metadata file path is:

```
file:/nfs_external/iceberg/data/tpch/tiny/nation/metadata
```

3. To register a new table, click Register. To discard all changes, click Cancel.

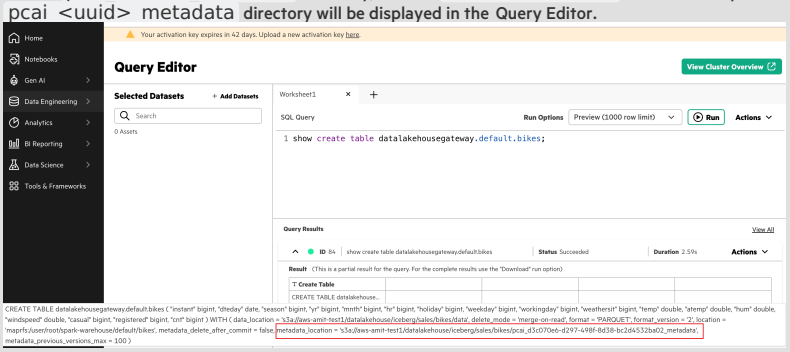
Results:

You can view the newly registered table in the Data Sets section of the Data Lakehouse Gateway screen.

When you register your table, the HPE AI Essentials creates a new metadata directory named `pcai_<uuid>_metadata` at the source table location. The Data Lakehouse Gateway uses this directory to perform data operations on the registered table.

TIP

After registering a table, you must note the `pcai_<uuid>_metadata` directory associated with the registered table. To identify the correct `pcai_<uuid>_metadata` directory, run the `show create table` command for your table. After running the command, the `pcai_<uuid>_metadata` directory will be displayed in the Query Editor.



Refreshing Table

You can refresh registered tables in the Data Lakehouse Gateway to reflect any changes made to the source table, such as inserting new data, updating current data, or deleting any rows, among other operations.

To refresh the table, follow these steps:

1. Click the Actions menu for the table you want to refresh.
2. Select Refresh and follow the prompt.

Results:

HPE AI Essentials refreshes your table and fetches the updated data from the source table.

NOTE

While the table is being refreshed, any read queries on the table will fail.

Unregistering Table

You can delete registered tables from the Data Lakehouse Gateway using the Unregister option.

To unregister the table, follow these steps:

1. Click the Actions menu for the table you want to delete.
2. Select Unregister and follow the prompt.

Results:

HPE AI Essentials deletes your table from the Data Lakehouse Gateway and deletes the metadata directory named `pcai_<uuid>_metadata` at the source table location.

Using Tables Registered via Data Lakehouse Gateway

Describes how to use tables registered via Data Lakehouse Gateway to query data in HPE AI Essentials Software.

Once a table is registered, you can access, query data, and view query results. Perform the following steps:

- Accessing Data Catalog
- Viewing Connected Data Sources
- Searching and Selecting Data
- Running Queries
- Viewing Query Results
- Viewing Cluster Overview

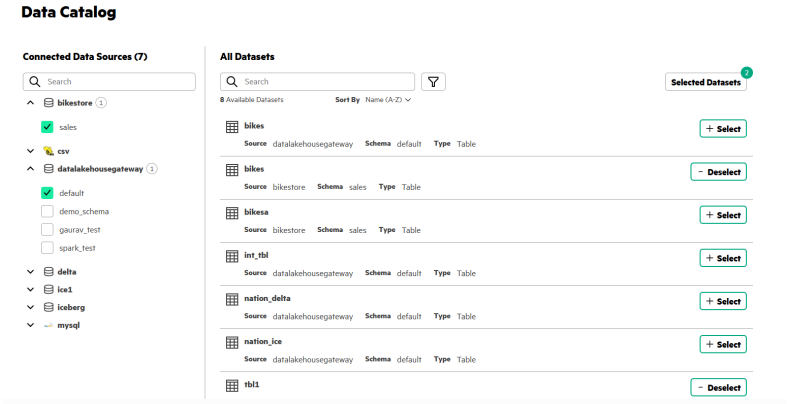
Accessing Data Catalog

After a table is registered, you can access the data with appropriate permissions.

To navigate to the Data Catalog screen, click the Query using Data Catalog link.

Viewing Connected Data Sources

Under Connected Data Sources, you can view a list of available data sources.



These data sources can include:

- Structured data
- Data Lakehouse Gateway data

Expand any connected data source to view and select its schema.

Once a schema is selected, the associated tables and views will appear under All Datasets.

Searching and Selecting Data

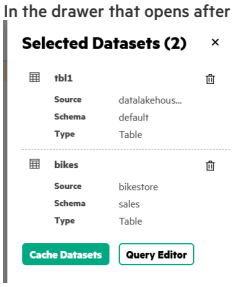
If a schema contains many tables or views, use the search function to locate specific tables or views by name. If you have selected multiple schemas, you can also filter results by schema.

Identify the tables and views you want to work with and click **Select** next to each item of your choice.

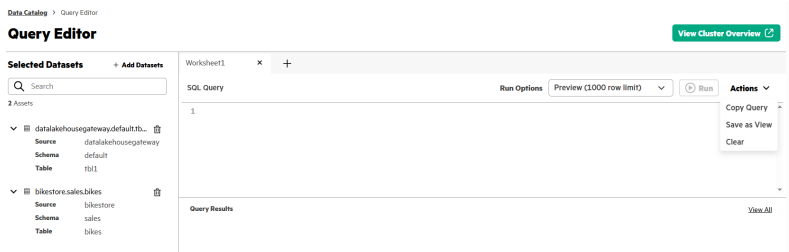
When you are done selecting, click **Selected Datasets**.

Running Queries

In the drawer that opens after clicking Selected Datasets,



1. Click Query Editor to access the workspace for running queries.

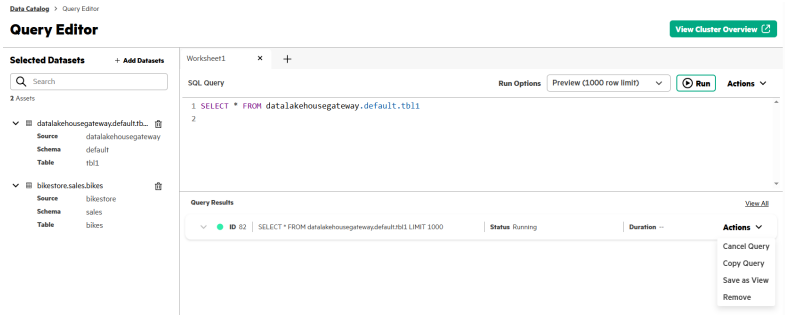


2. Enter your query into the worksheet and click Run. The results will appear under Query Results.
3. To download query results, select Download in the Run Options field, then click Run.
4. To manage a query, perform the following actions:

Actions	Descriptions
Copy Query	Copies the query to your clipboard.
Save as View	Saves the query as a view for future use.
Clear	Clear the query from workspace.

Viewing Query Results

Once your query runs, view it on the Query Results section.



To view a log of all queries, click the View All link on the Query Results section.

You can perform the following actions on queries using the Actions column:

Actions	Descriptions
Cancel Query	Cancel the current query run.
Copy Query	Copies the query to your clipboard.
Save as View	Saves the query as a view for future use.
Remove	Deletes the query from the results list.
Open in Query Editor	Open query in the Query Editor screen.
View Details	Displays session information and resource utilization for the query.

Viewing Cluster Overview

To see details for all queries that have been run, click the **View Cluster Overview** button. Here, a Private Cloud AI Administrator can access details for all users' queries, while a Private Cloud AI User can only view their own query details.

More information

- Connecting Data Sources
- Using Spark to Query EzPresto
- Connecting External Applications to EzPresto via JDBC
- Data Source Connectivity and Exploration

Airflow

Provides an overview of Apache Airflow in HPE AI Essentials 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 AI Essentials 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 AI Essentials 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.

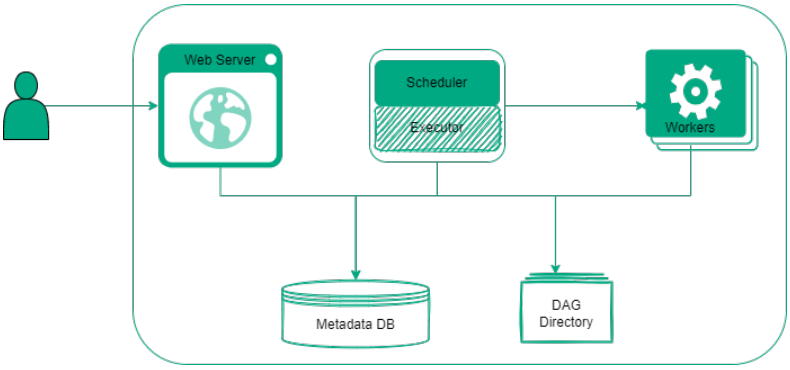
Airflow Cluster

Deploys the UI and scheduler components of Airflow.

In HPE AI Essentials 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 AI Essentials 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**.

Subtopics

Airflow DAGs Git Repository
Describes how HPE AI Essentials Software reads DAGs and how to configure a GitHub repository in Airflow.
Configuring Airflow
Describes how to configure Airflow in HPE AI Essentials Software.
Submitting Spark Applications by Using DAGs
Describes how to submit the Spark applications by using DAGs in Airflow.
Defining RBACs on DAGs
Describes role-based access controls (RBACs) with respect to Airflow in HPE AI Essentials Software and how to define RBACs to permit access to DAGs.

Airflow DAGs Git Repository

Describes how HPE AI Essentials 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 AI Essentials Software supports both private and public GitHub repositories. HPE AI Essentials 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.

Configuring a Git Repository for Airflow

To configure Airflow with the GitHub repository where DAGs are stored:

1. Sign in to HPE AI Essentials Software as Administrator.
2. In the left navigation bar, click Tools & Frameworks.
3. On the **Airflow** tile, click the three-dots menu and then select Configure. The YAML file editor opens.

4. In the editor, find the `git:` section.
5. Configure the following parameters in the `git:` section:
   repo:

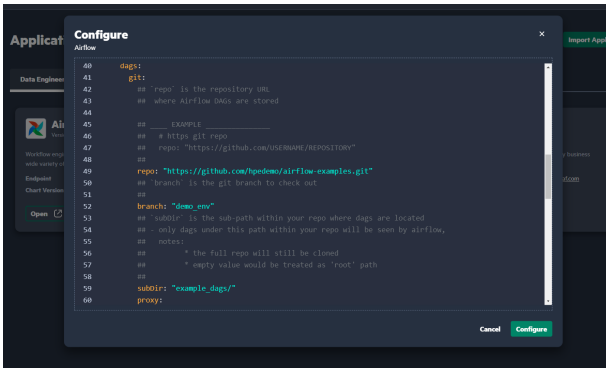
   The repository URL for private or public Git repository which stores the DAGs.

   branch:

   The name of the branch within the repository to use.

   subDir:

   The path to the directory where the DAGs are located.



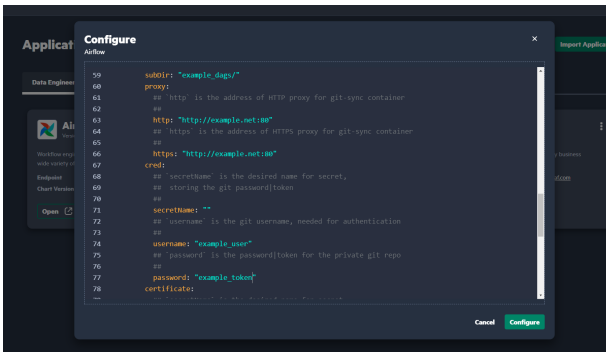
   If the git repository is private, configure the following fields:
   username

   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.



6. Click Configure and wait until Airflow is configured.

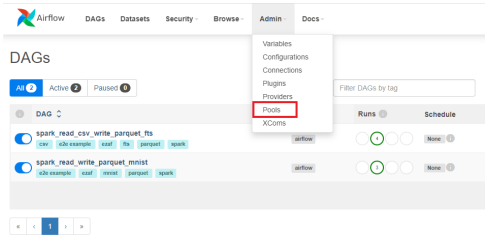
Configuring Airflow

Describes how to configure Airflow in HPE AI Essentials 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 AI Essentials Software.
2. Click the Tools & Frameworks icon on the left navigation bar.
3. Click Open on Airflow.
4. Click Admin and select Pools.



5. Click Edit Record and update the value of Slots.

6. 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.

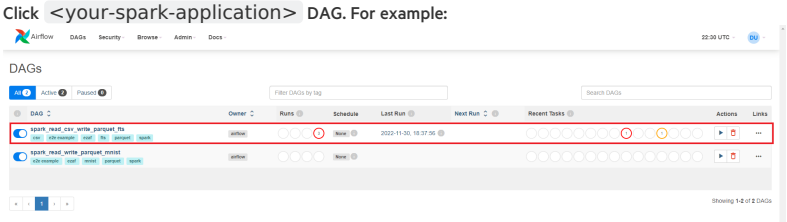
> **NOTE**
>
> If you do not have the repository to store Airflow DAGs, request an administrator to configure the Git repository now. For details, see Airflow DAGs Git Repository.

3. After your DAG is available in the Git repository, sign in to HPE AI Essentials Software as a member.

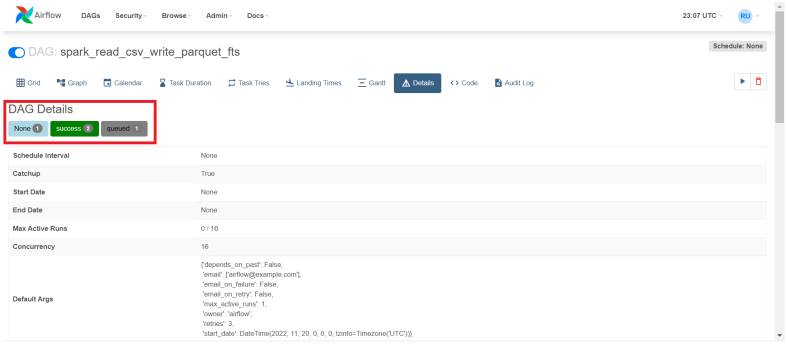
About this task

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 and click Open on Airflow.
2. In **Airflow**, verify that you are on the DAGs screen.
3. Click `<your-spark-application>` DAG. For example:

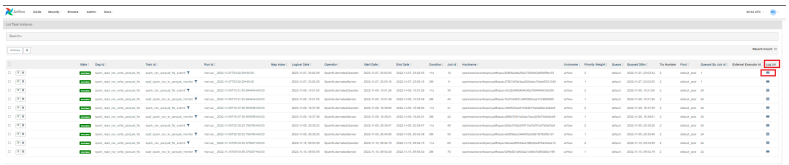


4. Click Code to view the DAG code.
5. Click Graph to view the graphical representation of the DAG.
6. Click **Run** (play button) and select **Trigger DAG w/ config** to specify the custom configuration.
7. To run a DAG after making configuration changes, click Trigger.
8. To view details for the DAG, 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.



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"
```



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

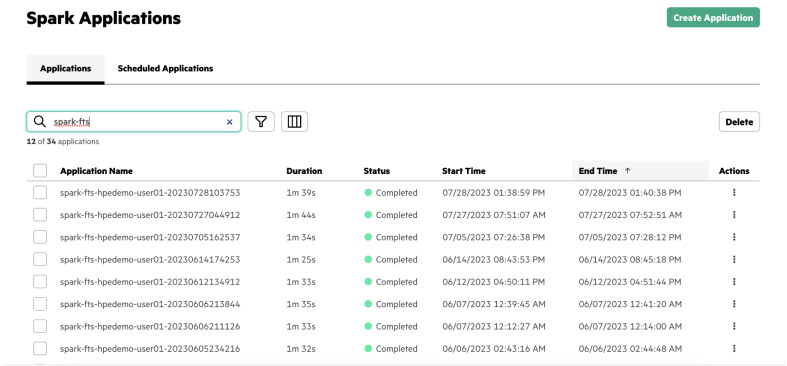
> **IMPORTANT**
>
> The cluster clears the logs that result from the DAG runs. The duration after which the cluster clears the logs depends on the Airflow task, cluster configuration, and policy.

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 click Open on Spark Operator.



Defining RBACs on DAGs

Describes role-based access controls (RBACs) with respect to Airflow in HPE AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials Software to give the user full access to DAGs; however, this action must occur before the user signs in to the HPE AI Essentials Software UI and accesses Airflow. See **User Roles**.
- If an admin removes a user from HPE AI Essentials 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 AI Essentials Software UI. Role changes made in HPE AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials Software. Adding a user to HPE AI Essentials Software after you define roles on the user (username) in the DAG constructor will not cause any issues between the systems. An HPE AI Essentials Software admin can add or create the user.

IMPORTANT

The DAG must exist in the GitHub repository or a Git submodule that the Airflow instance in HPE AI Essentials 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 AI Essentials 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 AI Essentials Software.
2. In the left navigation bar, select Tools and Frameworks.
3. Click Open on Airflow.
4. In **Airflow**, click the Security tab and select **List Roles**.

Superset

Provides a brief overview of Superset in HPE AI Essentials 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 AI Essentials Software by going to BI Reporting > Dashboards or Tools & Frameworks in the left navigation panel.

When you connect HPE AI Essentials 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, the HPE AI Essentials 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 AI Essentials Software. The unified data source connection enables you to:

- Add the data sets you create in the HPE AI Essentials 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 AI Essentials Software).

Refer to the following tutorials to get started with Superset in HPE AI Essentials Software:

- BI Reporting (Superset) Basics
- Retail Store Analysis Dashboard (Superset)

For additional information about Superset, see **Apache Superset** and EzPresto.

Subtopics

Defining RBACs in Superset
Describes role-based access controls (RBACs) with respect to Superset in HPE AI Essentials Software and how to define RBACs to permit access to Superset dashboards.
Configuring Horizontal Pod Autoscaling (HPA)
Describes how to configure HPA for Superset in HPE AI Essentials Software.

Defining RBACs in Superset

Describes role-based access controls (RBACs) with respect to Superset in HPE AI Essentials 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 AI Essentials Software maps the HPE AI Essentials 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 AI Essentials 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. Additionally, if the database connection is EzPresto, see User Access to Data to grant the appropriate privileges to the member user.

- 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 AI Essentials Software UI. Role changes made in HPE AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials Software.
2. In the left navigation bar, go to **Tools & Frameworks**.
3. Click Open on Superset.
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 AI Essentials Software.
2. In the left navigation bar, go to **Tools & Frameworks**.
3. Click Open on Superset.
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 AI Essentials Software.
2. In the left navigation bar, go to **Tools & Frameworks**.
3. Click Open on Superset.
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 AI Essentials Software.

HPE AI Essentials 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`:

- The HPA configuration with autoscaling enabled for `supersetWorker`:

Data Analytics

Provides a brief overview of data analytics in HPE AI Essentials Software.

HPE AI Essentials 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 AI Essentials Software simplifies data access and data workflows and pipelines. HPE AI Essentials 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.

Subtopics

Spark
Provides a brief overview of Apache Spark in HPE AI Essentials Software.

More information

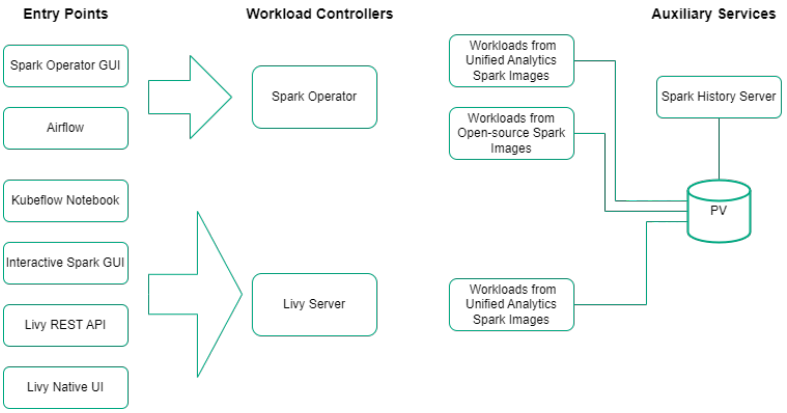
- Get Started

Spark

Provides a brief overview of Apache Spark in HPE AI Essentials 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 AI Essentials Software, there are two controllers for running Spark workloads. These controllers are Spark Operator and Livy server.



HPE AI Essentials 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.

To see the list of the Spark images distributed by HPE AI Essentials Software, see List of Spark Images.

Livy server uses the Rest API and Spark images (supporting Data Fabric services) provided by HPE AI Essentials Software to submit the Spark applications. To learn about the supported version of Spark, see Support Matrix.

> **NOTE**
> Livy does not support Spark OSS images or your own open-source Spark images on HPE AI Essentials Software.

Features and Functionality

HPE AI Essentials Software provides an enterprise-ready, unified Spark experience that supports an Apache Livy-based interactive sessions..

Spark in HPE AI Essentials 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.
- Run Spark jobs from HPE AI Essentials Software using the following components:
  - Spark Operator: The following are entry points for the Spark Operator:
    - Airflow
    - Spark Operator GUI in HPE AI Essentials Software. See Creating Spark Applications.
  - 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.
    - Interactive Spark Sessions GUI available in HPE AI Essentials Software. See Creating Interactive Sessions.
    - 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 AI Essentials 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.

Subtopics

Using Spark Images
Describes different types of Spark images supported by HPE AI Essentials Software.
List of Spark Images
Lists the Spark images distributed by HPE AI Essentials Software.
Creating Spark Applications
Describes how to create and submit Spark applications using HPE AI Essentials Software.
Managing Spark Applications
Describes how to view and manage Spark applications using HPE AI Essentials Software.
Configuring Memory for Spark Applications
Describes how to set memory options for Spark applications.
Creating Interactive Sessions
Describes how to create interactive sessions in HPE AI Essentials Software.
Submitting Statements

Describes how to submit statements in HPE AI Essentials Software.

Managing Interactive Sessions

Describes how to view and manage Spark interactive sessions in HPE AI Essentials Software.

Spark History Server

Provides an overview of Spark History Server.

Using Spark SQL API

Describes how to use Spark SQL API in HPE AI Essentials Software.

Enabling GPU Support for Spark

Describes NVIDIA `spark-rapids` accelerator support for Spark, and how to enable and allocate the GPU resources on Spark.

Securely Passing Spark Configuration Values

Describes how to pass the sensitive data to Spark configuration using the Kubernetes Secret.

Running Spark Applications in Namespaces

Describes how namespaces work with regard to Spark applications in HPE AI Essentials Software.

Using Spark Images

Describes different types of Spark images supported by HPE AI Essentials Software.

HPE AI Essentials Software packages two different types of images:

1. HPE-curated Spark images. For details, see Using HPE-Curated Spark Images.
2. Spark Open-Source Software (OSS) images (Spark OSS images). For details, see Using Spark OSS Images.

The following table compares the two different types of Spark images packaged with HPE AI Essentials 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 AI Essentials Software. See Using Your Own Open-Source Spark Images.

Subtopics

Using HPE-Curated Spark Images

Describes how to use HPE-curated Spark images to submit Spark applications.

Using Spark OSS Images

Describes how to use Spark Open-Source Software (OSS) images to submit Spark applications.

Using Your Own Open-Source Spark Images

Describes how to use your own open-source Spark images to submit Spark applications.

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.

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.
- Spark Operator workflow using Airflow. See Using Airflow.
- Livy workflow using the Spark Interactive Sessions GUI. See Using the Spark Interactive Sessions GUI.
- Livy workflow using Jupyter Notebooks. See Using Notebooks.

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 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.

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.
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 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.

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 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 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.

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.

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. Spark OSS images also do not support Data Fabric-specific security features (data-fabric SASL (`maprsasl`)).

You can use Spark OSS images in either of the following workflows:

- Spark Operator workflow using the Create Spark Application GUI. See Using the Create Spark Application GUI.
- Spark Operator workflow using Airflow. See Using Airflow.

Regardless of which workflow you follow, you must add the following environment variables and properties to the Spark YAML for the Spark application to successfully access data in local-S3 and external S3 object storage:

Add the env map as a sub-map to both the driver and executor maps as shown in the following Spark YAML example:

```
driver:
    labels:
      version: ${SPARK_VERSION}
    cores: 1
    env:
      - name: AWS_ACCESS_KEY_ID
        valueFrom:
          secretKeyRef:
            key: AUTH_TOKEN
            name: access-token
      - name: AUTH_TOKEN
        valueFrom:
          secretKeyRef:
            key: AUTH_TOKEN
            name: access-token
```

```
    coreLimit: "1"
    memory: 512m
  executor:
    labels:
      version: ${SPARK_VERSION}
    cores: 1
    env:
      - name: AWS_ACCESS_KEY_ID
        valueFrom:
          secretKeyRef:
            key: AUTH_TOKEN
            name: access-token
      - name: AUTH_TOKEN
        valueFrom:
          secretKeyRef:
            key: AUTH_TOKEN
            name: access-token
    coreLimit: "1"
    memory: 512m
    instances: 1
```

Add the following properties to the sparkConf map in the Spark YAML as shown in the following example:

```
sparkConf:
    spark.eventLog.dir: file:///opt/mapr/spark/sparkhs-eventlog-storage
    spark.eventLog.enabled: "true"
    spark.executorEnv.AWS_SECRET_ACCESS_KEY: s3
    spark.executorEnv.SPARK_USER: my-user-name
    spark.kubernetes.driverEnv.AWS_SECRET_ACCESS_KEY: s3
    spark.kubernetes.driverEnv.SPARK_USER: my-user-name
    spark.mapr.user.secret: hpe-autotix-generated-secret
    spark.mapr.user.secret.autogen: "true"
```

Using the Create Spark Application GUI

To use Spark OSS images, choose one of the following options in the GUI:
Using Upload YAML in GUI

1. Select the Spark OSS image from the List of Spark Images.
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.
4. Perform the instructions to create a Spark application as described in Creating Spark Applications 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.

Using New application in GUI

1. Perform the instructions to create a Spark application as described in Creating Spark Applications 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.
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.
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.

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 .

Subtopics

Setting the User Context
Describes how to set the user context when using the Spark OSS images.

Setting the User Context

Describes how to set the user context when using the Spark OSS images.

When you set the user context for a Spark application, you specify the user identity under which the Spark application or its tasks are executed.

By default, when you run a Spark application, the application interacts with the cluster and cluster resources on your behalf, having the same security privileges as you. However, you may want to change the user context (user identity) of a Spark application to change the application's privileges. For example, you may want to limit the Spark application's access to certain data.

To set the user context, add podSecurityContext to the driver and executor maps in the Spark YAML, as shown in the following example:

NOTE
If you are using the default Spark images (HPE-curated Spark images), do not add this configuration to your Spark YAML.

```
autoconfigure oss spark
apiVersion: sparkoperator.hpe.com/v1beta2
kind: SparkApplication
metadata:
  annotations:
    description: ""
  creationTimestamp: 2025-05-16T16:20:08Z
  generation: 2
  labels:
    hpe-ezua/app: spark
    hpe-ezua/type: app-service-user
    sidecar.istio.io/inject: "false"
  name: test
  namespace: user
  resourceVersion: "5464676"
  uid: e0ec9dea-4537-4fa5-b5f2-9fd9fd80ebd6
spec:
  arguments:
    - file:///mounts/shared-volume/shared/aie-tutorials/current-release/Data-
Analytics/Spark/wordcount/wordcount.txt
  driver:
    coreLimit: "1"
    cores: 1
    labels:
      version: 3.5.2
    memory: 4G
    podSecurityContext:
      runAsGroup: 1001
      runAsUser: 10000003
      supplementalGroups:
        - 0
    serviceAccount: spark-runner
    volumeMounts:
      - mountPath: /mounts/shared-volume/user
        name: upv1
      - mountPath: /mounts/shared-volume/shared
        name: pv1
      - mountPath: /opt/mapr/spark/sparkhs-eventlog-storage
        name: sparkhs-eventlog-storage
  executor:
    coreLimit: "1"
    cores: 1
    instances: 1
    labels:
      version: 3.5.2
    memory: 2G
    podSecurityContext:
      runAsGroup: 1001
      runAsUser: 10000003
      supplementalGroups:
```

```
      - 0
  volumeMounts:
    - mountPath: /mounts/shared-volume/user
      name: upv1
    - mountPath: /mounts/shared-volume/shared
      name: pv1
 image: 172.17.16.80/ezmeral-common/hpe-spark/spark:v3.5.2.3.0
 imagePullPolicy: Always
 imagePullSecrets:
  - imagepull
 mainApplicationFile: local:///mounts/shared-volume/shared/aie-tutorials/current-
release/Data-Analytics/Spark/wordcount/wordcount.py
 mode: cluster
 restartPolicy:
  type: Never
 sparkConf:
  spark.eventLog.dir: file:///opt/mapr/spark/sparkhs-eventlog-storage
  spark.eventLog.enabled: "true"
  spark.executorEnv.SPARK_USER: user
  spark.hpe.webhook.security.context.autoconfigure: "true"
  spark.kubernetes.driverEnv.SPARK_USER: user
  spark.mapr.user.secret: hpe-autotix-generated-secret-njrvat
  spark.mapr.user.secret.autogen: "true"
 sparkVersion: 3.5.2
 type: Python
 volumes:
  - name: upv1
    persistentVolumeClaim:
      claimName: user-pvc
  - name: pv1
    persistentVolumeClaim:
      claimName: kubeflow-shared-pvc
  - name: sparkhs-eventlog-storage
    persistentVolumeClaim:
      claimName: private-cloud-a-838eea37-sparkhs-pvc
```

To learn more about pod security context, see **Configure a Security Context for a Pod or Container**.

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 AI Essentials 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 AI Essentials Software. See **Building Images**.
3. Choose one of the following:
   - Using the Create Spark Application GUI
   - Using Airflow

Using the Create Spark Application GUI

To use your own open-source Spark images, choose one of the following options 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.

3. Perform the instructions to create a Spark application as described in Creating Spark Applications 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.

Using New application

1. Perform the instructions to create a Spark application as described in Creating Spark Applications 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.
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.

List of Spark Images

Lists the Spark images distributed by HPE AI Essentials Software.

The images follow the following format:

```
<base-repository>/<image-name>:<image-tag>
```

HPE AI Essentials 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:

```
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-spark/livy-
0.8.0:202501151114R
```

Non-timestamped image tags

These image tags are dynamic, and they are updated when new changes are pushed to the image.

For example:

```
lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-spark/spark-
3.5.2:v3.5.2.1.1
```

Creating Spark Applications

Describes how to create and submit Spark applications using HPE AI Essentials 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.

> **NOTE**
>
> You may want to read about Spark images before creating a Spark application. See Using Spark Images.

Create a Spark Application

Complete the following steps to create and submit a Spark application:
1. Sign in to HPE AI Essentials 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 AI Essentials Software screen.
   - Click the Tools & Frameworks icon on the left navigation bar and click Open on Spark Operator.

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.

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 AI Essentials 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 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. 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: 1 • Memory: 512m
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. 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: 1 • Memory per Executor: 512m

Steps	Instructions
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, s), 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.

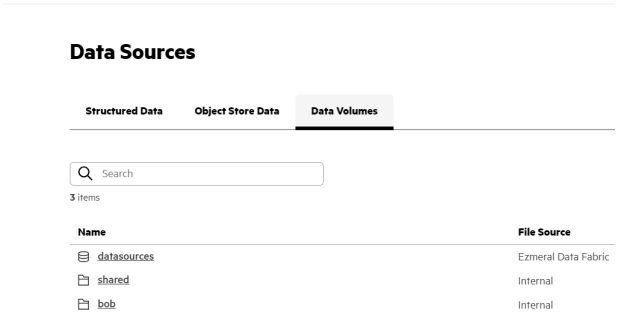
Selecting the Location of the Main Application File

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 AI Essentials 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:



   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 HPE AI Essentials Software, 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 .

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.
- 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 and Configuring a Spark Application to Access Data in an External S3 Data Source through the S3 Proxy Layer.

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
```

Subtopics

Configuring a Spark Application to Access External S3 Object Storage
Describes configuration options for connecting Spark to external S3 object storage.

More information

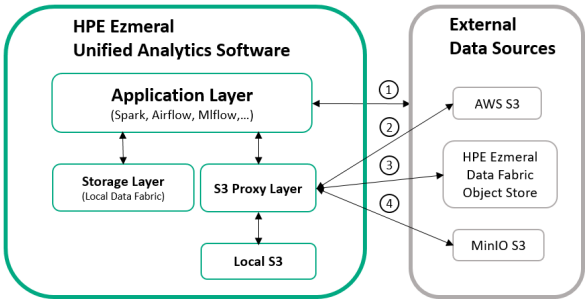
- Configuring a Spark Application to Access External S3 Object Storage

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 AI Essentials Software.

The following diagram shows how applications in HPE AI Essentials Software 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.

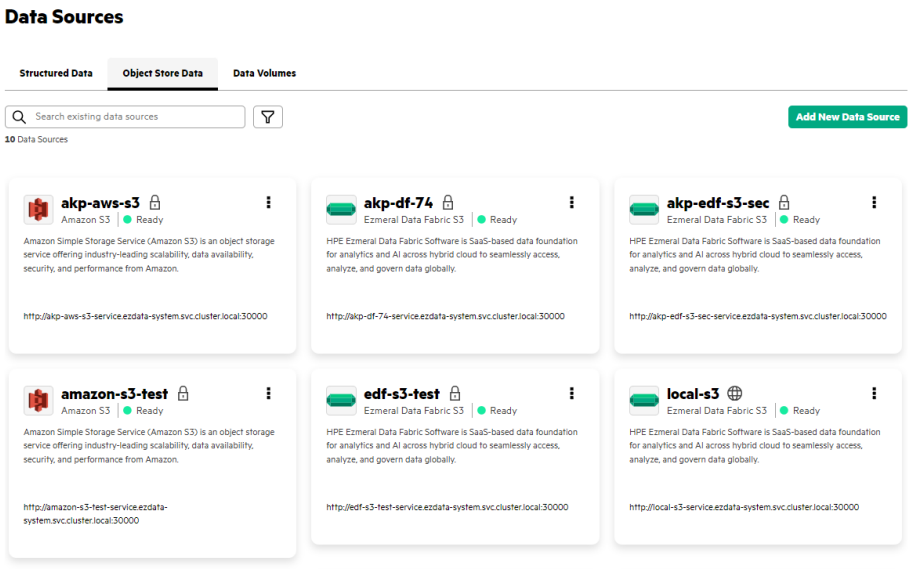


The S3 proxy layer securely connects HPE AI Essentials Software to external data sources, such as AWS S3, MinIO S3, and HPE Ezmeral Data Fabric Object Store.

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 HPE AI Essentials Software 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 HPE AI Essentials Software.

You can see the external S3 data sources that your administrator configured for you in the HPE AI Essentials Software 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.



The following topics describe each of the methods (direct or S3 proxy) for connecting Spark to an external S3 data source.

Subtopics

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.
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 AI Essentials Software.

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.

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=spark-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.

More information

- Securely Passing Spark Configuration Values
- 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 AI Essentials 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 AI Essentials Software UI.
Once connected, the Spark application can:

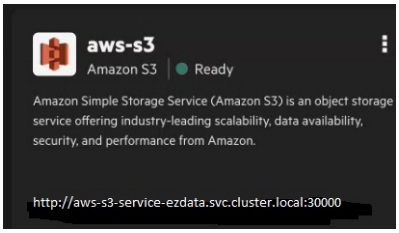
- Read and download files in a bucket

- 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 AI Essentials 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):

   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 AI Essentials 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.
- 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

   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-Analyti`

`CS/` 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_ACCESS_KEY',
os.environ['AUTH_TOKEN'])
s3_data += "\n" + s3_secret_data.replace("EXAMPLE_SECRET_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

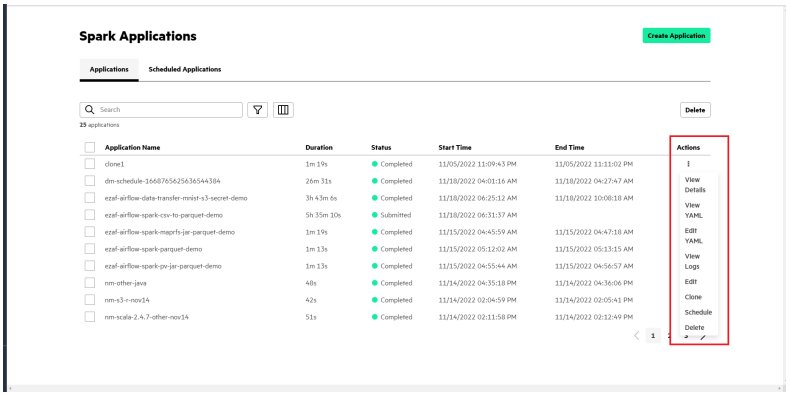
Describes how to view and manage Spark applications using HPE AI Essentials Software.

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 AI Essentials Software screen.
   - Click the Tools & Frameworks icon on the left navigation bar and click Open on Spark Operator.
2. To view actions that you can perform on the Applications and Scheduled Applications tab, click the menu icon in the Actions column.



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.

Suspend:

To stop the application from running at its scheduled time, select Suspend from the Actions menu in the Scheduled Applications tab.

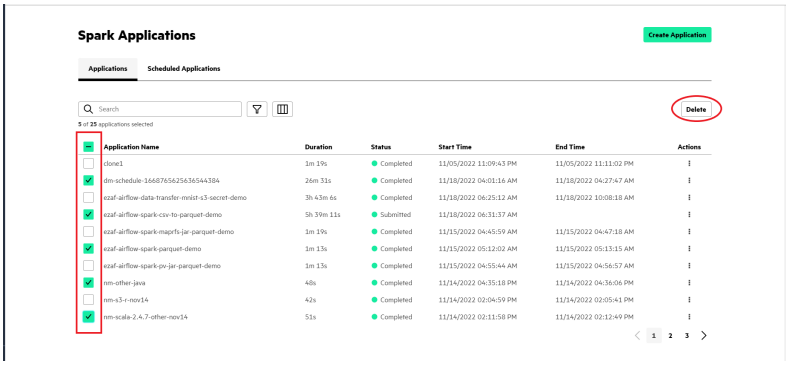
Resume:

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:



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 AI Essentials Software. See Creating Spark Applications.

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 AI Essentials 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.

```
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 AI Essentials Software.

Prerequisites

- Sign in to HPE AI Essentials Software. See Get Started.

About this task

Create an interactive session in HPE AI Essentials 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 AI Essentials Software screen.
   - Click the Tools & Frameworks icon on the left navigation bar and click Open on Livy.
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 AI Essentials Software.

Prerequisites

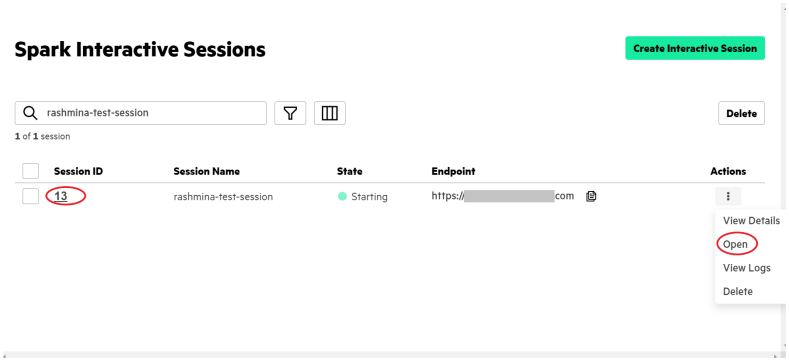
- Create an interactive session. See Creating Interactive Sessions.

About this task

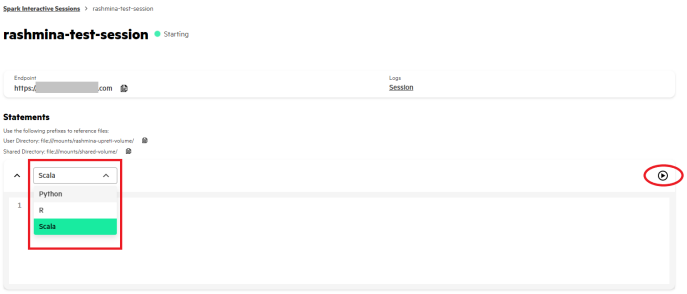
Run statements in Python, R, or Scala.

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.



2. Select either Python, R, or Scala as the statements' programming language.



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:

> **NOTE**
>
> Each Spark interactive session expires in 60 minutes.

Managing Interactive Sessions

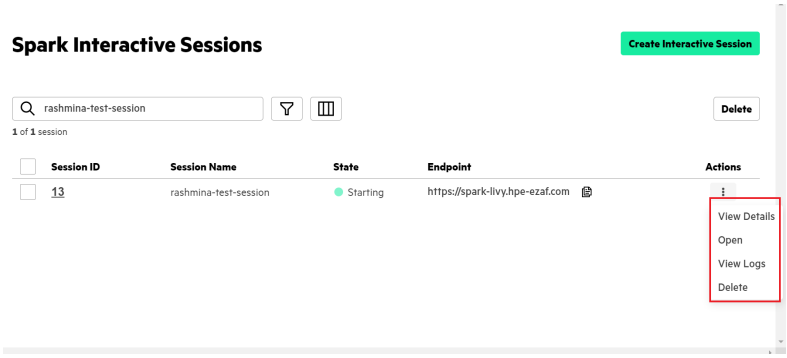
Describes how to view and manage Spark interactive sessions in HPE AI Essentials Software.

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 AI Essentials Software screen.
   - Click the Tools & Frameworks icon on the left navigation bar and click Open on Livy.
2. To view actions that you can perform on the Spark Interactive Sessions screen, click the menu icon in the Actions column.



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.

View Logs:

To view the session logs provided by Livy server, select View Logs.

Delete:

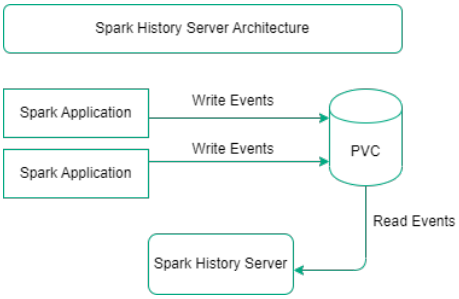
To delete the Spark interactive session, select Delete.

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 AI Essentials Software, click the Tools & Frameworks icon on the left navigation bar. Navigate to the Spark History Server tile 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 AI Essentials Software.

In HPE AI Essentials 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 AI Essentials Software, you can use **RAPIDS** Accelerator for Apache Spark by NVIDIA to accelerate the processing for Spark by using the GPUs.

The default Spark image has a built-in open-source RAPIDS plugin in HPE AI Essentials Software.

To see the list of Spark images, see List of Spark Images.

NOTE

- Do not allocate GPUs for a driver pod. GPUs are used by executor pods only.

Spark Configurations for GPU

Spark Configurations	Key	Value
GPU Images See List of Spark Images	`spark.kubernetes.container.image`	`lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-spark/<spark-version>:<image-tag>` .
Enable RAPIDS plugin	`spark.plugins`	`com.nvidia.spark.SQLPlugin`
	`spark.rapids.sql.enabled`	`true`
	`spark.rapids.force.caller.classloader`	`false`
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.

Subtopics

Enabling GPU Support for Spark Operator
Describes how to enable and allocate GPU resources on Spark Operator.
Enabling GPU Support for Livy Sessions
Describes how to enable and allocate GPU resources on Livy Server.

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 `lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-spark/<spark-version>:<image-tag>`. To see the list of Spark images, see List of Spark Images.
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>.SparkShimServiceProvider"
     ```

     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 3.5.2

Example:

```
apiVersion: "sparkoperator.hpe.com/v1beta2"
```

```
kind: SparkApplication
metadata:
  name:
spark-eep-gpu-352
  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-
3.5.2/examples/src/main/scripts/getGpusResources.sh"
    spark.rapids.shims-provider-override:
"com.nvidia.spark.rapids.shims.spark352.SparkShimServiceProvider"

  type: Python
  sparkVersion:3.5.2
  mode: cluster
  image: lr1-bd-harbor-registry.mip.storage.hpecorp.net/develop/hpe-
spark/spark-3.5.2:v3.5.2.1.1
  imagePullPolicy: Always
  mainApplicationFile: .../path/to/application.py
  restartPolicy:
    type: Never
  imagePullSecrets:
    - imagepull
  driver:
    cores: 1
    coreLimit: "1000m"
    memory: "1024m"
    labels:
      version: 3.5.2
  executor:
    cores: 1
    coreLimit: "1000m"
    instances: 1
    memory: "2G"
    labels:
      version: 3.5.2
```

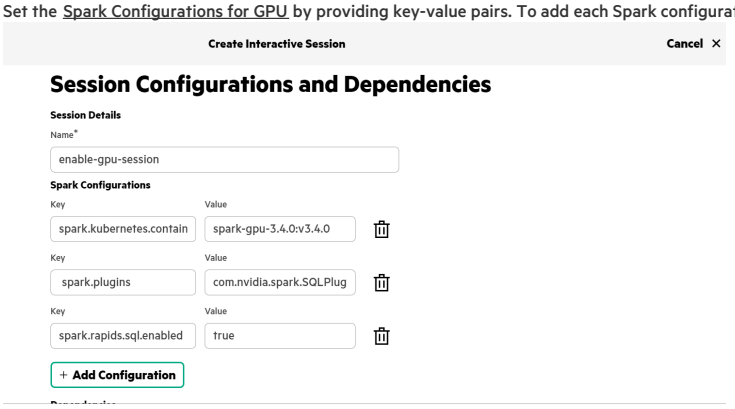
Enabling GPU Support for Livy Sessions

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.
2. Set the Spark Configurations for GPU by providing key-value pairs. To add each Spark configurations required to run your session, click Add Configuration.



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.

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 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 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.

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 AI Essentials 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.

HPE AI Essentials 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 AI Essentials Software cluster. For example, if `user01` is signed into HPE AI Essentials 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 AI Essentials 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 AI Essentials 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 AI Essentials Software responds when you submit Spark applications through the supported clients and interfaces:

Client/Interface	Description
HPE AI Essentials 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 AI Essentials 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 AI Essentials Software. HPE AI Essentials 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 AI Essentials Software to recognize you as a valid HPE AI Essentials 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 and Setting the User Context.

Data Science

Provides a brief overview of data science in HPE AI Essentials Software.

Data scientists can use programming languages such as Python, R, Java, and SQL to build, train, and deploy machine learning models in HPE AI Essentials Software using open-source tools that optimize the performance of predictive machine learning models.

Data scientists can use the tools provided in HPE AI Essentials 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.

Subtopics

Kubeflow
Provides a brief overview of Kubeflow in HPE AI Essentials Software.
MLflow
Provides a brief overview of MLflow in HPE AI Essentials Software.
MLIS
Provides a brief overview of HPE Machine Learning Inferencing Software (MLIS) in HPE AI Essentials Software.
Ray
Provides a brief overview of Ray in HPE AI Essentials Software.

Kubeflow

Provides a brief overview of Kubeflow in HPE AI Essentials 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 AI Essentials Software.

The external link for KServe InferenceService follows the following pattern:

```
service-name-namespace.domain.com
```

A dash is used between the service name and its namespace, creating one subdomain.

Features and Functionality

Kubeflow in HPE AI Essentials 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.
- 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**.

Subtopics

Kubeflow Sizing
Describes the resource allocation for different Kubeflow components.
Enabling GPU Support on Kubeflow Kserve Model Serving
Describes how to enable GPU support on Kubeflow Kserve model serving instance.

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 AI Essentials 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 AI Essentials Software, follow these steps:

Procedure

1. Click the Tools & Frameworks icon on the left navigation bar. Navigate to the Kubeflow tile 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
```

Results

The GPU is now enabled on Kubeflow Kserve model serving instance.

MLflow

Provides a brief overview of MLflow in HPE AI Essentials 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 AI Essentials Software. To learn more, see open-source **MLflow documentation**.

The model management framework with MLflow integration in HPE AI Essentials 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 AI Essentials Software

HPE AI Essentials 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 AI Essentials 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.

Subtopics

Defining RBACs on MLflow Experiments
Describes role-based access controls (RBACs) with respect to MLflow in HPE AI Essentials Software and how to define RBACs to permit access to experiments in MLflow.

Defining RBACs on MLflow Experiments

Describes role-based access controls (RBACs) with respect to MLflow in HPE AI Essentials 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 AI Essentials 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 AI Essentials Software, as either an admin or a member.

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 AI Essentials Software to *admin*. When a user has the *admin* role in HPE AI Essentials 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 AI Essentials Software admin changes a member's role to *admin* in the HPE AI Essentials 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 AI Essentials Software does not delete user experiment or model permission objects associated with the user during a hard delete. HPE AI Essentials Software 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 AI Essentials Software UI. Role changes made in HPE AI Essentials 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 AI Essentials Software, which can cause unexpected system behaviors.

Supported Access Controls

HPE AI Essentials 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(experiment_name).experiment_id
client = AuthServiceClient("http://mlflow.mlflow.svc.cluster.local:5000")
```

Create permission

```
permission = "READ"
exp_permission = client.create_experiment_permission(exp_id, user, permission)
```

Modify permission

```
permission = "EDIT"
exp_permission = client.update_experiment_permission(exp_id, user, permission)

permission = "NO_PERMISSIONS"
exp_permission = client.update_experiment_permission(exp_id, user, permission)
```

Delete permission

```
exp_permission = client.delete_experiment_permission(exp_id, user, permission)
client.get_user( 'admin' )
```

MLIS

Provides a brief overview of HPE Machine Learning Inferencing Software (MLIS) in HPE AI Essentials Software.

HPE Machine Learning Inference Software (MLIS) is a user-friendly solution designed to simplify and control the deployment, management, and monitoring of machine learning (ML) models at any scale. With a particular strength in handling large language models (LLMs), it is designed for ML practitioners and IT infrastructure engineers, helping minimize the complexity and specialized expertise needed to efficiently launch and update ML models. It provides robust tools for monitoring the performance of these models, helping ensure they operate at scale.

HPE Machine Learning Inferencing Software (MLIS) is available for use as part of HPE AI Essentials Software. MLIS is a prepackaged application that you can access in the same way that you launch applications such as Kubeflow and Ray.

MLIS Supported Versions

See the Tools and Frameworks section of the Support Matrix.

NVIDIA NIM Version Support

Upgrading to the latest version of HPE AI Essentials Software includes some updated NVIDIA NIM. Although the previous versions of these NVIDIA NIM are no longer supported, you can continue to use them in the solutions you deployed before the upgrade; however, HPE highly recommends moving to the latest models as soon as possible to avoid deployment issues. Any new solutions that you deploy must use the latest NVIDIA NIM.

In the MLIS **NGC Registry**, the **NGC Supported Models** list displays the obsolete version of the model and the updated version of the model. The latest model has a higher version number. For example, you may see `meta/llama-3.1-8b-instruct ve3` and `meta/llama-3.1-8b-instruct ve8`. The model with the `ve8` tag is the latest version.

MLIS Documentation and Release Notes

There are some special considerations for using MLIS when it is launched from HPE AI Essentials Software. For more information, see the **MLIS release notes**.

Get Started Using MLIS

To get started using MLIS applications in HPE AI Essentials Software:

1. On the left navigation bar, click the Tools & Frameworks icon.
2. Navigate to the HPE MLIS tile, and click Open.

To get started with deployments, see the Inference Service Deployment Quickstart.

Subtopics

Manage Registries
Registries are the storage locations for your models and deployments. Any storage solution that supports the S3 protocol can be used as a registry. Once added, you can use the models to create a service and deploy it for inferencing.

Manage Packaged Models
A packaged model describes the model and user code that you want to deploy as an inference service.

Manage Deployments
Deployments spin up the actual instances that run your inference services. They provide an endpoint that can be used by your applications to interact with your models.

Configuring HTTPS/TLS Certificate Trust for External Repositories
Some external repositories may use certificates that are either self-signed or issued by non-public Certificate Authorities (CAs), making them untrusted by default. This guide provides instructions on configuring MLIS to recognize and trust these certificates. Note these steps are not required for external repositories such as `huggingface.co` or `s3.amazonaws.com`, which use valid SSL/TLS certificates issued by trusted certificate authorities (CAs) for their website.

Environment Variables

Object Model Reference Schema
Description and control of your inference service is accomplished via two primary objects: **Packaged Model** and **Deployment**. Additionally, when referencing a Package Model via an external hosting method such as S3 bucket, or huggingface.co registry, you may need to configure a **Registry** to enable that access.

Pre-loading Models
When deploying an inference service, it can take a significant amount of time for the service to reach the `Ready` state if it has to download and load a large AI model during startup. Pre-loading models in MLIS significantly reduces inference service startup times, especially for large AI models.

Developer Environment Setup
The following steps guide you through setting up your MLIS developer environment to create and deploy containerized inference services. The HPE Machine Learning Inferencing Software (MLIS) developer environment is the software platform where you want to run MLIS CLI commands. This platform could be a notebook or, if you are doing remote development, a desktop shell. The MLIS developer environment is based on a Python (version 3.9 or greater) environment and includes commands from the CLI, BentoML, and OpenLLM.

Inference Service Deployment Quickstart
This guide walks you through the entire service deployment journey, from registry setup to requesting a prediction from your deployed service.

Proxy-Related Inference Failures

Failed Deployment
Deploying an inference service can fail for various reasons. This guide helps you troubleshoot common issues that may cause a deployment to fail.

No Streamed Responses
Learn how to troubleshoot responses that are not streaming as expected.

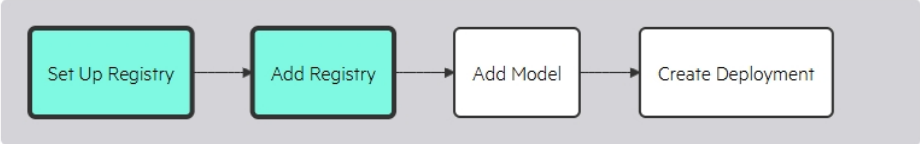
Manage Registries

Registries are the storage locations for your models and deployments. Any storage solution that supports the S3 protocol can be used as a registry. Once added, you can use the models to

create a service and deploy it for inferencing.

Service Deployment Journey

Registry setup and creation are the first two steps in the service deployment journey. It requires that you have already set up a compatible storage solution and configured any necessary policy permissions for the platform to access it.



Subtopics

Set Up a Registry
A registry is the storage location for your models and deployments. Any storage solution that supports the S3 protocol can be used as a registry. Once added, you can use the models to add a deployment for your inference service.
Add Registry
By adding a registry to HPE Machine Learning Inferencing Software (MLIS) , you grant the platform read access to any models stored in that registry. Services can then be created and deployed in your Kubernetes cluster by pulling in these models and assigning the required resources.
Edit Registry
You can edit a registry's details, such as its **access key**, **secret access key**, and **endpoint**.
Show Registry
You can inspect a registry's details, such as its **name**, **access key**, **secret access key**, and **endpoint**.
Delete Registry
You can remove a registry from HPE Machine Learning Inferencing Software (MLIS) if it is no longer required.

Set Up a Registry

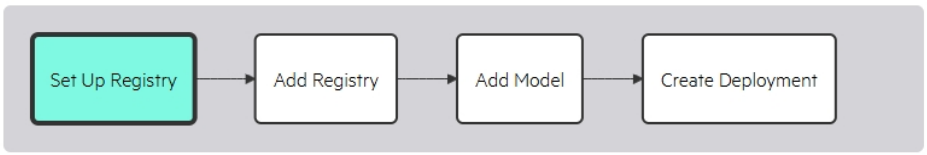
A registry is the storage location for your models and deployments. Any storage solution that supports the S3 protocol can be used as a registry. Once added, you can use the models to add a deployment for your inference service.

The following table lists the supported model types for each kind of registry you can set up:

Model Type	Registries
Bento Archive	S3
Custom	OpenLLM, PVC, S3, None
NIM	NGC, PVC
OpenLLM	OpenLLM, S3
vLLM	HuggingFace, S3

Service Deployment Journey

Registry setup is the first step in the service deployment journey. It requires that you have admin access to a compatible storage solution and are familiar with creating and configuring buckets and policy permissions.



Subtopics

AWS S3 Registry Setup
This guide provides a comprehensive walkthrough on how to leverage an AWS S3 bucket as a centralized registry for storing your machine learning models and facilitating their deployments.
Huggingface Registry
You can create an HuggingFace registry through a **Hugging Face** account. This enables HPE Machine Learning Inferencing Software to pull vLLM-compatible models directly from the Hugging Face model hub.
Minio Registry Setup
You can use Minio as a registry to store your models and deployments. Minio is an open-source object storage server that is compatible with Amazon S3. For the most up-to-date information on using Minio, see the **Minio documentation**.
NGC Registry Setup
You can create an NGC registry by setting up your **NVIDIA Enterprise Account** account. This enables HPE Machine Learning Inferencing Software (MLIS) to pull your models directly from the NVIDIA NGC catalog.
OpenLLM Registry
You can create an OpenLLM registry through a **Hugging Face** account. This enables HPE Machine Learning Inferencing Software to pull OpenLLM-compatible models directly from the Hugging Face model hub.

AWS S3 Registry Setup

This guide provides a comprehensive walkthrough on how to leverage an AWS S3 bucket as a centralized registry for storing your machine learning models and facilitating their

deployments.

By following the outlined steps, you will:

- **Create an S3 Bucket**: Establish the foundational storage location for your models.
- **Generate a Read-Only Policy**: Ensure secure, read-only access to your models, safeguarding them from unauthorized modifications.
- **Create an IAM User**: Set up a dedicated identity for managing and accessing the S3 bucket.
- **Issue Access and Secret Keys**: Obtain credentials for the IAM user, enabling authenticated interactions between your platform and the S3 bucket.

This guide does not cover the process of adding the S3 bucket as a registry within the HPE Machine Learning Inferencing Software platform. For instructions on how to add a registry, refer to the Add Registry guide.

Before You Start

- Ensure that you have an AWS account and the AWS CLI installed.
- Ensure that you have the necessary permissions to create an S3 bucket, policies, and IAM users.

How to Set Up an AWS S3 Registry

1. Create an S3 Bucket:
   a. Sign in to the AWS CLI or Console.
   b. Create an S3 Bucket.

```
aws s3 mb s3://<BUCKET_NAME> --region <REGION>
```

2. Create a Read-Only Policy:
   a. Define the policy details as a JSON file.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::<BUCKET_NAME>",
        "arn:aws:s3:::<BUCKET_NAME>/*"
      ]
    }
  ]
}
```

   b. Create the policy.

```
aws iam create-policy --policy-name <POLICY_NAME> --policy-document
file://<POLICY_FILE>.json
```

3. Create an IAM User & Attach the Policy:
   a. Create a new IAM user.

```
aws iam create-user --user-name <BUCKET_NAME>-read-only
```

   b. Attach the policy to the user.

```
aws iam attach-user-policy --user-name <BUCKET_NAME>-read-only --policy-arn
<POLICY_ARN>
```

4. Create an Access & Secret Key:
   a. Create an access key for the user.

```
aws iam create-access-key --user-name <BUCKET_NAME>-read-only
```

```
Creating access key for user: model-registry-24f62156-read-only
{
  "AccessKey": {
    "UserName": "model-registry-24f62156-read-only",
    "AccessKeyId": "AKIAYLZZO5KKY7PPOME7",
    "Status": "Active",
    "SecretAccessKey": "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",
    "CreateDate": "2024-03-21T16:18:33+00:00"
  }
}
```

   b. Save the `accessKeyID` and `secretAccessKey` values. You will need these credentials to add the S3 bucket as a registry within the HPE Machine Learning Inferencing

Software platform.

You're now ready to add an S3 registry to HPE Machine Learning Inferencing Software.

Script for Registry Setup

You can use the following script to automate the process of setting up an AWS S3 bucket as a registry. Replace the placeholder values with your actual bucket name and region.

```
#!/bin/bash

RANDOM_IDENTIFIER=$(openssl rand -hex 4)

# Define variables
REGION=""
BUCKET_NAME="model-registry-${REGION:-us-east-2}-${RANDOM_IDENTIFIER}"
POLICY_NAME="${BUCKET_NAME}-read-only-policy"
USER_NAME="${BUCKET_NAME}-read-only"
POLICY_FILE="${BUCKET_NAME}_policy.json"

# Create the policy JSON file
cat <<EOF >${POLICY_FILE}
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "s3:GetObject",
            "s3:ListBucket",
            "s3:GetBucketLocation"
         ],
         "Resource": [
            "arn:aws:s3:::${BUCKET_NAME}",
            "arn:aws:s3:::${BUCKET_NAME}/*"
         ]
      }
   ]
}
EOF

# Step 1: Create an S3 Bucket
echo "Creating S3 bucket: ${BUCKET_NAME}"
aws s3 mb s3://${BUCKET_NAME} --region ${REGION}

# Step 2: Create a Read-Only Policy
echo "Creating IAM policy: ${POLICY_NAME}"
POLICY_ARN=$(aws iam create-policy --policy-name ${POLICY_NAME} --policy-
document file://${POLICY_FILE} --query 'Policy.Arn' --output text)

# Step 3: Create an IAM User & Attach the Policy
echo "Creating IAM user: ${USER_NAME}"
aws iam create-user --user-name ${USER_NAME}

echo "Attaching policy to user"
aws iam attach-user-policy --user-name ${USER_NAME} --policy-arn ${POLICY_ARN}

# Step 4: Create an Access & Secret Key
echo "Creating access key for user: ${USER_NAME}"
aws iam create-access-key --user-name ${USER_NAME}

echo "Setup completed successfully. Remember to securely store the generated
access and secret keys."

# Cleanup the policy file
rm ${POLICY_FILE}
echo "Policy file ${POLICY_FILE} removed."
```

Huggingface Registry

You can create an HuggingFace registry through a **Hugging Face** account. This enables HPE Machine Learning Inferencing Software to pull vLLM-compatible models directly from the Hugging Face model hub.

Before You Start

- Create a Hugging Face account at **Hugging Face**.

How to Set Up an HuggingFace Registry

1. Go to the **Hugging Face website** and sign in to your account.
2. Navigate to your **Profile** and select **Settings**.
3. Select **Access Tokens**.
4. Select **New Token**.
5. Input the following:
   - **Name**: A name for your token.
   - **Type**: Select `read`.
6. Select **Generate a Token**.
7. Copy the token.

You can now proceed to Add Packaged Model for your models and deployments.

Minio Registry Setup

You can use Minio as a registry to store your models and deployments. Minio is an open-source object storage server that is compatible with Amazon S3. For the most up-to-date information on using Minio, see the **Minio documentation**.

Using Minio as a registry deployed within your organization's Kubernetes cluster can potentially improve the performance of your model deployments by reducing the latency of pulling models from a remote registry.

Once you have set up Minio, you can Add Registry for your models and deployments.

NGC Registry Setup

You can create an NGC registry by setting up your **NVIDIA Enterprise Account** account. This enables HPE Machine Learning Inferencing Software (MLIS) to pull your models directly from the NVIDIA NGC catalog.

Before You Start

- Register for an **NVIDIA Enterprise Account**, To learn how to activate the NVIDIA software subscription in HPE Private Cloud AI, see **Activating the NVIDIA software subscription** page.
- Review the **NVIDIA NIM for LLMS** guide.
- Install the **NGC CLI** on your local machine.

How to Set Up an NGC Registry

1. Generate a Personal Key:
   a. Go to **Setup > Personal Keys**.
   b. Select **+ Generate Personal Key**.
   c. Provide inputs for the following:
      - **Key Name**: A name for your key.
      - **Expiration**: Select an expiration date.
      - **Services Included**:
        - NGC Catalog
        - Private Registry
   d. Select **Generate Personal Key**. It will look similar to the following:

      ```
      nvapi-E6B8LiYAwRkJ3W-
      8wmaWs34vXPTOWa0bINjMeOWjGGA7OiOAk6xrrkwLmEOWsg8Y
      ```

   e. Copy the key and store it in a secure location.
2. Configure NGC CLI:
   a. Run the following command to configure the NGC CLI:

      ```
      ngc config set
      ```

   b. Provide the following inputs:

      ```
      Enter API key [no-apikey]. Choices: [<VALID_APIKEY>, 'no-apikey']: nvapi-
      E6B8LiYAwRkJ3W-8wmaWs34vXPTOWa0bINjMeOWjGGA7OiOAk6xrrkwLmEOWsg8Y
      Enter CLI output format type [ascii]. Choices: ['ascii', 'csv', 'json']:
      Enter org [no-org]. Choices: ['<ORG-NAME>']:
      Enter team [no-team]. Choices: ['no-team']:
      Enter ace [no-ace]. Choices: ['no-ace']:
      ```

3. Set organization and team names for NIM access:

   a. Set the organization name. This selects which provider's NIMs (NVIDIA Inference Microservices) you want to use.

      - If you are using NVIDIA-provided NIMs, set the organization name to `nim`.
      - If your organization publishes its own customized NIMs, set the organization name to your organization's name.

   b. Set the team name. This acts as a filter within the organization when you later select a NIM from the Packaged Model list.

      - If you leave the team name blank, all NIMs under the specified organization will be visible.
      - For NVIDIA-provided NIMs, you may also use a team name, if applicable.

   NOTE

   You can see the organization and team names in the NIM image path. For example:

   ```
   nvcr.io/nim/meta/llama-3.2-1b-instruct:1.8.3
   Org: nim
   Team: meta
   ```

4. Test:
   Authenticate your local docker client with the NGC registry to validate the configuration was successful.

   a. Log in to docker:

   ```
   docker login nvcr.io
   ```

   b. Use the following credentials:

   ```
   Username: $oauthtoken
   Password: nvapi-E6B8LiYAwRkJ3W-
   8wmaWs34vXPTOWa0bINjMeOWjGGA7OiOAk6xrrkwLmEOWsg8Y
   ```

   ```
   Login Succeeded
   ```

   You're now ready to add an NGC registry to HPE Machine Learning Inferencing Software.

NGC CLI Commands

For an in-depth list of NGC CLI commands, see the **NGC CLI documentation**.

OpenLLM Registry

You can create an OpenLLM registry through a **Hugging Face** account. This enables HPE Machine Learning Inferencing Software to pull OpenLLM-compatible models directly from the Hugging Face model hub.

Before You Start

- Create a Hugging Face account at **Hugging Face**.
- OpenLLM-compatible models (see **openllm 0.4.44** for the list of supported models).

How to Set Up an OpenLLM Registry

1. Go to the **Hugging Face website** and sign in to your account.
2. Navigate to your **Profile** and select **Settings**.
3. Select **Access Tokens**.
4. Select **New Token**.
5. Input the following:
   - **Name**: A name for your token.
   - **Type**: Select `read`.
6. Select **Generate a Token**.
7. Copy the token.

You can now proceed to adding a registry for your models and deployments.

Add Registry

By adding a registry to HPE Machine Learning Inferencing Software (MLIS) , you grant the platform read access to any models stored in that registry. Services can then be created and deployed in your Kubernetes cluster by pulling in these models and assigning the required resources.

Before You Start

- **S3:** Ensure that you have set up a Registry (S3 bucket) and obtained the necessary access keys.
- **HuggingFace or OpenLLM:** Sign up for a Hugging Face account and create an access token.
  - **Profile** > **Settings** > **Access Tokens**
  - **New Token**
- **NGC:** Sign up for an Nvidia NGC account and obtain the necessary API key.
  - **Profile** > **Setup** > **Generate API Key**

Add a Registry Via the UI

1. Click Tools & Frameworks on the left-navigation bar, and navigate to the HPE MLIS tile and click Open.
2. Navigate to **Registries**, and click **Create new registry**.
3. Select your model registry provider from the available options and click Continue.
4. Input details for the following based on the registry type you are adding:
   External S3 Registry

   This registry type enables access to an S3 bucket or s3-compatible storage.

   - **Name***: The name of the registry within HPE Machine Learning Inferencing Software (MLIS).
   - **Bucket ***: The actual name of the S3 bucket.
   - **Endpoint***: `http://s3.amazonaws.com` or a region-specific endpoint if your S3 bucket was created in a region other than the default (`us-east-1`); for example, `http://s3-us-west-2.amazonaws.com`.
   - **Access Key***: The access key for the S3 bucket.
   - **Secret Key***: The secret key for the S3 bucket; this is only visible when creating the S3 bucket.

   > **NOTE**
   > BentoML/OpenLLM does not support the "Insecure HTTPS" option. If you are attempting to download a Bento or Bento archive from S3 via HTTPS with an untrusted CA, you must use the S3 Proxy Object Data Store method.

   Internal S3 registry

   This registry type enables access to a bucket in an object store data source that HPE AI Essentials Software connects to. Access is made possible by the S3 proxy in HPE AI Essentials Software.
   For details about connecting HPE AI Essentials Software to an object store, see Connecting Data Sources.

   To connect a registry to an object store data source in HPE AI Essentials Software, you must have the endpoint URL and bucket name. You can get the endpoint URL from the data source tile.

   In the HPE AI Essentials Software UI:

   a. Select **Engineering > Data Sources** in the left navigation panel.
   b. Select the **Object Store Data** tab.
   c. Copy the endpoint URL on the tile of the object store that you want the registry to access.

   Once you have the endpoint URL, you can proceed with adding the registry.

   - **Name***: The name of the registry within HPE Machine Learning Inferencing Software (MLIS).
   - **Object Store***: Select the Object Store from the drop-down.
   - **Bucket***: Select the bucket from the drop-down.

   External NGC Registry

   This registry type enables direct download from the **NVIDA NGC: AI Development Catalog**.

   - **Name***: The name of the registry within HPE Machine Learning Inferencing Software (MLIS).
   - **API key***: The API key for the Nvidia NGC registry.
   - **Org name***: The organization name for the NGC registry.
   - **Team name**: The team name for the NGC registry; optional.
   - **Endpoint**: The endpoint for the NGC registry; optional.

   HuggingFace vLLM

   This registry type enables direct download from the **Hugging Face model hub**.

   - **Name***: The name of the registry within HPE Machine Learning Inferencing Software (MLIS).
   - **HuggingFace Token***: A huggingface.co user access token. See Huggingface Registry.

   Hugging Face OpenLLM

   This registry type enables direct download from the **Hugging Face model hub**.

- **Name***: The name of the registry within HPE Machine Learning Inferencing Software (MLIS).
- **HuggingFace Token***: A huggingface.co user access token. See OpenLLM Registry.

5. Select **Create registry**.

That's it! You have successfully added a new registry to HPE Machine Learning Inferencing Software (MLIS). You can now create a packaged model and associate it with this registry.

Add a Registry Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

```
aioli auth login <YOUR_USERNAME>
```

2. Create a new registry with the following command:

```
aioli registry create <REGISTRY_NAME> \
--type s3 \
--bucket <BUCKET_ADDRESS> \
--access-key <ACCESS_KEY> \
--secret-key <SECRET_KEY> \
--endpoint-url <BUCKET_ENDPOINT_URL> \
```

Add a Registry Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Use the following cURL command to add a new registry.

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/registries' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
  -d '{
    "accessKey": "<ACCESS_KEY>",
    "bucket": "<BUCKET_ADDRESS>",
    "endpointUrl": "<BUCKET_ENDPOINT_URL>",
    "insecureHttps": true,
    "name": "<REGISTRY_NAME>",
    "secretKey": "<SECRET_KEY>",
    "type": "<TYPE>"
  }'
```

Edit Registry

You can edit a registry's details, such as its **access key**, **secret access key**, and **endpoint**.

Edit a Registry Via the UI

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to **Registries**.
4. Scroll to the registry you want to edit.
5. Select the **Ellipses** icon.
6. Select **Edit**.
7. Update the details for the following:
   S3

   This registry type enables access to an S3 bucket or s3-compatible storage.

   - **Type**: `S3`
   - **Bucket Name**: The actual name of the S3 bucket.

- **Endpoint:** `http://s3.amazonaws.com`
- **Access Key:** The access key for the S3 bucket.
- **Secret Key:** The secret key for the S3 bucket; this is only visible when creating the S3 bucket.

OpenLLM

This registry type enables direct download from the **Hugging Face model hub**.

- **Type:** `OpenLLM`
- **Endpoint:** `https://huggingface.co`
- **HuggingFace Token:** A huggingface.co user access token. See OpenLLM Registry.

NGC

This registry type enables direct download from the **NVIDA NGC: AI Development Catalog**.

- **Type:** `NGC`
- **API key:** The API key for the Nvidia NGC registry.
- **Org name:** The organization name for the NGC registry.
- **Team name:** The team name for the NGC registry; optional.
- **Endpoint:** The endpoint for the NGC registry; optional.

PFS

This registry type enables direct download from HPE Machine Learning Data Management's repositories.

- **Type:** `PFS`
- **Cluster Location:** The location of the HPE Machine Learning Data Management cluster (`pachd` address).
- **Auth Token:** The authentication token for the HPE Machine Learning Data Management cluster.
- **Project Name:** The name of the project in HPE Machine Learning Data Management (**`[<commit>.]<branch>.<repo>.<project>`**).
- **Repository Name:** The name of the repository in HPE Machine Learning Data Management.

8. Select **Save**.

Edit a Registry Via the CLI

1. Sign in via the CLI.

```
aioli user login <YOUR_USERNAME>
```

2. Update the registry with the following command:

```
aioli registry update <REGISTRY_NAME> \
--type <REGISTRY_TYPE> \
--bucket <BUCKET_ADDRESS> \
--access-key <ACCESS_KEY> \
--secret-key <SECRET_KEY> \
--endpoint-url <BUCKET_ENDPOINT_URL>
```

Edit a Registry Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Get a list of registries to find the registry ID.

```
curl -X 'GET' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/registries' \
    -H 'accept: application/json' \
    -H 'Authorization : Bearer <YOUR_ACCESS_TOKEN>'
```

4. Copy the registry ID from the response.
5. Use the following cURL command to edit the registry.

```
curl -X 'PUT' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/registries/<REGISTRY_ID>' \
```

```
 -H 'accept: application/json' \
 -H 'Content-Type: application/json' \
 -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
 -d '{
 "accessKey": "<ACCESS_KEY>",
 "bucket": "<BUCKET_ADDRESS>",
 "endpointURL": "<BUCKET_ENDPOINT_URL>",
 "insecureHttps": true,
 "name": "<REGISTRY_NAME>",
 "secretKey": "<SECRET_KEY>",
 "type": "<REGISTRY_TYPE>"
}'
```

Show Registry

You can inspect a registry's details, such as its **name**, **access key**, **secret access key**, and **endpoint**.

Before You Start

- If using the API, obtain the registry's ID before proceeding.

Show Registry Via the UI

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to **Registries**.
4. Scroll to the registry you want to edit.
5. Select the **Ellipses** icon.
6. Select **Open**.

Show Registry Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

   ```
   aioli auth login <YOUR_USERNAME>
   ```

2. Inspect the registry with the following command:

   ```
   aioli registry list
   ```

3. Scroll to the row of the registry you want to inspect.

Show Registry Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

   ```
   curl -X 'POST' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
    -H 'accept: application/json' \
    -H 'Content-Type: application/json' \
    -d '{
    "username": "<YOUR_USERNAME>",
    "password": "<YOUR_PASSWORD>"
   }'
   ```

2. Obtain the Bearer token from the response.
3. Use the following cURL command to inspect a registry.

   ```
   curl -X 'GET' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/registries/<REGISTRY_ID>' \
     -H 'accept: application/json' \
     -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>'
   ```

Delete Registry

You can remove a registry from HPE Machine Learning Inferencing Software (MLIS) if it is no longer required.

Before You Start

- You must first delete any packaged models that are associated with the registry before you can remove the registry.

Delete a Registry Via the UI

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to **Registries**.
4. Scroll to the registry you want to delete.
5. Select the **Ellipses** icon.
6. Select **Delete**.
7. Confirm by selecting **Yes, Delete**.

Delete a Registry Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

```
aioli auth login <YOUR_USERNAME>
```

2. Delete the registry with the following command:

```
aioli registry delete <REGISTRY_NAME>
```

Delete a Registry Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS) .

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Get a list of registries to find the registry ID.

```
curl -X 'GET' \
   '<YOUR_EXT_CLUSTER_IP>/api/v1/registries' \
   -H 'accept: application/json' \
   -H 'Authorization : Bearer <YOUR_ACCESS_TOKEN>'
```

4. Copy the registry ID from the response.
5. Use the following cURL command to delete the registry.

```
curl -X 'DELETE' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/registries/<REGISTRY_ID>' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>'
```

Manage Packaged Models

A packaged model describes the model and user code that you want to deploy as an inference service.

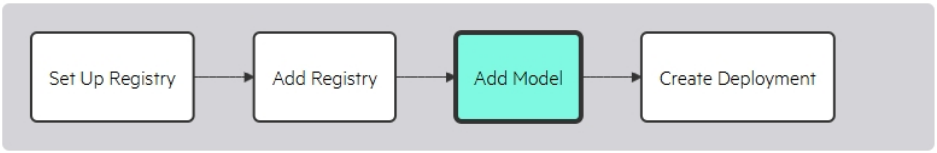
Model Packaging Options

The following model types and registry options are supported:

Model Type	Registries
Bento Archive	S3, PFS
Custom	OpenLLM, PVC, S3, PFS, None
NIM	NGC, PVC
OpenLLM	OpenLLM, S3, PFS
vLLM	HuggingFace, S3, PFS

Service Deployment Journey

Adding a packaged model is the third step in the service deployment journey. It requires that you have already created a compatible image available.



Subtopics

Add Packaged Model
Edit Packaged Model
You can edit a packaged model to update the model's details. When you edit a packaged model's **URL**, it creates a new version of that model, which is then selectable in the Model dropdown on the Deployment creation page.
Custom Image Requirements
HPE Machine Learning Inferencing Software (MLIS) provides direct support for BentoML and limited support for OpenLLM (version 0.4.44 only). For newer models, including Llama 3.1/3.2, we recommend using custom model images with KServe HuggingFace Server or vLLM. You can also build custom model images using any container build with any model runtime.
Advanced Configuration Options
While adding or editing a packaged model, you can customize the default runtime configuration MLIS uses to start your model server. This is useful for models that require modifications to MLIS's default settings or when using a custom image that needs specific configuration parameters.
Delete Packaged Model
You can delete a packaged model from HPE Machine Learning Inferencing Software (MLIS) if it is no longer required. This action will revoke access to any services that use that model.
Manage Resource Templates
HPE Machine Learning Inferencing Software (MLIS) includes default resource templates that users can select when adding or editing a packaged model. You can manage these resource templates and create new ones using the MLIS UI, CLI, or API.

Add Packaged Model

By adding a packaged model to HPE Machine Learning Inferencing Software (MLIS) , you create a versioned pointer to a model stored in a specified registry or Persistant Volume Claim (PVC). Access to reading and pulling the model's files is controlled by the registry's assigned keys.

Registry Options

The following model types and registry options are supported:

Model Type	Registries
Bento Archive	S3, PFS
Custom	OpenLLM, PVC, S3, PFS, None
NIM	NGC, PVC
OpenLLM	OpenLLM, S3, PFS
vLLM	HuggingFace, S3, PFS

Model Categories

Model Category	Description
`llm`	For large language models that support the Open AI Chat Completions interface. Enables Gen AI Knowledge Base integration and interactive chat using the Gen AI PlayGround.
`embedder`	For models that generate embeddings from input text.
`reranker`	For models that re-rank lists of candidate results.
`other`	For models that don't fit any of the above categories.

Subtopics

From Registry (UI)
Learn how to add a packaged model from a registry for your inference service using the UI.
From Registry (CLI)
From Registry (API)
From PVC (UI)
This guide explains how to add a model that's already been pre-loaded on a Persistent Volume Claim (PVC) within your Kubernetes namespace for MLIS.
From PVC (CLI)
If you have already pre-loaded a model onto a Persistant Volume Claim (PVC), you can add the model to HPE Machine Learning Inferencing Software (MLIS) by following the steps below.
From PVC (API)
If you have already pre-loaded a model onto a Persistant Volume Claim (PVC), you can add the model to HPE Machine Learning Inferencing Software (MLIS) by following the steps below.

From Registry (UI)

Learn how to add a packaged model from a registry for your inference service using the UI.

Before You Start

- Set Up a Registry.
- Confirm that your model is available in your chosen registry.
- **HuggingFace**: Sign up for a Hugging Face account and create an access token.
  - **Profile** > **Settings** > **Access Tokens**
  - **New Token**
- **OpenLLM**: Sign up for a Hugging Face account and create an access token.
  - **Profile** > **Settings** > **Access Tokens**
  - **New Token**
- **NGC**: Sign up for an Nvidia NGC account and obtain the necessary API key.
  - **Profile** > **Setup** > **Generate API Key**

Basic Details

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to Packaged Models.
4. Select **Add new model**.
5. Input details for the following:
   - **Name**: The name of the model within HPE Machine Learning Inferencing Software (MLIS).
   - **Description**: A brief description of the model.
6. Select **Next**.

Storage Details

NOTE

MLIS shows the currently available NIMs for the **organization** specified in the registry. Review the support matrix for a specific NVIDIA NIM for LLMS to properly configure the resources needed to run the model.

When the **organization** is not specified in your registry configuration, MLIS displays all available NIMs, including those not designed for LLMs. Non-LLM NIMs (such as ProteinMPNN or TTS FastPitch) are incompatible with the LLM configuration interface and may fail to launch using MLIS's default settings. To run these specialized NIMs, you may need to use the custom model format and provide specific environment variables or arguments as detailed in each NIM's documentation.

1. Input details for the following:
   - **Registry**: The registry where the model is stored.
   - **Model Format**: Options are `HuggingFace`, `OpenLLM`, `Bento archive`, `NIM`, `Custom`.
   - **Image**: The container image servicing the model; must be the name of the image + a release tag. For NIM, see the NGC catalog for the image options.
   - **URL/Path**: The location of the model object in the registry.

Prefix	URI Syntax	Description
hf://	`hf://<model-ref>`	A vLLM-compatible model name from huggingface.co dynamically loaded and executed with a vLLM backend. See supported models here: **Supported Models**.
openllm://	`openllm://<model-ref>`	An openllm model name from huggingface.co dynamically loaded and executed with a OpenLLM + VLLM backend.
s3://	`s3://<bucket-name>/<path-to-model>`	A model directory dynamically downloaded from an associated s3 registry bucket. This is supported for the bento-archive, openllm, and custom model formats.
pvc://	See From PVC (UI)	A PVC model path that can be used for pre-downloaded NIM and Custom models.

   Custom models can use any of the supported registry URI syntaxes, depending on where the model is stored.

2. Optionally, enable the **local caching** toggle to cache the model on first use. This speeds up startup times for subsequent deployments. You must have the Admin **user role** to enable local caching.
3. Select **Next**.

Resource Templates

1. Choose a Resource Template or define custom resources.

Name	Description	Request CPU	Request Memory	Request GPU	Limit CPU	Limit Memory	Limit GPU
cpu-tiny	1 cpu, 10Gi memory, no gpu per replica	1	10Gi		1	10Gi	
cpu-small	4 cpu, 20Gi memory, no gpu per replica	4	20Gi		6	40Gi	
cpu-large	8 cpu, 40Gi memory, no gpu per replica	8	40Gi		10	60Gi	
gpu-tiny	1 cpu, 10Gi, 1 gpu per replica	1	10Gi	1	1	10Gi	1
gpu-small	2 cpu, 20Gi, 2 gpu per replica	2	20Gi	2	6	40Gi	2
gpu-large	8 cpu, 40Gi, 4 gpu per replica	8	40Gi	4	10	60Gi	4

NOTE

Specifying a GPU type requires **heterogenous GPU support** be enabled.

2. Select **Next**.

Environment Variables & Arguments

Environment variables and arguments are advanced configuration options that you can set for your packaged model. These inputs will vary based on your model's requirements. For more information, see the Advanced Configuration Options reference article.

1. Provide any needed **Environment Variables**.
2. Provide any needed **Arguments**.
3. Select **Create model**.

From Registry (CLI)

Before You Start

- Set Up a Registry.
- Confirm that your model is available in your chosen registry.
- **HuggingFace:** Sign up for a Hugging Face account and create an access token.
  - **Profile** > **Settings** > **Access Tokens**
  - **New Token**
- **OpenLLM:** Sign up for a Hugging Face account and create an access token.
  - **Profile** > **Settings** > **Access Tokens**
  - **New Token**
- **NGC:** Sign up for an Nvidia NGC account and obtain the necessary API key.
  - **Profile** > **Setup** > **Generate API Key**

Supported Registry URI Syntax

Review the supported registry URI syntax before adding a model from a registry. This URI is passed in using the `--url` flag.

Prefix	URI Syntax	Description
hf://	`hf://<model-ref>`	A vLLM-compatible model name from huggingface.co dynamically loaded and executed with a vLLM backend.
openllm://	`openllm://<model-ref>`	An openllm model name from huggingface.co dynamically loaded and executed with a OpenLLM + VLLM backend.
s3://	`s3://<bucket-name>/<path-to-model>`	A model directory dynamically downloaded from an associated s3 registry bucket. This is supported for the bento-archive, openllm, and custom model formats.
pvc://	See From PVC (UI)	A PVC model path that can be used for pre-downloaded NIM and Custom models.
ngc://	Not supported	

How to Add a Packaged Model From a Registry

Step 1: Sign in via the CLI

See Developer Environment Setup for information on using the CLI.

Use the following command to sign in:

```
aioli auth login <YOUR_USERNAME>
```

Step 2: List Available Models and Obtain Values

Retrieve the allowed image and URL values for your model with the following command:

```
aioli registries models <REGISTRY-NAME>
```

The output of this command provides:

- URLs for openllm registries.
- Image specifications for NGC registries (URLs are not used for nim-format packaged models).

Step 3: Create a New Model

Now, create a model using the retrieved values:

```
aioli model create <MODEL_NAME> \
--description <DESCRIPTION> \
--image <USER_NAME>/<MODEL_NAME>:<TAG> \
--url <OBJECT_URL> \
--registry <REGISTRY_NAME> \
--enable-caching
```

- For more information on the `aioli model create` command, see the **CLI command reference**.
- For information on setting environment variables and arguments, see the Advanced Configuration Options reference article.

From Registry (API)

Before You Start

- Set Up a Registry.
- Confirm that your model is available in your chosen registry.
- **HuggingFace:** **Sign up for a Hugging Face account** and create an access token.
  - **Profile** > **Settings** > **Access Tokens**
  - **New Token**
- **OpenLLM:** **Sign up for a Hugging Face account** and create an access token.
  - **Profile** > **Settings** > **Access Tokens**
  - **New Token**
- **NGC:** **Sign up for an Nvidia NGC account** and **obtain the necessary API key**.
  - **Profile** > **Setup** > **Generate API Key**

Supported Registry URI Syntax

Review the supported registry URI syntax before adding a model from a registry. This URI is passed in using the `url` field.

Prefix	URI Syntax	Description
hf://	`hf://<model-ref>`	A vLLM-compatible model name from huggingface.co dynamically loaded and executed with a vLLM backend.
openllm://	`openllm://<model-ref>`	An openllm model name from huggingface.co dynamically loaded and executed with a OpenLLM + VLLM backend.
s3://	`s3://<bucket-name>/<path-to-model>`	A model directory dynamically downloaded from an associated s3 registry bucket. This is supported for the bento-archive, openllm, and custom model formats.
pvc://	See From Registry (UI)	A PVC model path that can be used for pre-downloaded NIM and Custom models.
pfs://	`pfs://<project>/<repo>@<commit>[:<path>][?containerPath=<path>]`	A PFS model path that can be used for models stored in HPE Machine Learning Data Management repositories.

How to Add a Packaged Model From a Registry

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Use the following cURL command to add a new packaged model.

```
curl -X 'POST' \
```

```
'<YOUR_EXT_CLUSTER_IP>/api/v1/models' \
-H 'accept: application/json' \
-H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
-H 'Content-Type: application/json' \
-d '{
"arguments": ["--debug"],
"description": "<DESCRIPTION>",
"environment": {
   "key": "value",
   "key2": "value2"
}
"image": "<USER_NAME>/<MODEL_NAME>:<TAG>",
"modelFormat": "<MODEL_FORMAT>",
"name": "<MODEL_NAME>",
"registry": "<REGISTRY>",
"resources": {
   "gpuType": "<GPU_TYPE>",
   "limits": {
      "cpu": "<CPU_LIMIT>",
      "gpu": "<GPU_LIMIT>",
      "memory": "<MEMORY_LIMIT>"
   },
   "requests": {
      "cpu": "<CPU_REQUEST>",
      "gpu": "<GPU_REQUEST>",
      "memory": "<MEMORY_REQUEST>"
   }
},
"url": "<OBJECT_URL>"
}'
```

From PVC (UI)

This guide explains how to add a model that's already been pre-loaded on a Persistent Volume Claim (PVC) within your Kubernetes namespace for MLIS.

Before You Start

- Create a PVC and pre-load a model.
- Obtain the model's path within the PVC.
- Review model-specific resources (e.g., **NVIDIA NIM documentation**) that you may need for adding specific environment variables and arguments during packaged model creation.

PVC Syntax & URL Options

Option	Description	Example	Default
PVC Name	Name of the Persistent Volume Claim (PVC) to be mounted	`pvc://my-model-pvc`	Required, no default
Path	Optional path within the PVC to be mounted	`pvc://my-model-pvc/models`	If not specified, the entire PVC is mounted
ContainerPath	Directory in container where the PVC is mounted	`pvc://my-model-pvc?containerPath=/mnt/models`	`/mnt/models`
readOnly	Whether the volume is read-only	`pvc://my-model-pvc?readOnly`	If not specified, the volume is read-write

NOTE
The PVC must exist in the Kubernetes namespace where the packaged model will be deployed.

Adding the Model

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to **Packaged Models**.
4. Select **Add new model** and fill in the basic details.
5. In **Storage Details**:
   - Set **Registry** to `None` and **Model Format** to `Custom`.
   - Enter the appropriate **Image** name.
   - Specify the **URL/Path** using PVC syntax (e.g., `pvc://models-cache-pvc?containerPath=/mnt/models`).
6. Choose a Resource Template or define custom resources.

Name	Description	Request CPU	Request Memory	Request GPU	Limit CPU	Limit Memory	Limit GPU
cpu-tiny	1 cpu, 10Gi memory, no gpu per replica	1	10Gi		1	10Gi	
cpu-small	4 cpu, 20Gi memory, no gpu per replica	4	20Gi		6	40Gi	
cpu-large	8 cpu, 40Gi memory, no gpu per replica	8	40Gi		10	60Gi	
gpu-tiny	1 cpu, 10Gi, 1 gpu per replica	1	10Gi	1	1	10Gi	1
gpu-small	2 cpu, 20Gi, 2 gpu per replica	2	20Gi	2	6	40Gi	2
gpu-large	8 cpu, 40Gi, 4 gpu per replica	8	40Gi	4	10	60Gi	4

NOTE

Specifying a GPU type requires **heterogenous GPU support** be enabled.

7. Set any necessary **Environment Variables** and **Arguments** based on your packaged model's framework type. For more information, see the Advanced Configuration Options reference article.
8. Select **Create model**.

From PVC (CLI)

If you have already pre-loaded a model onto a Persistant Volume Claim (PVC), you can add the model to HPE Machine Learning Inferencing Software (MLIS) by following the steps below.

Before You Start

- Create a PVC.
- Obtain the model's path within the PVC.
- Review any necessary resources specific to the model you have pre-loaded:
  - **NIM**: **NVIDIA NIM - Get Started, Serving Models**, **NIM Tutorials**, **NIM Deploy on KServe**

PVC Syntax & URL Options

Review the following PVC syntax and URL options to ensure you have the correct information for adding your model.

Option	Description	Example	Default
PVC Name	Name of the Persistent Volume Claim (PVC) to be mounted	`pvc://my-model-pvc`	Required, no default
Path	Optional path within the PVC to be mounted	`pvc://my-model-pvc/models`	If not specified, the entire PVC is mounted
ContainerPath	Directory in container where the PVC is mounted	`pvc://my-model-pvc?containerPath=/mnt/models`	`/mnt/models`
readOnly	Whether the volume is read-only	`pvc://my-model-pvc?readOnly`	If not specified, the volume is read-write

NOTE

The PVC Name (e.g., `<my-model-pvc>`) must already exist in the Kubernetes namespace where the packaged model will be deployed.

How to Add a Packaged Model From PVC

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

```
aioli auth login <YOUR_USERNAME>
```

2. Create a new model with the following command:

```
aioli model create <MODEL_NAME> \
--modelformat custom \
--image <USER_NAME>/<MODEL_NAME>:<TAG> \
--url pvc://<PVC_NAME>/<OPTIONAL_PATH>?containerPath=<DIR_IN_CONTAINER> \
--description <DESCRIPTION> \
--arg=--model_dir=<PATH_WHERE_MODEL_IS_STORED>
```

- For more information on the `aioli model create` command, see the **CLI command reference**.
- For information on setting environment variables and arguments, see the Advanced Configuration Options reference article.

From PVC (API)

If you have already pre-loaded a model onto a Persistant Volume Claim (PVC), you can add the model to HPE Machine Learning Inferencing Software (MLIS) by following the steps below.

Before You Start

- Create a PVC.
- Obtain the model's path within the PVC.
- Review any necessary resources specific to the model you have pre-loaded:
  - **NIM:** NVIDIA NIM - Get Started, Serving Models, NIM Tutorials, NIM Deploy on KServe

PVC Syntax & URL Options

Review the following PVC syntax and URL options to ensure you have the correct information for adding your model.

Option	Description	Example	Default
PVC Name	Name of the Persistent Volume Claim (PVC) to be mounted	`pvc://my-model-pvc`	Required, no default
Path	Optional path within the PVC to be mounted	`pvc://my-model-pvc/models`	If not specified, the entire PVC is mounted
ContainerPath	Directory in container where the PVC is mounted	`pvc://my-model-pvc?containerPath=/mnt/models`	`/mnt/models`
readOnly	Whether the volume is read-only	`pvc://my-model-pvc?readOnly`	If not specified, the volume is read-write

NOTE

The PVC Name (e.g., `<my-model-pvc>`) must already exist in the Kubernetes namespace where the packaged model will be deployed.

How to Add a Packaged Model From a Registry

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Use the following cURL command to add a new packaged model. For information on setting environment variables and arguments, see the Advanced Configuration Options reference article.

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/models' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
  -H 'Content-Type: application/json' \
  -d '{
  "arguments": ["--model_dir <PATH_WHERE_MODEL_IS_STORED>"],
  "description": "<DESCRIPTION>",
  "environment": {
     "key": "value",
     "key2": "value2"
  },
  "image": "<USER_NAME>/<MODEL_NAME>:<TAG>",
  "modelFormat": "<MODEL_FORMAT>",
  "name": "<MODEL_NAME>",
  "resources": {
     "gpuType": "<GPU_TYPE>",
     "limits": {
        "cpu": "<CPU_LIMIT>",
        "gpu": "<GPU_LIMIT>",
        "memory": "<MEMORY_LIMIT>"
     },
     "requests": {
        "cpu": "<CPU_REQUEST>",
        "gpu": "<GPU_REQUEST>",
        "memory": "<MEMORY_REQUEST>"
     }
  },
  "url": "pvc://<PVC_NAME>/<OPTIONAL_PATH>?containerPath=
<DIR_IN_CONTAINER>"
}'
```

Edit Packaged Model

You can edit a packaged model to update the model's details. When you edit a packaged model's **URL**, it creates a new version of that model, which is then selectable in the Model dropdown on the Deployment creation page.

Via the UI

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to the **Packaged Models** page.
4. Select the **Ellipsis** icon next to the model you want to edit.
5. Select **Edit**.
6. Update the packaged model details.
7. Select **Save**.

Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

```
aioli auth login admin
```

2. Update the packaged model with the following command:

```
aioli model update <MODEL_NAME> \
--description <DESCRIPTION> \
--image <USER_NAME>/<MODEL_NAME>:<TAG> \
--registry <REGISTRY_NAME>
```

Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
'<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"username": "<YOUR_USERNAME>",
"password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Get a list of packaged models to find the model ID.

```
curl -X 'GET' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/models' \
    -H 'accept: application/json' \
    -H 'Authorization : Bearer <YOUR_ACCESS_TOKEN>'
```

4. Copy the model ID from the response.
5. Use the following cURL command to edit a packaged model.

```
curl -X 'PUT' \
'<YOUR_EXT_CLUSTER_IP>/api/v1/models/<MODEL_ID>' \
-H 'accept: application/json' \
-H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
-H 'Content-Type: application/json' \
-d '{
  "arguments": ["--debug"],
  "description": "<DESCRIPTION>",
  "environment": {
      "key": "value",
      "key2": "value2"
  }
  "image": "<USER_NAME>/<MODEL_NAME>:<TAG>",
  "modelFormat": "<MODEL_FORMAT>",
  "name": "<MODEL_NAME>",
  "registry": "<REGISTRY>",
  "resources": {
      "gpuType": "<GPU_TYPE>",
      "limits": {
          "cpu": "<CPU_LIMIT>",
          "gpu": "<GPU_LIMIT>",
          "memory": "<MEMORY_LIMIT>"
      },
      "requests": {
          "cpu": "<CPU_REQUEST>",
          "gpu": "<GPU_REQUEST>",
          "memory": "<MEMORY_REQUEST>"
      }
```

```
  },
  "url": "<OBJECT_URL>"
}'
```

Custom Image Requirements

HPE Machine Learning Inferencing Software (MLIS) provides direct support for BentoML and limited support for OpenLLM (version 0.4.44 only). For newer models, including Llama 3.1/3.2, we recommend using custom model images with KServe HuggingFace Server or vLLM. You can also build custom model images using any container build with any model runtime.

Before You Start

All custom images must meet the following requirements:

- A custom containerized application must listen on port `8080`.
- Any custom metrics must be provided via a `/metrics` endpoint.
- Readiness and liveness checks are done by attempting to connect to port `8080/tcp`. (If successful, the inference service is deemed ready.)

Configure Listening Port

To configure the port for your packaged model, you have a few options:

1. Configure your container to listen on port `8080` by default.
2. Configure your container to read and use the `PORT` environment variable, which is provided by Knative. This allows the container to listen on the proper port for the inference service. It's important to note that you don't need to explicitly set the `PORT` yourself, as it is provided by Knative. However, your container needs to be designed to honor and use this `PORT` variable when it's available.
3. If your container command line can accept the desired port via command line arguments, you can configure it through the arguments of the packaged model. For example:

   ```
   aioli model update <MODEL_NAME> --env="PORT=8080"
   ```

Customizing Metrics

The default KServe-provided prometheus metrics are automatically available for any container. See the Monitoring section for more information.

Readiness & Health Checks

By default, KServe performs readiness/health checks by attempting to connect to port `8080/tcp`. So as long as the container is listening on `8080`, an explicit `/health` endpoint is not needed.

Customizing the Base Container Image

You can also customize the base container image used by HPE Machine Learning Inferencing Software (MLIS) to service your models.

Example Container Images

KServe offers an image called **`kserve/huggingfaceserver`** that can fetch and run VLLM-compatible models from **Hugging Face**. Before using this image, consider the following:

- The image accepts various parameters, but only `--model_id` is mandatory. This specifies the Hugging Face model (e.g., `facebook/opt-125m` or `meta-llama/Llama-2-7b-chat-hf`).
- If not specified, `--model_name` defaults to "model".
- For models requiring authentication, provide an `openllm` registry or set the `HF_TOKEN` environment variable in your model definition.
- By default, the image uses VLLM as the backend, which needs a GPU. For GPU-free testing of small models, you can use `--backend=huggingface` to switch to the HuggingFace backend.

Subtopics

Create Bento Archive
This guide shows you how to create a Bento Archive and export it (`bentoml export`) to an S3 bucket.

Create Image (Bentoml)
The following steps guide you through creating a containerized image for an inference service using the bentoml CLI. The result of this is a publicly accessible container published at `<user>/<model-name>` which can be referenced in an HPE Machine Learning Inferencing Software (MLIS) Model definition and deployed as a service.

Create Image (OpenLLM)
The following steps guide you through creating a containerized image for an inference service using the **OpenLLM** CLI. The result of this is a publicly accessible container published at `<user>/<model-name>` which can be referenced in an HPE Machine Learning Inferencing Software (MLIS) Model definition and deployed as a service.

Customize the Base Container Image
HPE Machine Learning Inferencing Software (MLIS) provides default images to execute `bentoml` and `openllm` models from `openllm://` or `s3://` --- but you can provide a different base image to add additional libraries or change things like dependency versions.

Create Bento Archive

This guide shows you how to create a Bento Archive and export it (`bentoml export`) to an S3 bucket.

Before You Start

- Ensure you have completed **Developer Environment Setup**.
  - You may need to install `bentoml[aws]` if you have not done so already.
- This guide assumes you have already trained a model and are ready to create a Bento.

Create Bento

Prepare your model and save it as a Bento:

```
bentoml build -o tag
```

Create S3 Bucket

1. Create an AWS profile:

   ```
   aws --profile my-aws-profile configure
   ```

   ```
   AWS Access Key ID [None]: AKIAYLGG12ZIQZLL6PUG
   AWS Secret Access Key [None]: AeazKuGrL4UgiUABVnfhwW02oNtTeB4ec9cQwE1y
   Default region name [None]:
   Default output format [None]:
   ```

2. Create an S3 bucket:

   ```
   aws --profile aioli-demo s3 mb s3://mlis-bucket --region us-east-1
   make_bucket: USER-bucket
   ```

3. List your bentos:

   ```
   bentoml list
   ```

   ```
    Tag                                Size       Model Size  Creation Time
   pytorch_mnist_demo:fml2z4xekol6llg6  20.63 KiB  1.30 MiB    2024-03-17 07:40:49
   ```

Export Bento as Archive

1. Export the Bento as an archive using the **`bentoml export`** command. The Bento archive **must** end with `.bento`:

   ```
   bentoml export pytorch_mnist_demo:latest s3://mlis-
   bucket/bentoml/pytorch_mnist_demo_.bento
   ```

   > **NOTE**
   >
   > If this command fails to export the Bento to S3 storage, make sure you have installed `bentoml` with aws support (e.g., `pip install 'bentoml[aws]'`).

2. Verify the Bento archive is accessible:

   ```
   aws --profile aioli-demo s3 ls s3://mlis-bucket/bentoml/
   ```

   ```
   2024-03-18 14:46:19    1255948 pytorch_mnist_demo.bento
   ```

Create Image (Bentoml)

The following steps guide you through creating a containerized image for an inference service using the bentoml CLI. The result of this is a publicly accessible container published at `<user>/<model-name>` which can be referenced in an HPE Machine Learning Inferencing Software (MLIS) Model definition and deployed as a service.

Before You Start

- Ensure you have completed **Developer Environment Setup**.
- Ensure you have docker installed and running.

Custom Models

If you have a custom ML model, you must provide some additional configuration to describe the desired inference service interface. See the **BentoML Iris Classification Example** for reference. Generally, the following three small files need to be provided:

- `save_model.py`: A script to import and register it in BentoML's model registry on the local filesystem.
- `service.py`: Defines the BentoML service, including the API endpoints for model inference.
- `bentofile.yaml`: Configuration file to build a BentoML service, specifying dependencies and Python environment setup and generated into the container.

How to Create a Containerized Image

1. Build the container using the `bentoml build` command.

   ```
   bentoml build -f bentofile.yaml
   ```

2. Containerize the model using the `bentoml containerize` command.

   ```
   bentoml containerize iris:latest --opt platform=linux/amd64  --image-tag iris:latest
   ```

3. Get the `IMAGE ID` of the resulting container.

   ```
   docker image ls
   ```

4. Tag the resulting container.

   ```
   docker tag <image-id> <user>/<model-name>
   ```

5. Push the container to a publicly-accessible docker repo.

   ```
   docker push <user>/<model-name>
   ```

6. Verify the container is accessible.

   ```
   docker pull <user>/<model-name>
   ```

> **NOTE**
> See the **BentoML documentation** for details for other options for configuration/building.

Create Image (OpenLLM)

The following steps guide you through creating a containerized image for an inference service using the **OpenLLM** CLI. The result of this is a publicly accessible container published at `<user>/<model-name>` which can be referenced in an HPE Machine Learning Inferencing Software (MLIS) Model definition and deployed as a service.

Before You Start

- Ensure you have completed **Developer Environment Setup**.
- Ensure you have Docker installed and running.
- Ensure you have deployed the HPE Machine Learning Inferencing Software (MLIS) controller and have the `openllm` CLI installed.
- Ensure you have an available GPU on the system.
- MLIS supports OpenLLM 0.4.44. That means it does not currently support Llama 3.1 models. For more information, see the Considerations section.

How to Create a Containerized Image

1. Build the container using the `openllm build` command.

   ```
   openllm build --backend vllm --containerize <user/model-name>
   ```

2. Get the `IMAGE ID` of the resulting container.

   ```
   docker image ls
   ```

   ```
   REPOSITORY                          TAG                 IMAGE ID       CREATED
   SIZE
   tiiuae--falcon-7b-service           898df1396f35e447d5fe44e0a3ccaaaa69f30d36
   4ac4bb1f2dce   3 minutes ago   24.9GB
   ```

3. Tag the resulting container.

   ```
   docker tag <image-id> <user>/<model-name>
   ```

4. Push the container to a publicly-accessible docker repo.

   ```
   docker push <user>/<model-name>
   ```

5. Verify the container is accessible.

   ```
   docker pull <user>/<model-name>
   ```

You are now ready to upload this image to a registry and create a packaged model in HPE Machine Learning Inferencing Software (MLIS) .

Model Testing

You can serve a model for interactive testing before building the container by specifying the desired LLM name in the following command:

`openllm start --backend vllm facebook/opt-125m`

Once started, the LLM service will be listening on `http://localhost:3000`. You may interact with it via the SwaggerUI web interface using a browser pointed to that URL. The SwaggerUI shows the available REST API methods of the service.

You can also use the `openllm query` command:

```
openllm query "What is an LLM?"
What is an LLM?
A degree in engineering
```

Considerations: Llama 3.1 and 3.2 Models

OpenLLM 0.4.44 (the version supported by HPE Machine Learning Inferencing Software (MLIS)) does not support Llama 3.1/3.2 models. For these newer models, we recommend using one of the following alternatives:

1. Use the new direct vLLM model support added in HPE Machine Learning Inferencing Software (MLIS) 1.3.0.
   - Refer to Huggingface Registry and Add Packaged Model.
2. Custom model with KServe HuggingFace Server:
   - Provides flexibility and direct integration with the Hugging Face model hub.
   - Suitable for running Llama 3.1/3.2 models.
   - For setup instructions, see our Custom Image Requirements guide.
   - For available options, run: `docker run kserve/huggingfaceserver -h`
3. vLLM container:
   - Offers high-performance inference for large language models, including Llama 3.1/3.2.
   - Use the **`vllm/vllm-openai`** image with specific CLI arguments.
   - For deployment instructions and available options, refer to the **vLLM documentation**.
   - To see available options, run: `docker run vllm/vllm-openai:latest -h`

Example for deploying a Llama 3.1 model with vLLM:

```
... --model meta-llama/Meta-Llama-3.1-70B-Instruct -tp 8 --port 8080
```

Example for creating a model with KServe HuggingFace Server:

```
aioli model create fb125m --image kserve/huggingfaceserver --arg=--
model_id=facebook/opt-125m --requests-gpu=1 --limits-gpu=1 --limits-
memory=10Gi --registry Huggingface.co
```

More information

- Open LLM 0.4.44

Customize the Base Container Image

HPE Machine Learning Inferencing Software (MLIS) provides default images to execute `bentoml` and `openllm` models from `openllm://` or `s3://` --- but you can provide a different base image to add additional libraries or change things like dependency versions.

You can provide different default images by:

- Updating the **Helm chart** `defaultImages` values at `defaultImages.bentoml` and `defaultImages.openllm`.
- Specifying a value for the `image` field when creating a Packaged Model.

Default Images

BentoML

```
python 3.9
RUN pip install bentoml[aws]==1.1.11
```

The container must provide the `bentoml` command on the PATH and allow it to be executed as container arguments. You may use the default BentoML container as a base container, or provide your own. HPE Machine Learning Inferencing Software (MLIS) will inject the necessary scripts to download the model and launch the `bentoml` command pointing to the download model.

OpenLLM

> **NOTE**
> OpenLLM support is limited to version 0.4.44 and does not support newer models like Llama 3.1/3.2. For these models, consider using KServe HuggingFace Server or vLLM alternatives.

```
python 3.11
 apt-get install -y --no-install-recommends \
  build-essential \
  ca-certificates \
  ccache \
  curl \
  bc \
  libssl-dev ca-certificates make \
  git python3-pip
 pip3 install -v --no-cache-dir "vllm==0.2.7"

 RUN pip3 install openllm==0.4.44
```

The container must provide the `openllm` command on the PATH and allow it to be executed as container arguments. You may use the default OpenLLM container as a base container, or provide your own. HPE Machine Learning Inferencing Software (MLIS) will inject the necessary scripts to download the model and launch the `openllm` command pointing to the download model.

Advanced Configuration Options

While adding or editing a packaged model, you can customize the default runtime configuration MLIS uses to start your model server. This is useful for models that require modifications to MLIS's default settings or when using a custom image that needs specific configuration parameters.

> **NOTE**
> If you want to use a fully custom container image instead of simply modifying the default runtime configuration, see Customize the Base Container Image.

Before You Start

- You should review your model's framework documentation (e.g., OpenLLM, BentoML, NIM) and its CLI options.
- You should already know the arguments and environment variables you'd like to set for your model based on previous testing.

Runtime Configuration Options

Environment Variables

Variable	Description
`AIOLI_LOGGER_PORT`	The port that the logger service listens on; default is `49160`.
`AIOLI_PROGRESS_DEADLINE`	The deadline for downloading the model; default is `1500s`.
`AIOLI_READINESS_FAILURE_THRESHOLD`	The number of readiness probe failures before the deployment is considered unhealthy; default is `100`.
`AIOLI_COMMAND_OVERRIDE`	The customized deployment command that enables you to override the default deployment command within a predefined runtime.
`AIOLI_SERVICE_PORT`	The inference service container port used for communication; default is `8080` except for NIMs, which is `8000`.
`AIOLI_DISABLE_MODEL_CACHE`	Disables automatic model caching for a deployment, even if it is enabled for the model; default is `false`.
`AIOLI_DISABLE_LOGGER`	A workaround for the Kserve defect concerning streamed responses.

Command Override Arguments

MLIS executes a default command for your container runtime based on the type of packaged model you have selected. However, you can modify this command using the `AIOLI_COMMAND_OVERRIDE` environment variable. Any arguments from the packaged model are then appended to the end of this command, followed by any arguments from the deployment (`AIOLI_COMMAND_OVERRIDE = [CLI_COMMAND] [MODEL_ARGS] [DEPLOYMENT_ARGS]`).

Default Commands

MLIS provides template variables for customizing the runtime command:

Framework	Command
OpenLLM	`openllm start --port {{.containerPort}} {{.modelDir}}`
Bento Archive	`bentoml serve ...`
Custom	`none`
NVIDIA NIM	`none`
vLLM	`--model {{.modelName}} --port {{.containerPort}} --download-dir {{.modelDir}} --tensor-parallel-size {{.numGpus}}`

Template Variables

You can also use the following variables to modify the command's arguments:

Named Argument	Description
`{{.numGpus}}`	The number of GPUs the model is requesting.
`{{.modelName}}`	The model name being deployed. For Openllm/vLLM the model name path from the url.
`{{.modelDir}}`	The directory into which the model will be downloaded. This is typically `/mnt/models`. This applies to NIM, OpenLLM, and S3 models.
`{{.containerPort}}`	The http port that the container must listen on for inference requests and readiness checks.

Examples

```
AIOLI_COMMAND_OVERRIDE="openllm start {{.modelName}} --port
{{.containerPort}} --gpu-memory-utilization 0.9 --max-total-tokens 4096"
```

```
AIOLI_COMMAND_OVERRIDE="bentoml serve {{.modelDir}}/bentofile.yaml --
production --port {{.containerPort}} --host 0.0.0.0"
```

Multi-Node Deployments and Large Models

Multi-Node Deployments

MLIS does not provide built-in support for multi-node deployments. If your model requires more resources than available from a single node in your cluster, your model runtime must explicitly support multi-distribution.

For example, the Llama 3.1 405B Instruct NIM can run on a single node with 8 H100 GPUs, but requires 16 A100 GPUs. For multi-node NIM deployments, see the NVIDIA documentation.

Large Models

Large models, such as LLama 3.1, have very high and specific resource configuration requirements to run on a single node. When working with these models, ensure that your cluster has the necessary resources and that your model runtime supports the required distribution method.

For more information, see the official LLama Model Requirements page.

Delete Packaged Model

You can delete a packaged model from HPE Machine Learning Inferencing Software (MLIS) if it is no longer required. This action will revoke access to any services that use that model.

Via the UI

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to the **Packaged Models** page.
4. Select the **Ellipsis** icon next to the model you want to delete.
5. Select **Delete**.
6. Select **Yes, Delete**.

Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

```
aioli auth login <YOUR_USERNAME>
```

2. Delete the packaged model with the following command:

```
aioli model delete <MODEL_NAME> <VERSION_NUMBER>
```

Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Get a list of packaged models to find the model ID.

```
curl -X 'GET' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/models' \
    -H 'accept: application/json' \
    -H 'Authorization : Bearer <YOUR_ACCESS_TOKEN>'
```

4. Copy the model ID from the response.
5. Use the following cURL command to delete a packaged model.

```
curl -X 'DELETE' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/models/<MODEL_ID>' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>'
```

Manage Resource Templates

HPE Machine Learning Inferencing Software (MLIS) includes default resource templates that users can select when adding or editing a packaged model. You can manage these resource templates and create new ones using the MLIS UI, CLI, or API.

Before You Start

- You must have the **Admin user role** to manage resource templates.
- Ensure your cluster has sufficient resources and appropriate GPU types to support the resource templates you create.
- Active deployments using an updated resource template will apply the new settings during the next canary rollout or upon re-deployment.

Default Resource Templates

The following table shows the default resource templates available in MLIS.

Name	Description	Request CPU	Request Memory	Request GPU	Limit CPU	Limit Memory	Limit GPU
cpu-tiny	1 cpu, 10Gi memory, no gpu per replica	1	10Gi		1	10Gi	
cpu-small	4 cpu, 20Gi memory, no gpu per replica	4	20Gi		6	40Gi	
cpu-large	8 cpu, 40Gi memory, no gpu per replica	8	40Gi		10	60Gi	
gpu-tiny	1 cpu, 10Gi, 1 gpu per replica	1	10Gi	1	1	10Gi	1
gpu-small	2 cpu, 20Gi, 2 gpu per replica	2	20Gi	2	6	40Gi	2
gpu-large	8 cpu, 40Gi, 4 gpu per replica	8	40Gi	4	10	60Gi	4

Add a Resource Template Via the UI

1. In the MLIS UI, navigate to **Settings** > **Resource templates**.
2. Select **Add new resource template**.
3. Input the name, description, and resource requirements for the template.

Field	Description
Template name	A unique identifier for the resource template
Description	A brief explanation of the template's purpose or characteristics
CPU request	The minimum CPU resources required for the packaged model to operate
CPU limit	The maximum CPU resources allowed for handling traffic spikes
Memory request	The minimum memory resources required for the packaged model to operate
Memory limit	The maximum memory resources allowed for handling traffic spikes
GPU request	The minimum GPU resources required for the packaged model to operate
GPU limit	The maximum GPU resources allowed for handling traffic spikes
GPU type	The specific GPU model required for the packaged model(e.g., NVIDIA A100). Specifying a GPU type requires **heterogenous GPU support** be enabled.

4. Select **Create template**.

The new template is now available from the **Resource Template** dropdown on the **Resources** tab when adding or editing a packaged model. To update this template, select the ellipsis icon next to the template name and choose **Edit**.

Add a Resource Template Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

```
aioli auth login admin
```

2. Add a new resource template with the following command:

```
aioli templates resource create <TEMPLATE_NAME> \
--description <TEMPLATE_DESCRIPTION> \
--gpu-type <GPU_TYPE> \
--limits-cpu <LIMITED_CPU> \
--limits-memory <LIMITED_MEMORY> \
--requests-cpu <REQUESTED_CPU> \
--requests-memory <REQUESTED_MEMORY>
```

Add a Resource Template Via the API

1. Sign in to MLIS.

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Use the following cURL command to add a new resource template.

```
curl -X 'POST' \
  'https://<YOUR_EXT_CLUSTER_IP>/api/v1/templates/resources' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
  -H 'Content-Type: application/json' \
  -d '{
    "description": "A resource template",
    "name": "my-template",
```

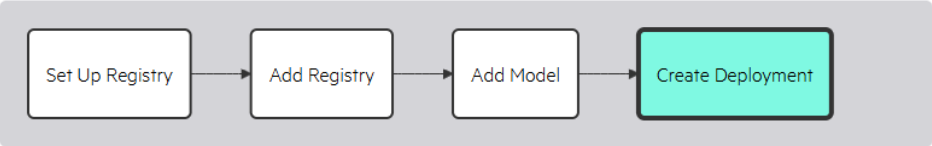
```
  "resources": {
    "gpuType": "NVIDIA_A100",
    "limits": {
      "cpu": "0.5",
      "gpu": "5",
      "memory": "2.5Gi"
    },
    "requests": {
      "cpu": "0.5",
      "gpu": "5",
      "memory": "2.5Gi"
    }
  }
}'
```

Manage Deployments

Deployments spin up the actual instances that run your inference services. They provide an endpoint that can be used by your applications to interact with your models.

Service Deployment Journey

Deployment creation is the fourth and final step in the service deployment journey. It requires that you know which Kuberentes namespace you want to deploy your service to.



Subtopics

Create Deployment
By creating a deployment, you are deploying an inference service to a Kubernetes cluster. The end result is a service with an available endpoint that can be accessed by clients to make predictions.

Edit Deployment
You can make changes to certain aspects of an active deployment, such as the packaged model, authentication requirements, and auto-scaling configurations. If you change the packaged model, you must choose between an abrupt upgrade or a canary rollout to handle the change.

Interact with a Deployment
You can interact with a deployed inference service by sending a request to the service's endpoint. The service will respond with a prediction based on the input data.

Initiate Canary Rollout
Canary rollouts enable you to safely test your new model version's performance alongside an existing model version that is currently deployed by directing a percentage of the traffic to the new model. This allows you to compare the performance of the new model with the existing model before fully deploying the new model.

Monitor Services
HPE Machine Learning Inferencing Software (MLIS) automatically configures monitoring of all deployed AI inference services.

Advanced Configuration Options
While adding or editing a deployment, you can customize the default runtime configuration MLIS uses to start your model server.

Manage Auto Scaling Templates
HPE Machine Learning Inferencing Software (MLIS) includes default autoscaling templates that users can select when adding or editing a deployment. You can manage these auto scaling templates and create new ones using the MLIS UI, CLI, or API.

Create Deployment

By creating a deployment, you are deploying an inference service to a Kubernetes cluster. The end result is a service with an available endpoint that can be accessed by clients to make predictions.

Before You Start

Ensure that you have already:

- Added a registry and uploaded a model image to that registry
- Added a packaged model

WARNING

Ephemeral Storage Considerations

Default disk sizes on cloud providers may not be sufficient for large models, which often require significant ephemeral storage. This can result in the inference service failing to start serving—in some instances, without providing an error message about being out of disk space.

Ephemeral storage is normally provided by the boot disk of the compute nodes. You can inspect the amount of ephemeral storage on your nodes using the kubectl describe node `<node-name>` command.

NOTE

Improve Startup Time

For optimal startup time, use a custom container image. This combines code and model in one container, eliminating model download time, which is important for scaling inference services. You can build a custom container image using BentoMLor OpenLLM's `build --containerize` command, then reference it from a packaged model using the custom model format and the image container.

Create a Deployment Via the UI

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to **Deployments**.
4. Select **Create new deployment**.
5. Provide a **Deployment Name**.
6. Select **Next**.
7. Choose a **Packaged Model** from the dropdown.
8. Select **Next**.
9. Select a Kubernetes **Namespace** from the dropdown. If this option is not available, the Namespace is already set for you.
10. Optionally, toggle endpoint security. If enabled, all endpoint interactions will required a **deployment token** in the header (e.g., "Authorization: Bearer <YOUR_ACCESS_TOKEN>"). In some cases, endpoint security is enabled by default and cannot be disabled.
11. Select **Next**.
12. Optionally, specify a node selector label and value, and select a deployment priority (not available to all clusters). Select **Next**.
13. Choose an Auto scaling targets template or provide custom values for all of the following:
    - **Auto scaling target templates**:

name	description	autoscaling_min_replicas	autoscaling_max_replicas	autoscaling_metric	autoscaling_target
fixed-1	One inference service replica, always available.	1	1	rps	0
fixed-2	Two inference service replicas, always available.	2	2	rps	0
scale-0-to-1-concurrency-3	Scale from 0 to 1 replicas with metric concurrency 3.	0	1	concurrency	3
scale-0-to-4-rps-10	Scale from 0 to 4 replicas metric with requests-per-second 10.	0	4	rps	10
scale-0-to-8-rps-20	Scale from 0 to 8 replicas metric with requests-per-second 20.	0	8	rps	20
scale-1-to-4-rps-10	Scale from 1 to 4 replicas metric with requests-per-second 10.	1	4	rps	10
scale-1-to-8-concurrency-3	Scale from 1 to 8 replicas with metric concurrency 3.	1	8	concurrency	3

    - **Minimum instances**: The minimum number of instances to run.
    - **Maximum instances**: The maximum number of instances to run.
    - **Auto scaling target**: The target **metric** and `metric-value` to trigger scaling. The possible metric types that can be configured per revision depend on the type of Autoscaler implementation you are using:
      - The default **KPA Autoscaler** supports the `concurrency` and `rps` metrics.
      - The **HPA Autoscaler** supports the `cpu` metric.
14. Select **Next**.
15. Provide any needed **Environment Variables** or **Arguments**. See the Advanced Configuration Options reference article for more information.
16. Select **Done**.

Create a Deployment Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign into HPE Machine Learning Inferencing Software (MLIS).

```
aioli auth login <YOUR_USERNAME>
```

2. Create a new deployment with the following command:

```
aioli deployment create <DEPLOYMENT_NAME> \
  --model <PACKAGED_MODEL_NAME> \
  --namespace <K8S_NAMESPACE> \
```

```
  --authentication-required <BOOLEAN> \
  --auto-scaling-max-replicas <MAX_REPLICAS> \
  --auto-scaling-min-replicas <MIN_REPLICAS> \
  --auto-scaling-metric <METRIC> \
  --auto-scaling-target <TARGET_VALUE> \
  --environment <VAR_1>=<VALUE_1> <VAR_2>=<VALUE_2> \
  --arguments <ARG_1> <ARG_2>
```

3. Wait for your deployment to reach `Ready` state.

```
aioli deployment show <DEPLOYMENT_NAME>
```

- For more information on the `aioli deployment create` command, see the **CLI command reference**.
- For information on setting environment variables and arguments, see the Advanced Configuration Options reference article.

Create a Deployment Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.
3. Use the following cURL command to add a new deployment.

```
curl -X 'POST' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/deployments' \
    -H 'accept: application/json' \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
    -d '{
      "arguments": [
        "--debug"
      ],
      "autoScaling": {
        "maxReplicas": <MAX_REPLICAS>,
        "metric": "<METRIC>",
        "minReplicas": <MIN_REPLICAS>,
        "target": <TARGET_VALUE>
      },
      "canaryTrafficPercent": <CANARY_PERCENT>,
      "environment": {
        "<VAR_1>": "<VALUE_1>",
        "<VAR_2>": "<VALUE_2>"
      },
      "goalStatus": "<GOAL_STATUS>",
      "model": "<PACKAGED_MODEL_NAME>",
      "name": "<DEPLOYMENT_NAME>",
      "namespace": "<K8S_NAMESPACE>",
      "security": {
        "authenticationRequired": <BOOLEAN>
      }
    }'
```

Edit Deployment

You can make changes to certain aspects of an active deployment, such as the packaged model, authentication requirements, and auto-scaling configurations. If you change the packaged model, you must choose between an abrupt upgrade or a canary rollout to handle the change.

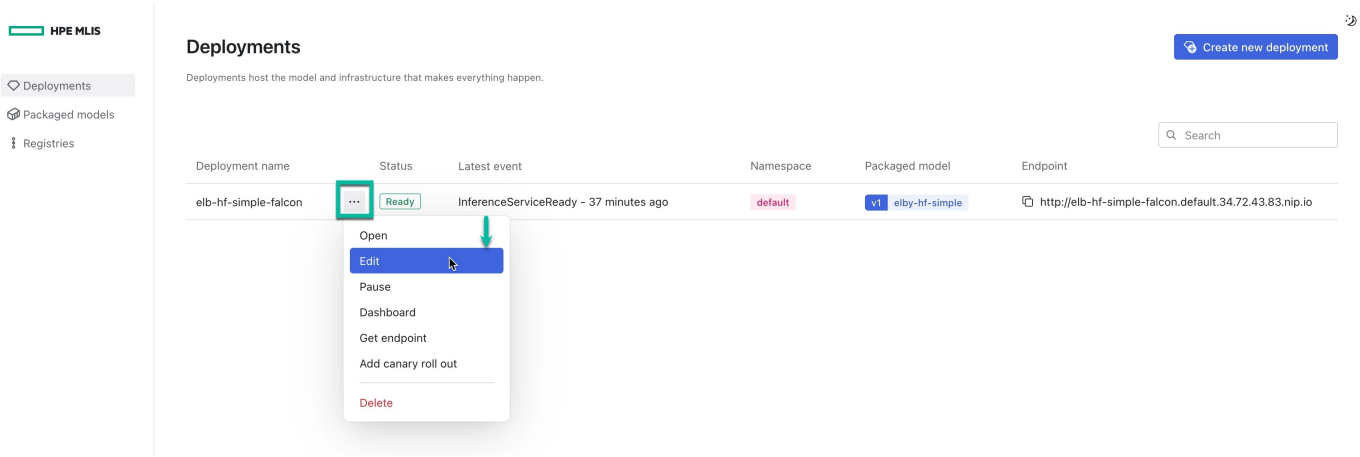
Before You Start

- You must have already created a deployment.
- You must have either a new packaged model or a new version of the existing deployed packaged model that you want to test.

> **IMPORTANT**
>
> KServe is designed to facilitate non-disruptive rollouts of inference service changes. KServe integrates with Knative Serving to achieve non-disruptive rollouts. KServe always waits for the new inference service revision to be ready before removing the old revision. Rolling out an edit to a deployment uses 2x resources. To circumvent this KServe feature due to limited resources, such as limited GPUs, use the pattern Pause/Edit-Deployment/Resume instead of simply editing the deployment.

Edit a Deployment Via the UI

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to **Deployments**.
4. Select the **ellipsis** icon next to the deployment you want to edit.
5. Select **Edit**.



6. Make the desired changes to the deployment.
7. Select **Done**.

Edit a Deployment Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
aioli auth login <YOUR_USERNAME>
```

2. Update a deployment with the following command:

```
aioli deployment update <DEPLOYMENT_NAME> \
  --model <PACKAGED_MODEL_NAME> \
  --canary-percentage <CANARY_PERCENTAGE> \
  --authentication-required <BOOLEAN> \
  --auto-scaling-max-replicas <MAX_REPLICAS> \
  --auto-scaling-min-replicas <MIN_REPLICAS> \
  --auto-scaling-metric <METRIC> \
  --auto-scaling-target <TARGET_VALUE> \
  --environment <VAR_1>=<VALUE_1> <VAR_2>=<VALUE_2> \
  --arguments <ARG_1> <ARG_2>
```

3. Wait for your deployment to reach `Ready` state.

```
aioli deployment show <DEPLOYMENT_NAME>
```

- For more information on the `aioli deployment update` command, see the **CLI command reference**.
- For information on setting environment variables and arguments, see the Advanced Configuration Options reference article.

Edit a Deployment Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
 '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
 -H 'accept: application/json' \
 -H 'Content-Type: application/json' \
 -d '{
 "username": "<YOUR_USERNAME>",
 "password": "<YOUR_PASSWORD>"
 }'
```

2. Obtain the Bearer token from the response.
3. Get a list of deployments to find the deployment ID.

```
curl -X 'GET' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/deployments' \
    -H 'accept: application/json' \
    -H 'Authorization : Bearer <YOUR_ACCESS_TOKEN>'
```

4. Get the deployment ID from the response.
5. Use the following curl command to update a deployment.

```
curl -X 'PUT' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/deployments/<DEPLOYMENT_ID>' \
    -H 'accept: application/json' \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
    -d '{
      "arguments": [
        "--debug"
      ],
      "autoScaling": {
        "maxReplicas": <MAX_REPLICAS>,
        "metric": "<METRIC>",
        "minReplicas": <MIN_REPLICAS>,
        "target": <TARGET_VALUE>
      },
      "canaryTrafficPercent": <CANARY_TRAFFIC_PERCENT>,
      "environment": {
        "<VAR_1>": "<VALUE_1>",
        "<VAR_2>": "<VALUE_2>"
      },
      "goalStatus": "<GOAL_STATUS>",
      "model": "<PACKAGED_MODEL_NAME>",
      "security": {
        "authenticationRequired": <BOOLEAN>
      }
    }'
```

Interact with a Deployment

You can interact with a deployed inference service by sending a request to the service's endpoint. The service will respond with a prediction based on the input data.

Before You Start

- Ensure you have completed the **Developer Environment Setup**.
- Ensure that you have an active inference service deployed and in a `Ready` state (`aioli d`).
- Depending upon your Kubernetes configuration, you may need to set up:
  - A DNS domain name or **Magic DNS configuration** to access the service via the endpoint hostname.
  - A `port-forward` to enable interactions with your service to perform inference requests. See the **Kserve Documentation** for more information.

Interact with a Service Using Service Endpoint Hostname

If a DNS or **Magic DNS** has been configured, you can interact with the service using the endpoint hostname provided in the UI or CLI output after deploying the service. For example, the endpoint hostname `http://iris-classifier-deployment.default.example.com` can be used to interact with the service.

```
curl -s \
-H "Authorization: Bearer <YOUR_ACCESS_TOKEN>" \
-H Content-Type:application/json http://iris-classifier-
deployment.default.example.com/classify \
-d   [[5.9, 3, 5.1, 1.8]]
```

For information on generating an API token, see Inference Model Endpoints.

Determining the API Route

The API route used (in this guide, `/classify`) is specific to the service and may vary based on the service's implementation. For example, if the model image was created using `bentoml`, the API route would be defined in the `service.py` file.

```
from typing import Dict, Any
import numpy as np
import bentoml
from bentoml.io import NumpyNdarray, JSON
import torch

iris_clf_runner = bentoml.pytorch.get("iris:latest").to_runner()

svc = bentoml.Service("iris", runners=[iris_clf_runner])

@svc.api(input=NumpyNdarray(), output=JSON())
async def classify(input_array: np.ndarray) -> Dict[str, Any]:
    # Convert the numpy array to a Float Tensor
    input_tensor = torch.FloatTensor(input_array)
```

```
    # Run the prediction. Since it is not an async function, we use `run` instead of `async_run`
    result_tensor = await iris_clf_runner.async_run(input_tensor)

    # Convert the output tensor to numpy for JSON serialization
    result_array = result_tensor.numpy()

    # Assuming the model outputs class probabilities, we take the argmax to get the class label
    class_label = np.argmax(result_array, axis=1)[0]

    return {"prediction": class_label} # Convert to list for JSON serialization
```

Initiate Canary Rollout

Canary rollouts enable you to safely test your new model version's performance alongside an existing model version that is currently deployed by directing a percentage of the traffic to the new model. This allows you to compare the performance of the new model with the existing model before fully deploying the new model.

Before You Start

- You must have already created a deployment.
- You must have either a new packaged model or a new version of the existing deployed packaged model that you want to test.

Rollout Behavior

How a canary rollout behaves depends on the traffic percentage (`--canary-traffic-percent`) set on the deployment for the new model version. The traffic percentage you provide maps to the following behaviors:

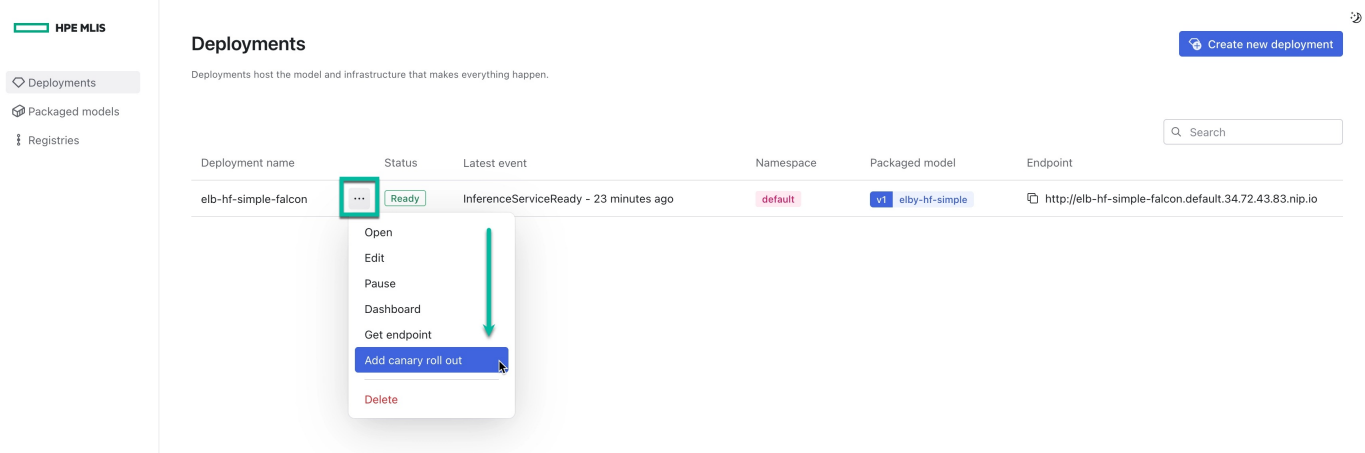
- `100%` : Triggers a full, immediate rollout to the new version instead of performing a canary rollout. The prior model version stops serving traffic as soon as the new version is properly serving requests.
- `1-99%` : Triggers a canary rollout where both versions serve requests; the defined canary traffic percentage is sent to the new instance once it has become ready.
- `0%` : Cancels the canary rollout and reverts all traffic back to the prior model version deployed.

Once you are satisfied with the new model version's performance, complete the rollout by updating the canary traffic percentage to `100%` .

> **NOTE**
>
> **Canceled Canary Rollouts**
>
> If you cancel a canary rollout and set its traffic to `0%` , the inference service will remain present but handle no requests and continue to consume any GPU resources assigned until you perform another rollout. This can be either a `100%` roll out of the prior version, or a new rollout of another model version.

How to Initiate a Canary Rollout Via the UI

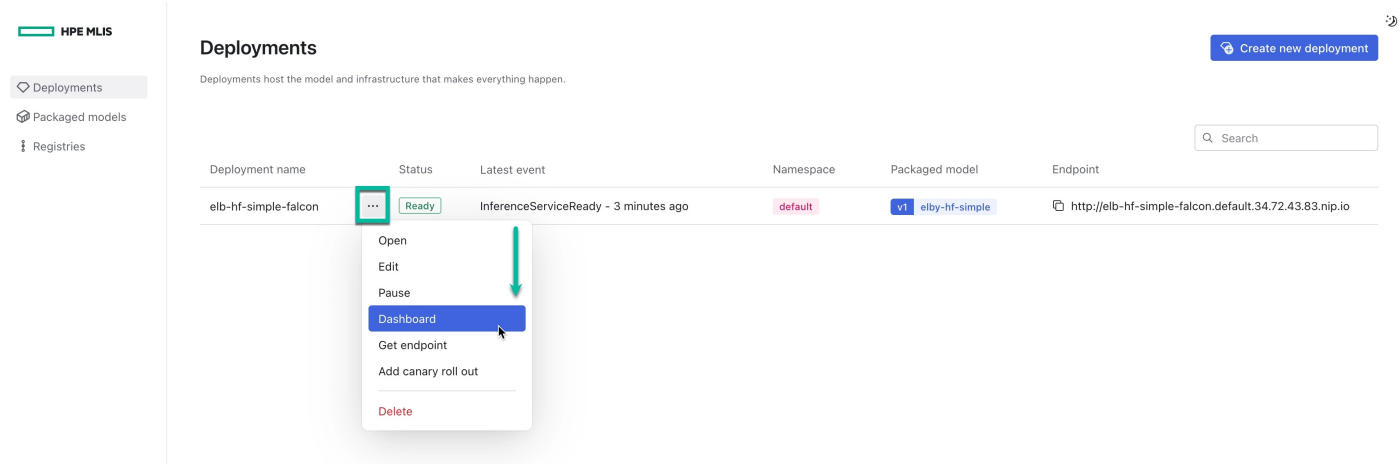
1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Navigate to **Deployments**.
4. Select the **ellipsis** next to the deployment you want to initiate a canary rollout for.
5. Select **Add canary rollout**.



6. Select which **Packaged Model** you want to roll out.
7. Select how much traffic you want to direct to the new model as a percentage.

8. Select **Done**.

The rollout proccess will begin immediately. You can monitor and analyze the traffic between the two models by navigating to your deployment's Grafana dashboard. See Grafana.



How to Initiate a Canary Rollout Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
aioli auth login <YOUR_USERNAME>
```

2. Update a deployment with the following command:

```
aioli deployment update <DEPLOYMENT_NAME> \
  --model <PACKAGED_MODEL_NAME> \
  --canary-percentage <CANARY_PERCENTAGE>
```

3. Wait for your deployment to reach `Ready` state.

```
aioli deployment show <DEPLOYMENT_NAME>
```

```
arguments: []
autoScaling:
maxReplicas: 1
metric: rps
minReplicas: 1
target: 0
canaryTrafficPercent: 5
environment: {}
goalStatus: Ready
id: 857ac422-ccf3-42a9-8d69-5af4cdc859dc
model: elby-hf-simple
modifiedAt: '2024-05-20T18:56:41.384852Z'
name: elb-hf-simple-falcon
namespace: default
secondaryState:
endpoint: http://prev-elby-hf-simple-falcon-predictor.default.34.72.43.83.nip.io
nativeAppName: elby-hf-simple-falcon-predictor-00001
status: Ready
trafficPercentage: 95
security:
authenticationRequired: false
state:
endpoint: http://elby-hf-simple-falcon.default.34.72.43.83.nip.io
nativeAppName: elby-hf-simple-falcon-predictor-00002
status: Ready
trafficPercentage: 5
status: Ready
```

How to Initiate a Canary Rollout Via the API

1. Sign in to HPE Machine Learning Inferencing Software (MLIS).

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
```

```
}'
```

2. Obtain the Bearer token from the response.
3. Get a list of deployments to find the deployment ID.

```
curl -X 'GET' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/deployments' \
    -H 'accept: application/json' \
    -H 'Authorization : Bearer <YOUR_ACCESS_TOKEN>'
```

4. Get the deployment ID from the response.
5. Use the following cURL command to submit a canary rollout put request.

```
curl -X 'PUT' \
    '<YOUR_EXT_CLUSTER_IP>/api/v1/deployments/{id}' \
    -H 'accept: application/json' \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
    -d '{
      "model": "<PACKAGED_MODEL_NAME>",
      "canaryPercentage": <CANARY_PERCENTAGE>
    }'
```

Monitor Services

HPE Machine Learning Inferencing Software (MLIS) automatically configures monitoring of all deployed AI inference services.

See Observability.

Inference Service Labels

Your inference service can be identified via one of its labels.

Label Name	Value	Description
`serving.kserve.io/inferenceservice`	The deployment name.	Selects all instances of all versions of your inference service. Selectable in the Deployment Dashboard via the `Deployment Name` dropdown.
`inference/packaged-model`	The packaged model name and version. For example: `fb125m-model.v1`	Selectable in the Deployment Dashboard via the `Packaged Model Version` dropdown. By default, all versions of the deployment are shown.
`inference/deployment-id`	The deployment's `id` value.	For advanced use. Normally `serving.kserve.io/inferenceservice` is used as long as deployment names are not reused for different instances.
`inference/packaged-model-id`	The packaged model's `id` value.	For advanced use. Normally `inference/packaged-model` is used as long as packaged model names are not reused for different instances.

NOTE

Label Names in Grafana

These labels may be re-formatted by Grafana to use underscores instead of slashes (`/`), hyphens (`-`), and periods (`.`), for example, `inference/packaged-model` becomes `inference_packaged_model` .

Subtopics

View Metrics (Prometheus)

HPE Machine Learning Inferencing Software (MLIS) facilitates access to metrics generated by the underlying components used in your deployed model services (e.g., Kubernetes, KServe, BentoML, OpenLLM, and NIM). These components provide numerous metrics out of the box. The specific metrics available for your deployment depend on the type of model you deploy and the configuration of your system.

View Metrics (Prometheus)

HPE Machine Learning Inferencing Software (MLIS) facilitates access to metrics generated by the underlying components used in your deployed model services (e.g., Kubernetes, KServe, BentoML, OpenLLM, and NIM). These components provide numerous metrics out of the box. The specific metrics available for your deployment depend on the type of model you deploy and the configuration of your system.

Before You Start

Review the following information before you start:

- All containers get **Kserve and Knative metrics**.
- Any metrics surfaced via `/metrics` on the container port (`8080`) are also collected.
- MLIS does not generate any metrics of its own.

How to View Metrics

See Prometheus.

Metrics: BentoML & OpenLLM

The following table lists some metrics that are commonly useful when monitoring BentoML and OpenLLM deployments.

- For more information on BentoML metrics, see **the official documentation**.
- For details about metrics provided by the vLLM backend of BentoML and OpenLLM, see the **vLLM documentation**.

Description	Metric Name	Metric Type	Dimensions
API Server request in progress	`bentoml_api_server_request_in_progress`	Gauge	`endpoint`, `service_name`, `service_version`
Runner request in progress	`bentoml_runner_request_in_progress`	Gauge	`endpoint`, `runner_name`, `service_name`, `service_version`
API Server request total	`bentoml_api_server_request_total`	Counter	`endpoint`, `service_name`, `service_version`, `http_response_code`
Runner request total	`bentoml_runner_request_total`	Counter	`endpoint`, `service_name`, `runner_name`, `service_version`, `http_response_code`
API Server request duration in seconds	`bentoml_api_server_request_duration_seconds_sum`, `bentoml_api_server_request_duration_seconds_count`, `bentoml_api_server_request_duration_seconds_bucket`	Histogram	`endpoint`, `service_name`, `service_version`, `http_response_code`
Runner request duration in seconds	`bentoml_runner_request_duration_seconds_sum`, `bentoml_runner_request_duration_seconds_count`, `bentoml_runner_request_duration_seconds_bucket`	Histogram	`endpoint`, `service_name`, `runner_name`, `service_version`, `http_response_code`
Runner adaptive batch size	`bentoml_runner_adaptive_batch_size_sum`, `bentoml_runner_adaptive_batch_size_count`, `bentoml_runner_adaptive_batch_size_bucket`	Histogram	`method_name`, `service_name`, `runner_name`, `worker_index`

Metrics: KServe & Knative

See the **Knative observability documentation** for more information on metrics that can be made available for your deployment.

> **NOTE**
> Some of the metrics mentioned in the Knative observability documentation may require additional configuration to be available in your deployment.

Metrics: NIM

Some Nvidia NIM containers also provide metrics, such as the `llama3` container. However, other reranking and embedding NIMs currently do not. See **the official documentation** for details.

Advanced Configuration Options

While adding or editing a deployment, you can customize the default runtime configuration MLIS uses to start your model server.

This is useful for models that require modifications to MLIS's default settings or when using a custom image that needs specific configuration parameters.

> **NOTE**
> If you want to use a fully custom container image instead of simply modifying the default runtime configuration, see Customize the Base Container Image.

Before You Start

- You should review your model's framework documentation (e.g., OpenLLM, BentoML, NIM) and its CLI options.
- You should already know the arguments and environment variables you'd like to set for your model based on previous testing.

Runtime Configuration Options

Environment Variables

Variable	Description
`AIOLI_LOGGER_PORT`	The port that the logger service listens on; default is `49160`.
`AIOLI_PROGRESS_DEADLINE`	The deadline for downloading the model; default is `1500s`.
`AIOLI_READINESS_FAILURE_THRESHOLD`	The number of readiness probe failures before the deployment is considered unhealthy; default is `100`.
`AIOLI_COMMAND_OVERRIDE`	The customized deployment command that enables you to override the default deployment command within a predefined runtime.
`AIOLI_SERVICE_PORT`	The inference service container port used for communication; default is `8080` except for NIMs, which is `8000`.
`AIOLI_DISABLE_MODEL_CACHE`	Disables automatic model caching for a deployment, even if it is enabled for the model; default is `false`.
`AIOLI_DISABLE_LOGGER`	A workaround for the Kserve defect concerning streamed responses.

Command Override Arguments

MLIS executes a default command for your container runtime based on the type of packaged model you have selected. However, you can modify this command using the `AIOLI_COMMAND_OVERRIDE` environment variable. Any arguments from the packaged model are then appended to the end of this command, followed by any arguments from the deployment (`AI`

OLI_COMMAND_OVERRIDE = [CLI_COMMAND] [MODEL_ARGS] [DEPLOYMENT_ARGS]).

Default Commands

The following table shows the default command for each packaged model's framework type:

FRAMEWORK	COMMAND	DESCRIPTION
OpenLLM	`openllm start --port {{.containerPort}} {{.modelDir}}`	You can add any options from OpenLLM version `0.4.44` to your command (see `openllm start -h`).
Bento Archive	`bentoml serve ...`	You can add any options from BentoML version `1.1.11` to your command (see `bentoml serve -h`).
Custom	`none`	For custom models, the default entrypoint for the container is executed.
NVIDIA NIM	`none`	For NIM models, the default entrypoint for the container is executed. You must use **environment variables**; NIM contaiers do not honor CLI arguments.
vLLM	`--model {{.modelName}} --port {{.containerPort}} --download-dir {{.modelDir}} --tensor-parallel-size {{.numGpus}}`	Arguments vary for S3/PVC/PFS URLs.

Template Variables

MLIS provides template variables for customizing the runtime command:

Named Argument	Description
`{{.numGpus}}`	The number of GPUs the model is requesting.
`{{.modelName}}`	The model name being deployed. For Openllm/vLLM the model name path from the url.
`{{.modelDir}}`	The directory into which the model will be downloaded. This is typically `/mnt/models` . This applies to NIM, OpenLLM, and S3 models.
`{{.containerPort}}`	The http port that the container must listen on for inference requests and readiness checks.

Examples

```
AIOLI_COMMAND_OVERRIDE="openllm start {{.modelName}} --port
{{.containerPort}} --gpu-memory-utilization 0.9 --max-total-tokens 4096"
```

```
AIOLI_COMMAND_OVERRIDE="bentoml serve {{.modelDir}}/bentofile.yaml --
production --port {{.containerPort}} --host 0.0.0.0"
```

Manage Auto Scaling Templates

HPE Machine Learning Inferencing Software (MLIS) includes default autoscaling templates that users can select when adding or editing a deployment. You can manage these auto scaling templates and create new ones using the MLIS UI, CLI, or API.

Before You Start

- You must have the **Admin** **user role** to manage auto scaling templates.

Default Auto Scaling Templates

The following table shows the default auto scaling templates available in MLIS.

name	description	autoscaling_min_replicas	autoscaling_max_replicas	autoscaling_metric	autoscaling_target
fixed-1	One inference service replica, always available.	1	1	rps	0
fixed-2	Two inference service replicas, always available.	2	2	rps	0
scale-0-to-1-concurrency-3	Scale from 0 to 1 replicas with metric concurrency 3.	0	1	concurrency	3
scale-0-to-4-rps-10	Scale from 0 to 4 replicas metric with requests-per-second 10.	0	4	rps	10
scale-0-to-8-rps-20	Scale from 0 to 8 replicas metric with requests-per-second 20.	0	8	rps	20
scale-1-to-4-rps-10	Scale from 1 to 4 replicas metric with requests-per-second 10.	1	4	rps	10
scale-1-to-8-concurrency-3	Scale from 1 to 8 replicas with metric concurrency 3.	1	8	concurrency	3

Auto Scaling Templates Via the UI

1. In the MLIS UI, navigate to Settings > Auto scaling templates.
2. Select Add new auto scaling template.
3. Input the name, description, and autoscaling requirements for the template.

Field	Description
Template name	A unique identifier for the auto scaling template
Description	A brief explanation of the template's purpose or characteristics
Minimum instances	The lowest number of instances that will run, even during periods of low activity
Maximum instances	The highest number of instances allowed to run during peak demand
Auto scaling target	The metric and target value used to trigger scaling actions
- Metric	The type of metric to monitor (e.g., concurrency, CPU utilization, memory usage)
- Target	The threshold value for the chosen metric that triggers scaling actions

NOTE

The available metric types depend on the Autoscaler implementation:

- KPA Autoscaler (default): Supports `concurrency` and `rps` metrics
- HPA Autoscaler: Supports the `cpu` metric

4. Select **Create template**.

The new template is now available from the Auto scaling targets template dropdown on the Scaling tab when adding or editing a deployment. To update this template, select the ellipsis icon next to the template name and choose Edit.

Auto Scaling Templates Via the CLI

See Developer Environment Setup for information on using the CLI.

1. Sign in via the CLI.

```
aioli auth login admin
```

2. Add a new resource template with the following command:

```
aioli templates autoscaling create <TEMPLATE_NAME> \
--autoscaling-min-replicas <MIN_REPLICAS> \
--autoscaling-max-replicas <MAX_REPLICAS> \
--autoscaling-metric <METRIC> \
--autoscaling-target <TARGET>
```

Auto Scaling Templates Via the API

1. Sign in to MLIS.

```
curl -X 'POST' \
  '<YOUR_EXT_CLUSTER_IP>/api/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "<YOUR_USERNAME>",
  "password": "<YOUR_PASSWORD>"
}'
```

2. Obtain the Bearer token from the response.

3. Use the following cURL command to add a new auto scaling template.

```
curl -X 'POST' \
  'https://<YOUR_EXT_CLUSTER_IP>/api/v1/templates/autoscaling' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer <YOUR_ACCESS_TOKEN>' \
  -H 'Content-Type: application/json' \
  -d '{
    "autoScaling": {
      "maxReplicas": 1,
      "metric": "rps",
      "minReplicas": 0,
      "target": 1
    },
    "description": "An autoscaling template",
    "name": "my-template"
}'
```

Configuring HTTPS/TLS Certificate Trust for External Repositories

Some external repositories may use certificates that are either self-signed or issued by non-public Certificate Authorities (CAs), making them untrusted by default. This guide provides instructions on configuring MLIS to recognize and trust these certificates. Note these steps are not required for external repositories such as `huggingface.co` or `s3.amazonaws.c`

`om`, which use valid SSL/TLS certificates issued by trusted certificate authorities (CAs) for their website.

Before You Start

- Obtain the self-signed certificate or the certificate issued by the non-public CA from the external repository you want to trust.

How to Configure Custom Certificates for External Repository Access

To add custom certificates to the MLIS controller and deployment pods, you have two options:

1. **Manually create ConfigMaps:** Add certificates by creating ConfigMaps that contain your certificates directly in the namespace where the MLIS controller is installed, as well as in the namespaces where deployments will be made.
2. **Use Trust Manager for automated ConfigMap creation:** Configure Trust Manager to automatically create and distribute a ConfigMap containing the necessary certificates across the desired namespaces.

For configuring Trust Manager, refer to the **Trust Manager documentation** to control which namespaces will automatically receive the ConfigMap.

Once the ConfigMap containing your certificates is ready, either manually or via Trust Manager, specify its name as shown below:

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Specify the name of the ConfigMap that contains the trusted CAs you want to inject into the environment of your deployment.

   ```
   trustedCAsConfigMap: ""
   ```

   See **trust-manager** for details on how to create a ConfigMap with trusted CAs.

Once deployed, the certificates in the bundle will be automatically mounted into:

- The controller
- All newly created deployment pods

Example: Configuring MLIS to Trust a Self-Signed MinIO Certificate Using Trust Manager

This example demonstrates how to configure MLIS to trust a self-signed certificate named `minio-self-signed.crt` for a MinIO server. It assumes you have already installed Trust Manager in your cluster.

1. **Create a Secret for the Certificate**

   Create a secret named `minio-cert-secret` in the `cert-manager` namespace that contains the self-signed certificate.

   ```
   kubectl create secret generic minio-cert-secret --from-file=./minio-self-signed.crt -n
   cert-manager
   ```

   Example output:

   ```
   secret/minio-cert-secret created
   ```

2. **Verify the Secret's Contents**

   Use the following command to examine the contents of the `minio-cert-secret`:

   ```
   kubectl describe secret minio-cert-secret -n cert-manager
   ```

   Example output:

   ```
   Name:         minio-cert-secret
   Namespace:    cert-manager
   Labels:       <none>
   Annotations:  <none>

   Type:  Opaque

   Data
   ====
   minio-self-signed.crt:  1302 bytes
   ```

3. **Create a Trust Manager Bundle Configuration**

   Create a file named `mlis-trust-manager-bundle.yaml` with the following contents. The `spec.target.configMap.key` specifies the filename that will be created under the `/etc/ssl/certs/` directory in the container. Use the `useDefaultCAs: true` option to specify that the bundle should include the system's default trusted Certificate Authorities (CAs) along with any additional certificates you specify.

   ```
   apiVersion: trust.cert-manager.io/v1alpha1
   kind: Bundle
   metadata:
     name: mlis-trust-manager-bundle
   spec:
     sources:
     - useDefaultCAs: true
     - secret:
         name: minio-cert-secret
         key: minio-self-signed.crt
     target:
   ```

```
  configMap:
    key: "ca-certificates.crt"
```

4. **Apply the Trust Manager Bundle**

   Apply the configuration to create the ConfigMap in the namespaces.

   ```
   kubectl apply -f mlis-trust-manager-bundle.yaml
   ```

   Example output:

   ```
   bundle.trust.cert-manager.io/mlis-trust-manager-bundle created
   ```

5. **Configure to Use the ConfigMap**

   Specify the ConfigMap:

   a. Under Tools & Frameworks, select Configure on the HPE MLIS tile.

   b. Specify the name of the ConfigMap that contains the trusted CAs you want to inject into the environment of your deployment.

   ```
   trustedCAsConfigMap: ""
   ```

   See **trust-manager** for details on how to create a ConfigMap with trusted CAs.

Example: Configuring MLIS to Trust a Self-Signed MinIO Certificate Using a Manually Created ConfigMap

1. **Create the ConfigMap in the MLIS controller's namespace, as well as in the namespaces where deployments will be made.**

   ```
   kubectl create configmap my-cert-configmap --from-file=ca-certificates.crt=minio-
   self-signed.crt -n <MLIS_NAMESPACE>
   ```

   ```
   kubectl create configmap my-cert-configmap --from-file=ca-certificates.crt=minio-
   self-signed.crt -n <DEPLOYMENT_NAMESPACE_1>
   kubectl create configmap my-cert-configmap --from-file=ca-certificates.crt=minio-
   self-signed.crt -n <DEPLOYMENT_NAMESPACE_2>
   ...
   kubectl create configmap my-cert-configmap --from-file=ca-certificates.crt=minio-
   self-signed.crt -n <DEPLOYMENT_NAMESPACE_N>
   ```

   > **NOTE**
   >
   > To include public certificates along with your MinIO certificate, you can combine them into a single file before creating the ConfigMap:

   ```
   cat /etc/ssl/certs/ca-certificates.crt minio-self-signed.crt > combined-certs.crt
   ```

   Then, create the ConfigMap with the combined certificates:

   ```
   kubectl create configmap my-cert-configmap --from-file=ca-
   certificates.crt=combined-certs.crt -n <YOUR_NAMESPACE>
   ```

2. **Configure to Use the ConfigMap**

   Specify the ConfigMap:

   a. Under Tools & Frameworks, select Configure on the HPE MLIS tile.

   b. Specify the name of the ConfigMap that contains the trusted CAs you want to inject into the environment of your deployment.

   ```
   trustedCAsConfigMap: ""
   ```

   See **trust-manager** for details on how to create a ConfigMap with trusted CAs.

Verifying the Certificate in the MLIS Controller

To ensure the certificate was mounted and is being recognized by the MLIS controller, perform the following verification steps:

1. Extract the first line of your certificate, excluding the BEGIN CERTIFICATE line:

   ```
   awk '/BEGIN CERTIFICATE/{found=1; next} found {print; exit}' minio-self-signed.crt
   ```

   Example output:

   ```
   MIIDlDCCAnygAwIBAgIUFvnJdhuBfM6V94FG4KPEt3t68P0wDQYJKoZIhvcNAQEL
   ```

2. Check that the certificate is present in the MLIS controller's /etc/ssl/certs/ca-certificates.crt file:

   ```
   kubectl exec -it svc/aioli-master-service-aioli -- grep
   "MIIDlDCCAnygAwIBAgIUFvnJdhuBfM6V94FG4KPEt3t68P0wDQYJKoZIhvcNAQEL"
   /etc/ssl/certs/ca-certificates.crt
   ```

   Example output:

   ```
   MIIDlDCCAnygAwIBAgIUFvnJdhuBfM6V94FG4KPEt3t68P0wDQYJKoZIhvcNAQEL
   ```

Environment Variables

Depending on the tool or application used to download models from external repositories that use self-signed or non-public CA certificates, you may need to set environment variables that specify the location of trusted certificates.

For example, if you're using the `kserve:storage-initializer` container to download models from S3 storage, set the environment variable `AWS_CA_BUNDLE=/etc/ssl/certs/ca-certificates.crt` in your packaged model or deployment's environment settings. This example assumes `ca-certificates.crt` is the key used in your ConfigMap, though the actual value should match the key specified in your custom ConfigMap.

> **NOTE**
> For version `kserve/storage-initializer:v0.11.2`, only the `AWS_CA_BUNDLE` environment variable is required. For newer versions, you also need to set the `CA_BUNDLE_CONFIGMAP_NAME` variable to any non-empty value. This additional variable is necessary due to kserve's custom CA bundle configuration requirements. For more information, refer to the **Kserve storage-initializer configuration instructions**.

Other common environment variables for specifying CA bundle locations include `REQUESTS_CA_BUNDLE` and `SSL_CERT_FILE`, depending on the application being used for model downloads.

Environment Variables

CLI

The following environment variables can be set to configure the CLI:

Variable	Description
`AIOLI_USER`	The username used to authenticate the user.
`AIOLI_PASS`	The password used to authenticate the user.
`AIOLI_USER_TOKEN`	The token used to authenticate the user. See Auth Tokens.
`AIOLI_CONTROLLER`	The protocol, IP, and port of the controller (e.g., `https://mlis.{AIE-DOMAINAME}`).
`AIOLI_CONTROLLER_CERT_FILE`	The path to the certificate file used by the controller. You can get this value by either of the following way: • Down load this file using `https://${DOMAIN}/ca` via web browser. Or • To get it from Kubernetes: 1. Run `kubectl get cm -n ezua-system ezaf-root-ca -o yaml | yq '.data."ezaf-root-ca.crt"' > ezaf-root-ca.crt`. 2. Set as `AIOLI_CONTROLLER_CERT_FILE=ezaf-root-ca.crt`.
`AIOLI_CONTROLLER_CERT_NAME`	The name of the controller's certificate file.
`AIOLI_DEBUG_CONFIG_PATH`	The path to the debug configuration file.

Deployment & Packaged Model

The following environment variables can be set while adding a packaged model or creating a deployment to modify the default settings:

Variable	Description
`AIOLI_LOGGER_PORT`	The port that the logger service listens on; default is `49160`.
`AIOLI_PROGRESS_DEADLINE`	The deadline for downloading the model; default is `7200s`.
`AIOLI_READINESS_FAILURE_THRESHOLD`	The number of readiness probe failures before the deployment is considered unhealthy; default is `100`.
`AIOLI_COMMAND_OVERRIDE`	The customized deployment command that enables you to override the default deployment command within a predefined runtime.
`AIOLI_SERVICE_PORT`	The inference service container port used for communication; default is `8080` except for NIMs, which is `8000`.
`AIOLI_DISABLE_MODEL_CACHE`	Disables automatic model caching for a deployment, even if it is enabled for the model; default is `false`.
`AIOLI_DISABLE_LOGGER`	Default is `1`. To enable MLIS request/response logging, specify the value as `0` or `false`. See Kserve defect concerning streamed responses for workaround details.

Environment variables set on a deployment will override the values set on its packaged model.

Command Override Argument Options

MLIS executes a default command for your container runtime based on the type of package model you have selected. However, you can modify this command using the `AIOLI_COMMAND_OVERRIDE` environment variable. Any arguments from the packaged model are then appended to the end of this command, followed by any arguments from the deployment (`AIOLI_COMMAND_OVERRIDE = [CLI_COMMAND] [MODEL_ARGS] [DEPLOYMENT_ARGS]`).

The following table shows the default command for each packaged model's framework type:

FRAMEWORK	COMMAND	DESCRIPTION
OpenLLM	`openllm start --port {{.containerPort}} {{.modelDir}}`	You can add any options from OpenLLM version `0.4.44` to your command (see `openllm start -h`).
Bento Archive	`bentoml serve ...`	You can add any options from BentoML version `1.1.11` to your command (see `bentoml serve -h`).
Custom	`none`	For custom models, the default entrypoint for the container is executed.
NVIDIA NIM	`none`	For NIM models, the default entrypoint for the container is executed. You must use **environment variables**; NIM contaiers do not honor CLI arguments.
vLLM	`--model {{.modelName}} --port {{.containerPort}} --download-dir {{.modelDir}} --tensor-parallel-size {{.numGpus}}`	Arguments vary for S3/PVC URLs.

You can also use the following variables to modify the command's arguments:

Named Argument	Description
`{{.numGpus}}`	The number of GPUs the model is requesting.
`{{.modelName}}`	The model name being deployed. For Openllm/vLLM the model name path from the url.
`{{.modelDir}}`	The directory into which the model will be downloaded. This is typically `/mnt/models`. This applies to NIM, OpenLLM, and S3 models.
`{{.containerPort}}`	The http port that the container must listen on for inference requests and readiness checks.

Examples

```
AIOLI_COMMAND_OVERRIDE="nim_llm --model_name {{.modelName}} --
model_path {{.modelDir}} --port {{.containerPort}} --health_port {{.healthPort}} --
num_gpus {{.numGpus}}"
```

```
AIOLI_COMMAND_OVERRIDE="openllm start {{.modelName}} --port
{{.containerPort}} --gpu-memory-utilization 0.9 --max-total-tokens 4096"
```

```
AIOLI_COMMAND_OVERRIDE="bentoml serve {{.modelDir}}/bentofile.yaml --
production --port {{.containerPort}} --host 0.0.0.0"
```

Object Model Reference Schema

Description and control of your inference service is accomplished via two primary objects: **Packaged Model** and **Deployment**. Additionally, when referencing a Package Model via an external hosting method such as S3 bucket, or huggingface.co registry, you may need to configure a **Registry** to enable that access.

Deployment

The *Deployment* object controls when a **Packaged Model** is deployed. The Deployment object has the following attributes:

Input Attributes

name

The name of the deployment to enable access to it via the REST interface or CLI. This is the name used in the associated KServe inference service that will be created.

namespace

The Kubernetes namespace into which the service is deployed. It must already exist.

model

The name (or `id`) of the packaged model to be deployed.

security

Encapsulates the security option (authenticationRequired) for the deployed service.

autoScaling

Controls the scaling limits minReplicas/maxReplicas, metric to control scaling, and the target value.

canaryTrafficPercent

The percentage of traffic to route to this particular model version. The default is `100`.

goalStatus

Specifies the intended status to be achieved by the deployment. The default is `Ready`.

- **`Ready`**: The inference service will be deployed to enable inference calls.
- **`Paused`**: The inference service will be stopped and no longer accept calls.

envirionment

Environment variables to be provided to the container image when started.

arguments

Arguments to be passed to the container image when started. These are in addition to any configured on the packaged model.

Managed Attributes

id

A unique identifier to identify this service.

status

Summary status of the deployed service.

- **`Deploying`**: Service configuration is in progress.
- **`Failed`**: The service configuration failed.
- **`Ready`**: The service has been successfully configured and is serving.
- **`Updating`**: A new service revision is being rolled out.
- **`UpdateFailed`**: The current service revision failed to roll out due to an error. The prior version is still serving requests.
- **`Deleting`**: The deployed service is being removed.

- **Paused** : The deployed service has been stopped by the user or an external action.
- **Unknown** : Unable to determine the status.
- **Canceled** : The deployed service has been canceled.

state

State details of the current service configuration requested. See the DeploymentStateDetails component for details.

secondaryState

State details of a prior service configuration until the currently requested configuration has been fully rolled out.

PackagedModel

The *Packaged Model* object identifies the model and code that make up your inference service. The code may be provided via a container image, or via an external hosting method (S3 or huggingface.co registry). The Packaged Model object has the following attributes:

Input Attributes

name

The name of the model.

description

A text description of the model.

modelFormat

Model format for downloaded models (e.g. from `S3`, `http`, etc.).

- **custom** : The packaged model is provided in a container image.
- **bento-archive** : The packaged model is a bento archive file (`.bento`). It will be downloaded, expanded, and then will be served using the bentolm serve command in a provided bentoml serving container.
- **openllm** : The packaged model will be served using the openllm serve command in a provided openllm serving container.
- **vllm** : The packaged model is served using the vllm/openai-vllm container image.
- **nim** : The packaged model will be served using the specified NVIDIA NIM microservices container image.

registry

The name or `id` of a registry object. If the model data is not provided via a container image, this must be specified.

url

Reference to the Bento or model to be served.

- **openllm** : An OpenLLM model name from **huggingface.co** dynamically loaded and executed with a VLLM backend.
- **hf** : a vLLM-compatible model dynamically downloaded from **huggingface.co**.
- **s3** : An OpenLLM model path which will be dynamically downloaded from an associated S3 registry bucket.
- **ngc** : An NVIDIA NGC model will be dynamically downloaded from the associated ngc registry bucket and executed with the specified NVIDIA NIM microservices container image.

resources

The resource requirements for running the service (requests/limits) for cpu/memory/gpu.

imae

The containerized bento where the inference service is to be deployed.

environment

Environment variables to be provided to the container image when started. See Deployment & Packaged Model Environment Variables for a list of default options.

arguments

Arguments to be passed to the container image when started.

Managed Attributes

id

A unique identifier to identify this particular model version.

verison

An automatically incrementing integer version of the model as you make changes.

Registry

The Registry object provides the metadata that describes *how* to download a PackagedModel for deployment.

Input Attributes

name

The name of service to enable access to it via the REST interface or CLI.

description

A text description of the service.

type

The type of this model registry.

- **s3** : Configuration to enable access to an S3 bucket.
- **openllm** : Configuration to enable direct download of openllm models from **huggingface.co**. Provide your access token in the secretKey field.

- **ngc** : Configuration to enable direct download from the NGC: AI Development Catalog.

endpointUrl

The registry endpoint (host).

- **s3** : The S3 registry endpoint for the associated S3 region. Required.
- **openllm** : The **huggingface.co**-compatible endpoint (default `https://huggingface.co`).
- **vllm** : The **huggingface.co**-compatible endpoint (default `https://huggingface.co`).
- **ngc** : The NVIDIA NGC-compatible api endpoint (default `https://api.ngc.nvidia.com`).

bucket

The bucket or organization name, depending on which of the following values is selected as the model registry type.

- **s3** : The required S3 bucket name.
- **ngc** : The required NGC org name.

accessKey

The access key, username or team name for the registry.

- **s3** : The access key/username.
- **ngc** : The optional NGC team name.

secretKey

The secret key is the password, secret key, or access token for the registry.

- **s3** : The `secretKey` provides a secret key for the S3 bucket.
- **openllm** : The `secretKey` is the access token for **huggingface.co** and is supplied to the launched container via the `HF_TOKEN` environment variable.
- **ngc** : The NVIDIA NGC `apikey`.

insecureHttps

For `https` endpoints, the server certificate will be accepted without validation.

Managed Attributes

id

A unique identifier to identify this service.

DeploymentStateDetails

The state details of an inference service Deployment are described with the following attributes:

Attributes

endpoint

The endpoint `uri` used to access the inference service.

nativeAppName

The name of the Kubernetes application for the specific service version. Use this name to match the app value in Grafana/Prometheus to obtain logs and metrics for this deployed service version.

status

The status of a particular inference service revision.

- **Deploying** : The service configuration is in progress.
- **Failed** : The service configuration failed.
- **Ready** : The service has been successfully configured and is serving.
- **Updating** : A new service revision is being rolled out.
- **UpdateFailed** : The current service revision failed to rollout due to an error. The prior version is still serving requests.
- **Deleting** : The service is being removed.
- **Paused** : The service has been stopped by the user or an external action.
- **Unknown** : Unable to determined the service's status.
- **Canceled** : The specified model version of the deployment was canceled by the user.

trafficPercentage

Percent of traffic being processed by this service/model version.

failureInfo

A list of any failures associated with the deployment of this service/model version.

modelId

The `id` of the deployed packaged model associated with this state.

Pre-loading Models

When deploying an inference service, it can take a significant amount of time for the service to reach the `Ready` state if it has to download and load a large AI model during startup. Pre-loading models in MLIS significantly reduces inference service startup times, especially for large AI models.

This section covers two approaches to pre-loading models: **Model Caching** and **Manual Pre-loading**.

Feature	Model Caching	Manual Pre-loading
Description	Automatic cache-on-first-use when enabled by an Administrator	Manual creation of a PersistentVolumeClaim (PVC) for each model version and referencing the PVC URL
Scope	Multiple deployments & namespaces	Individual deployments
Model Types	Bento-Archive, OpenLLM, and Nim -- Custom models not supported; ideal for models with existing URLs (`pfs://`, `openllm://`, `s3://`)	Any; ideal for OpenLLM or bento-archive models
Storage	Shared PVC across all deployments	PVC can be shared or dedicated to a specific model
User Experience	- Transparent - Automatic removal of unused models	• Must create and manage a PVC - Must use PVC syntax & URL to add packaged model • Must have PVC in the same namespace as the model being deployed • Must manually remove unused models
Flexibility	Allows defining caching behavior	Allows mapping of arbitrary storage

Both methods ensure models are readily available at service startup, improving MLIS deployment responsiveness and efficiency. Choose the method that best fits your specific use case and requirements.

NOTE

Note that there are other uses for PVCs, such as:

- **Cache-on-first-use PVC**: Using a PVC that automatically caches the model on first use by referencing the PVC URL; only usable with Custom and NIM models where the URL isn't already used.
- **Arbitrary PVC mounts**: Mounting a PVC to add arbitrary storage to your inference service container; the PVC can contain a model or any other data.

Subtopics

Model Caching (PV/PVC)
Model caching is a method for pre-loading models in MLIS to reduce inference service startup times and improve performance. This approach uses a ReadWriteMany (RWX) PersistentVolumeClaim (PVC) that can be accessed across multiple namespaces where you deploy your inference services. Model caching is enabled by default in HPE AI Essentials Software.

Manual Model Pre-loading (PVC)
Manual pre-loading is an advanced method for pre-loading models in MLIS using PersistentVolumeClaims (PVCs) to store models. It offers more control for experienced users with specific storage requirements, making it ideal for scenarios involving OpenLLM or bento-archive models. However, this approach requires a deep understanding of Kubernetes and adds complexity, particularly for tasks like canary deployments.

Model Caching (PV/PVC)

Model caching is a method for pre-loading models in MLIS to reduce inference service startup times and improve performance. This approach uses a ReadWriteMany (RWX) PersistentVolumeClaim (PVC) that can be accessed across multiple namespaces where you deploy your inference services. Model caching is enabled by default in HPE AI Essentials Software.

During installation, when model caching is enabled, the MLIS controller automatically sets up and manages this PVC. When you deploy an inference service, the system creates copies of the underlying **PersistentVolume** (PV) and PVC in the target namespace, ensuring efficient access to cached models.

Benefits

- Improved startup time performance
- Automatic management by the MLIS controller
- Efficient access to models across multiple namespaces

Key features

- Utilizes a ReadWriteMany (RWX) PersistentVolumeClaim (PVC)
- Creates efficient PV and PVC copies in target namespaces upon deployment
- Ensures frequently used models are readily available

Model caching is particularly useful for optimizing resource usage and enhancing overall system efficiency in large-scale deployments.

Model Caching Options

The full set of options for configuring model caching are provided in the default `values.yaml` and can be reviewed in the following section.

```
modelsCacheStorage:
            enabled: false
            checkUnusedCachedModelsEvery: 1d
            purgeUnusedCachedModelsAfter: 1w
            storageSize: 100Gi
```

Configure Caching Behavior

Adjust the caching behavior using the following parameters:

- `checkUnusedCachedModelsEvery`: How frequently the controller checks for and deletes unused cached models.
- `purgeUnusedCachedModelsAfter`: How long the controller retains unused cached models.

The time unit can be any of the following: `s` (seconds), `m` (minutes), `h` (hours), `d` (days), or `w` (weeks).

```
modelsCacheStorage:
        enabled: true
        checkUnusedCachedModelsEvery: 1d
        purgeUnusedCachedModelsAfter: 1w
```

Enable Caching for Specific Models

Individual models must explicitly enable the use of model caching. Users with an Admin role can use the CLI or UI to enable caching for individual models. Once enabled, new versions of the model will continue to have caching enabled even if the new version is created by a user with the Maintainer role

```
aioli model update <model-name> --enable-caching/--disable-caching
```

Disable Caching for Specific Deployments

Model caching can be temporarily disabled for a particular deployment by setting the environment variable `AIOLI_DISABLE_MODEL_CACHE=true` in the deployment. This may be useful if the disk used for the shared network storage becomes full, causing deployments to fail. Disabling the model caching for a particular deployment reverts back to using the local disk on the pod at the expense of having to download the model for each deployment replica.

Manage Cached Models

- Models are automatically removed when deleted from the database
- Unused models are purged based on the configured time period
- The MLIS controller manages the cache content

Disable PV/PVC Caching

1. To disable model caching, navigate to HPE MLIS > Configure and set the following:

   ```
   modelsCacheStorage:
       enabled: false
   ```

2. Models continue to use space on the PV/PVC until a canary rollout is performed. This rollout must include at least one update to your **deployment** or **model** settings to trigger the canary rollout.

   ```
   aioli deployment update <DEPLOYMENT_NAME> \
   --model <PACKAGED_MODEL_NAME> \
   --canary-percentage 100
   ```

Manual Model Pre-loading (PVC)

Manual pre-loading is an advanced method for pre-loading models in MLIS using PersistentVolumeClaims (PVCs) to store models. It offers more control for experienced users with specific storage requirements, making it ideal for scenarios involving OpenLLM or bento-archive models. However, this approach requires a deep understanding of Kubernetes and adds complexity, particularly for tasks like canary deployments.

> **NOTE**
>
> **Best Practice**
>
> Each model, including different versions of the same model, should be pre-loaded into its own separate directory. **Do not store different versions of a model in the same directory.** To avoid confusion, it's recommended to include the model version in the directory name.

Benefits

- Greater control over model storage and access
- Fine-grained control over model management
- Ideal for models not compatible with automated caching

Key features

- Uses PersistentVolumeClaims (PVCs) for model storage
- Supports custom storage mapping
- Provides manual control over model placement and lifecycle

Manual pre-loading is particularly useful for OpenLLM or bento-archive models and scenarios requiring specific storage configurations. However, it requires more in-depth knowledge of Kubernetes and may add complexity to operations like canary deployments.

When to Use Manual Pre-loading

For most users, the simpler Model Caching (PV/PVC) method is recommended. Consider manual pre-loading if you:

- Need to pre-load models incompatible with standard caching

- Have specific storage needs not supported by model caching
- Possess advanced Kubernetes knowledge and can manage PVCs manually
- Require precise control over model lifecycles

Before You Start

To use Manual Pre-loading with PVCs, administrators need to ensure the following:

- **Create a PVC:** You must create a PVC in the same Kubernetes namespace where you intend to deploy your inference service.
- **Sufficient Storage:** Ensure the PVC has storage resources for storing the entire model, based on expected model sizes.
- Review any necessary resources specific to the model you want to pre-load:
  - **NIM:** **NIM Tutorials**, **NIM Deploy on KServe**

Note on HuggingFace Tokens

If the model you wish to download from HuggingFace requires authentication (i.e., a token), you must set the `HF_TOKEN` environment variable with your HuggingFace token. The following HuggingFace example assumes that this token is stored in the environment variable `HF_TOKEN`.

Example: OpenLLM from HuggingFace

The following example demonstrates how to preload the `meta-llama/Meta-Llama-3.1-8B-Instruct` model from HuggingFace onto a PVC named `models-pvc`. **This model requires a HuggingFace token and access permissions**, which can be requested at `huggingface.co`. This example employs a Kubernetes Job that executes the `huggingface-cli` command to download the model to the `/mnt/models/meta-llama-3.1-8b-instruct` directory on the PVC. However, you can customize the Job to use any method that suits your needs for preloading models.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: download-llama-3-1-8b-instruct-model
spec:
  template:
    spec:
      containers:
      - name: model-installer
        image: kserve/huggingfaceserver:latest
        command: ["/bin/sh", "-c"]
        args:
          - |
            MODEL_DIR="/mnt/models/meta-llama-3.1-8b-instruct"

            huggingface-cli download meta-llama/Meta-Llama-3.1-8B-Instruct --token
$HF_TOKEN --local-dir $MODEL_DIR

            [ $? -eq 0 ] && echo "Model download complete" || echo "Model download
failed"
        env:
        - name: HF_TOKEN
          value: hf_XXXXXXXXXXXXXXXXXXXXX
        volumeMounts:
        - name: models-cache
          mountPath: /mnt/models
      restartPolicy: OnFailure
      volumes:
      - name: models-cache
        persistentVolumeClaim:
          claimName: models-pvc
```

Applying the Kubernetes Job

After creating the YAML file (e.g., `download-llama-3-1-8b-instruct-model.yaml`) with the content provided in the example above, you can apply it to your Kubernetes cluster using the following command:

```
kubectl apply -f download-llama-3-1-8b-instruct-model.yaml
```

TIP

You can monitor the logs to view the progress of the model being downloaded onto the PVC or to check for any errors that may have occurred.

```
kubectl logs job/download-llama-3-1-8b-instruct-model
```

When the model has been successfully downloaded, the last few lines of the log should look similar to the following output:

```
Download complete. Moving file to /mnt/models/meta-llama-3.1-8b-instruct/model-
00001-of-00004.safetensors
Fetching 17 files:  41%|████      | 7/17 [01:04<01:36,  9.67s/it]Download complete.
Moving file to /mnt/models/meta-llama-3.1-8b-instruct/original/consolidated.00.pth
Fetching 17 files: 100%|██████████| 17/17 [02:04<00:00,  7.30s/it]
/mnt/models/meta-llama-3.1-8b-instruct
Model download complete
```

At this point, you can now reference the model stored on the PVC in your inference service deployment.

Developer Environment Setup

The following steps guide you through setting up your MLIS developer environment to create and deploy containerized inference services. The HPE Machine Learning Inferencing Software (MLIS) developer environment is the software platform where you want to run MLIS CLI commands. This platform could be a notebook or, if you are doing remote development, a desktop shell. The MLIS developer environment is based on a Python (version 3.9 or greater) environment and includes commands from the CLI, BentoML, and OpenLLM.

NOTE

You only need to complete this setup if you intend to use the CLI.

How to Set Up Your Developer Environment

WARNING

You must install the dependencies separately and in the order listed below.

If you try to install them at the same time (`pip install bentoml==x openllm==0.4.44 aioli-sdk==x`), you may run into compatibility errors.

1. Create a virtual environment to allow HPE Machine Learning Inferencing Software (MLIS) to have its own Python dependencies and configuration, without interfering with other projects or the system-wide Python installation.

   ```
   python -m venv ~/.virtualenvs/mlis
   ```

2. Activate the virtual environment to make it the active Python environment for your current shell session.

   ```
   source ~/.virtualenvs/mlis/bin/activate
   ```

3. Install the latest version of pip.

   ```
   python -m pip install --upgrade pip
   ```

4. You can perform either of the folowing step as required:

   - If you are using OpenLLM, Install both BentomL and OpenLLM using following command:

     ```
     python -m pip install bentoml==1.1.11 openllm==0.4.44 openllm-client==0.4.44
     ```

     Or

   - If you are using only BentoML, you can install any compatible version of bentoml. See Customize the Base Container Image to decide on the compale BentoML versions.
     - For AIE 1.8, install BentoML version 1.3.5 using:

       ```
       Install BentoML. python -m pip install bentoml==1.3.5
       ```

     - For AIE 1.9, install BentoML version 1.4.3 using:

       ```
       Install BentoML. python -m pip install bentoml==1.4.3
       ```

5. If you have a supported GPU (Nvidia), install the OpenLLM VLLM backend. This includes the appropriate drivers needed to build an inference service container that supports GPUs.

   ```
   python -m pip install "openllm[vllm]"
   ```

6. Install the SDK.

   ```
   python -m pip install aioli-sdk=={{<release/majorMinorPatchNumber>}}
   ```

7. Install the AWS CLI and configure it with your AWS credentials. You may need to create an IAM user with the necessary permissions and pass in the access key and secret key through the AWS CLI.

   ```
   brew install awscli
   ```

8. Install any other python dependencies your inference service may need into the same python environment.

Using the MLIS CLI with HPE AI Essentials Software

1. To use the MLIS CLI (aioli command):

   ```
   export AIOLI_CONTROLLER=https://mlis.{AIE-DOMAIN}
   ```

2. Log in to your cluster:

   ```
   aioli auth login
   ```

> **NOTE**
> You can also use the `-c https://mlis.{AIE-DOMAIN}` option to point at the MLIS endpoint on your HPE AI Essentials cluster.

Next Steps

You are now ready to:

- Connect to an existing HPE Machine Learning Inferencing Software (MLIS) platform instance and deploy an inference service. For information on accessing MLIS, see MLIS.

Inference Service Deployment Quickstart

This guide walks you through the entire service deployment journey, from registry setup to requesting a prediction from your deployed service.

Before You Start

- Ensure you have completed the Developer Environment Setup to make the openLLM commands available.

1. Set Up a Registry

For this quickstart, we'll use Hugging Face as our OpenLLM-compatible registry. For other registry setups, see the Registry documentation.

1. Go to the **Hugging Face website** and sign in to your account.
2. Navigate to your Profile and select Settings.
3. Go to Access Tokens and select New Token.
4. Fill in the following details:
   - **Name:** Example: `my-hf-token`.
   - **Type:** Select `read`.
5. Select Generate a Token and copy it.

2. Add a Registry

Now, add the Hugging Face registry to the platform.

1. Click Tools & Frameworks on the left-navigation bar.
2. Navigate to the HPE MLIS tile and click Open.
3. Go to Registries and select Add new registry.
4. Provide the following details:
   - **Name:** e.g., `hf-registry`.
   - **Type:** Choose `OpenLLM` from the dropdown.
   - **Hugging Face Token:** Paste the token from the previous step.
5. Select Create registry.

3. Add a Packaged Model

Hugging Face provides a wide range of pre-trained models that you can use. For this quickstart, we'll use the `facebook/opt-125m` model.

1. Go to Packaged models and select Add new model.
2. Provide the following details:
   - **Name:** Example: `opt-125m-inference-model`.
   - **Description:** Example: `OPT-125M model for inference tasks`.
3. Select Next.
4. Registry: Select the registry you created previously.
5. In Pick a model from a list, choose the `facebook/opt-125m` model.
6. Select the default image by leaving the image (advanced) field blank and select Next.
7. From the Resource Template dropdown, choose `gpu-tiny` and select Next.

8. Skip the Environment Variables and Arguments and select Create model.

4. Create a Deployment

Deploy the model as an inference service on your Kubernetes cluster.

A deployment is the final step in the process. It's the actual instantiation of the inference service on your Kubernetes cluster. Once you've created a deployment, you can start sending requests to the service's endpoint.

1. Go to **Deployments** and select **Create new deployment**.
2. Provide a **Deployment Name**, for example: `opt-inference-deployment`, and select **Next**.
3. From the **Packaged Model** dropdown, choose the model you created earlier and select **Next**.
4. **Namespace** Choose `default`, or your selected namespace. If this option is not available, the Namespace is already set for you.
5. Select **Next** twice to skip to the **Scaling** tab.
6. From the **Auto scaling targets template** dropdown, choose `fixed-1` and select **Next**.
7. Provide any needed **Environment Variables** or **Arguments** and select **Done**.

Wait a few minutes for the deployment to reach a `Ready` status. Once it's ready, you can start sending requests to the service's endpoint.

5. Send a Request to the Service

1. Open the deployment you created and copy its **Endpoint URL**.
2. Create an auth token. See Endpoints.

   ```
   export OPENLLM_AUTH_TOKEN="$AUTH_TOKEN"
   ```

   Or
   Alternatively, Navigate to Gen AI > Model Endpoints > Generate API Token to create an auth token.
3. Send a request to the endpoint using the following `openllm` command:

   ```
   openllm query --timeout=600 --endpoint <DEPLOYMENT_ENDPOINT>  "What is
   aioli?"
   ```

   ```
   What is aioli?
   It's a kind of "coconut milk" that has been made with a yeast called aioli.  It's pretty
   sweet.  I have it in my fridge.
   ```

Proxy-Related Inference Failures

Symptom

On clusters that use a proxy, inferencing requests for the `nv-embedqa-e5-v5` or `nv-embedqa-mistral-7b-v2` models may fail with the following error:

```
{
"object": "error",
"message": "Triton service is currently unavailable and may be currently initializing.
Please check the health endpoints and try again in a few moments. If problem
persists then please check the Triton logs for more details.",
"detail": {},
"type": "service_unavailable"
}
```

Additionally, the deployment pod logs will show the following error:

```
2024-10-01T18:08:46Z ERROR: root - Got error from Triton: failed to connect to all
addresses; last error: UNKNOWN: ipv4:[INTERNAL_IP]:443: HTTP proxy returned
response code 502
2024-10-01T18:08:46Z ERROR: root - Triton service unavailable
2024-10-01T18:08:46Z INFO: uvicorn.access - 127.0.0.6:0 - "POST /v1/embeddings
HTTP/1.1" 503
127.0.0.6:0 - "POST /v1/embeddings HTTP/1.1" 503
```

Cause

The deployment pod is unable to connect to the Triton service due to proxy-related issues.

Resolution

1. Add the IP address `0.0.0.0` to your `no_proxy` environment variable.
2. Update your Helm values file (`values.yaml`) or use the `--set` flag during installation to include the proxy configuration. For more details on setting global environment variables, refer to the **Global Environment Variables** section in the Helm Chart Values documentation.

   ```
   global:
     env:
       - name: no_proxy
         value: "localhost,127.0.0.1,0.0.0.0"
   ```

NOTE

To edit the `values.yaml` file:

- You can easily access the configuration by visiting the configuration button on the HPE MLIS product card.

- Edit the `NO_PROXY` & `no_proxy` values to include `0.0.0.0.` .
- Then click **Configure** to apply the changes.

3. After applying the changes, redeploy the affected models.

Verification

After applying the fix, attempt to run inference requests again. If successful, you should no longer encounter the error.

Failed Deployment

Deploying an inference service can fail for various reasons. This guide helps you troubleshoot common issues that may cause a deployment to fail.

Before You Start

Use `kubectl` to describe the revision and the deployment to get more information about the failure.

```
kubectl describe revision revision.serving.knative/<SERVICE_NAME>
kubectl describe deployment <DEPLOYMENT_NAME>
```

WARNING

Ephemeral Storage Considerations

Default disk sizes on cloud providers may not be sufficient for large models, which often require significant ephemeral storage. This can result in the inference service failing to start serving—in some instances, without providing an error message about being out of disk space.

Ephemeral storage is normally provided by the boot disk of the compute nodes. You can inspect the amount of ephemeral storage on your nodes using the `kubectl describe node <node-name>` command.

Error: Insufficient nvidia.com/gpu

Deploying a service with GPU request and limits of 1 on a cluster without nodes that contain a GPU will fail to deploy and remain in the deploying state. Using kubectl to describe the revision will show the status with the following messages:

Revision Output Example

```
...
   Resources:
     Limits:
       Cpu:          1
       Memory:       1Gi
       nvidia.com/gpu:  1
     Requests:
       Cpu:          1
       Memory:       1Gi
       nvidia.com/gpu:  1
...
Status:
  Actual Replicas:  0
  Conditions:
    Last Transition Time:  2024-02-23T21:35:21Z
    Message:               Requests to the target are being buffered as resources are
provisioned.
    Reason:                Queued
    Severity:              Info
    Status:                Unknown
    Type:                  Active
    Last Transition Time:  2024-02-23T21:35:21Z
    Reason:                Deploying
    Status:                Unknown
    Type:                  ContainerHealthy
    Last Transition Time:  2024-02-23T21:35:21Z
    Message:               0/1 nodes are available: 1 Insufficient nvidia.com/gpu.
preemption: 0/1 nodes are available: 1 No preemption victims found for incoming
pod..
    Reason:                Unschedulable
    Status:                False
```

```
  Type:                 Ready
  Last Transition Time:  2024-02-23T21:35:21Z
  Message:              0/1 nodes are available: 1 Insufficient nvidia.com/gpu.
preemption: 0/1 nodes are available: 1 No preemption victims found for incoming
pod..
  Reason:               Unschedulable
  Status:               False
  Type:                 ResourcesAvailable
```

Error: Internal Error

KServe generates `InternalError` events to report unexpected behavior within the KServe infrastructure. These events typically show up as Kubernetes events (`kubectl get events`). MLIS surfaces these errors both in a deployment's **Timeline** tab and when you run the `aioli d events` command.

These are not MLIS issues and, in most cases, they get resolved automatically by KServe with no user action required.

In cases where these errors appear in the **Errors** tab of the deployment (as opposed to the **Timeline** tab), there may be a configuration problem that needs to be resolved.

The following examples were all informational and were automatically resolved by KServe:

```
k get events |grep InternalError
59m         Warning   InternalError           revision/nim-jjh-predictor-00001
failed to update deployment "nim-jjh-predictor-00001-deployment": Operation
cannot be fulfilled on deployments.apps "nim-jjh-predictor-00001-deployment": the
object has been modified; please apply your changes to the latest version and try
again
5m39s       Warning   InternalError           revision/nim-jjh-predictor-00001
Unable to fetch image "determinedai/aioli-logger": failed to resolve image to digest:
Get "https://index.docker.io/v2/": context deadline exceeded
7m23s       Warning   InternalError           route/nim-jjh-predictor                 failed
to remove route annotation to /, Kind= "nim-jjh-predictor":
configurations.serving.knative.dev "nim-jjh-predictor" not found
5m35s       Warning   InternalError           inferenceservice/nim-jjh                fails
to update InferenceService status: Operation cannot be fulfilled on
inferenceservices.serving.kserve.io "nim-jjh": the object has been modified; please
apply your changes to the latest version and try again
14m         Warning   InternalError           revision/nim-jjjh2-predictor-00001
Unable to fetch image "determinedai/aioli-runtimes:utils.1": failed to resolve image
to digest: Head "https://index.docker.io/v2/determinedai/aioli-
runtimes/manifests/utils.1": context deadline exceeded
14m         Warning   InternalError           revision/nim-jjjh2-predictor-00001
failed to update deployment "nim-jjjh2-predictor-00001-deployment": Operation
cannot be fulfilled on deployments.apps "nim-jjjh2-predictor-00001-deployment": the
object has been modified; please apply your changes to the latest version and try
again
```

Error: Model Load Failed

When rolling out changes to a deployment that previously encountered errors, KServe may surface failure messages from a prior version of the inference service. This can cause the deployment to show a `Failed` status while the new deployment is in progress.

To determine if this is the case:

1. Check the **Latest event** column to see if a new deployment is in progress.
2. Select the deployment and review the **Timeline** tab to check previous events.

These errors will clear on their own as the new deployment reaches a `Ready` status.

Error: RevisionMissing, RevisionFailed

If you are getting opaque or "unknown" error messages from the failureType `RevisionMissing` or `RevisionFailed`, it may be that you have insufficient memory requests set on your packaged model. Try submitting a new version of the model with higher memory requests to resolve the issue.

Example error message:

```
'Revision "my-service-predictor-00001" failed with message: Container
failed with: failed to start containerd task
"f445020dbaa45c92f65a28cf8daad2b3aadaa00bc8e61decee2c498d8e576886":
cannot start a stopped process: unknown.'
```

```
kubectl get pods
...
my-service-predictor-00001-deployment-c5b55bb9f-g6484   1/4     CrashLoopBackOff
8 (111s ago)   18m
```

No Streamed Responses

Learn how to troubleshoot responses that are not streaming as expected.

Scenario

Responses that are configured to be streamed to the caller are not streamed. Instead, they are sent after the entire response is generated.

Triage

There is a known KServe defect that prevents streamed responses when logging is enabled. The agent sidecar added for logging disrupts the streaming process, causing the caller to receive the entire response only after it has been fully created, rather than receiving a streamed response.

A **bug ticket** has been filed with the KServe team to address this issue.

Resolution

This issue is resolved in **KServe 0.14 or greater**.

Historical Workaround

If you cannot upgrade KServe, the following workaround can be used to disable the KServe request/response logging but allow streaming to work as expected:

Define `AIOLI_DISABLE_LOGGER=1` in your packaged model or deployment environment variables.

NOTE

Default value for `AIOLI_DISABLE_LOGGER` is `1`. To enable MLIS request/response logging, change the value as `0` or `false`.

Ray

Provides a brief overview of Ray in HPE AI Essentials 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 AI Essentials Software supports the following Ray libraries:

- **Ray Serve**. See
- **Ray Tune**. .

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 AI Essentials Software supports the following features and functionality:

Ray Serve with Very Large Language Models (VLLM)

In HPE AI Essentials Software, you can deploy VLLMs using Ray Serve.

Ray Cluster Reconciliation

HPE AI Essentials Software provides an automatic Ray cluster reconciliation feature using Helm hooks.

When you upgrade Ray in HPE AI Essentials 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.

To submit jobs using Ray, you can connect to Ray cluster. See Connecting to Ray Cluster.

Ray Dashboard

Ray dashboard in HPE AI Essentials 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.

- 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 and click Open.

To enable Metrics view in Ray dashboard, see Enabling Metrics in the Ray Dashboard.

Security

To configure Ray to use TLS authentication for client-server communication, see **TLS Authentication**.

To learn more about Ray, see **Ray documentation**.

Subtopics

Connecting to Ray Cluster
Describes how to connect to Ray clusters to submit jobs.
Using JobSubmissionClient to Submit Ray Jobs
Describes how to connect to Ray cluster and submit Ray jobs using `JobSubmissionClient`.
Enabling Metrics in the Ray Dashboard
Describes how to enable metrics in the Ray dashboard.
Resource Configuration and Management
Describes resource configuration and management for Ray.
GPU Support for Ray
Describes how to enable GPU, configure the GPU resources, and disable GPU for Ray.
Ray Best Practices
Lists the best practices for Ray.

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.

To submit jobs using Ray, you can connect to Ray cluster in two different ways:
Connecting to Ray in HPE AI Essentials Software

To connect to Ray in HPE AI Essentials Software, run:

```
ray.init(address="ray://kuberay-head-svc.kuberay:10001")
```

Connecting to Ray from outside of HPE AI Essentials Software

To connect to Ray cluster from outside of HPE AI Essentials 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.

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
   ```

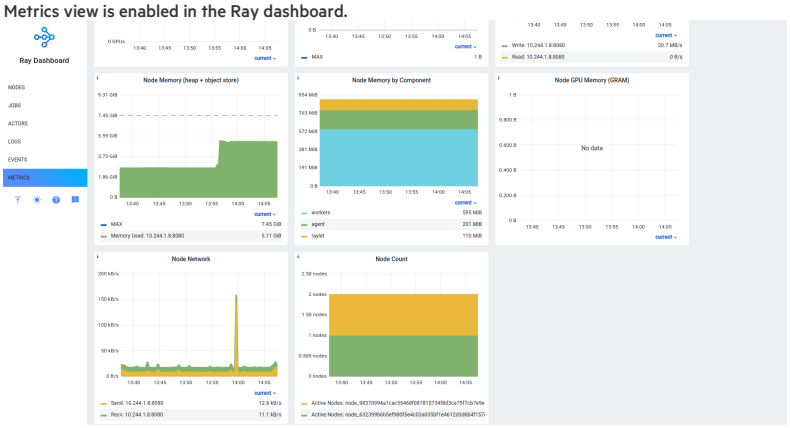
3. Forward Grafana's default port.

   ```
   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 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 AI Essentials 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.

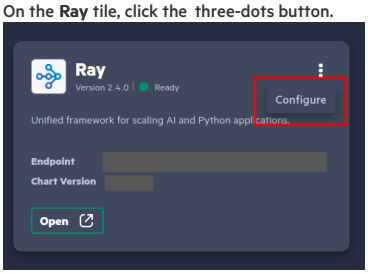
For example:

```
@ray.remote(scheduling_strategy="SPREAD")
def spread_func():
  return 2
```

To learn more see **Scheduling Strategies**.

Configuring Resources in the UI

1. Sign in to HPE AI Essentials Software as an Administrator.
2. Click the Tools & Frameworks icon on the left navigation bar.
3. On the **Ray** tile, click the three-dots button.



4. Select Configure to open the editor.
5. In the editor, modify the `resources` section to adjust resources.

GPU Support for Ray

Describes how to enable GPU, configure the GPU resources, and disable GPU for Ray.

Sign in as Administrator to HPE AI Essentials 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 AI Essentials Software installation.
- Enabling GPU support after HPE AI Essentials Software installation.

Enabling GPU Support During HPE AI Essentials Software Installation

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 AI Essentials Software Installation

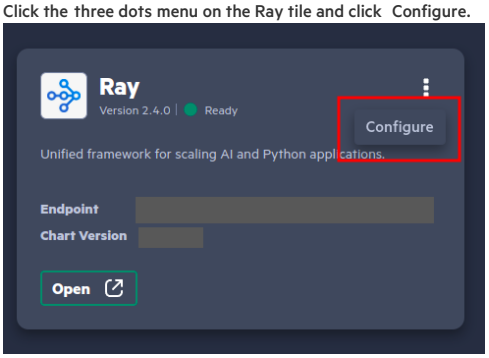
Before enabling the GPU support, when Ray is in an idle state, there are two pods running:

```
> 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 AI Essentials Software installation, follow these steps:

1. Click the Tools & Frameworks icon on the left navigation bar.
2. Click the three dots menu on the Ray tile and click Configure.



3. Set the value of `gpu.enabled` to `true`.

4. (Optional) Modify the available resources as required by updating the values within the `resources_gpu` section.

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:

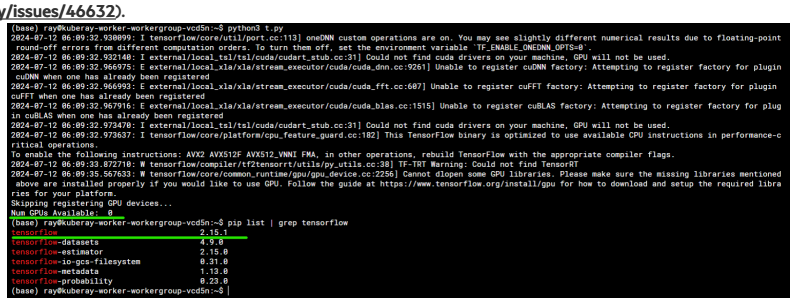
For this example to work, ensure you have installed GPU version of TensorFlow.

```
@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()
```

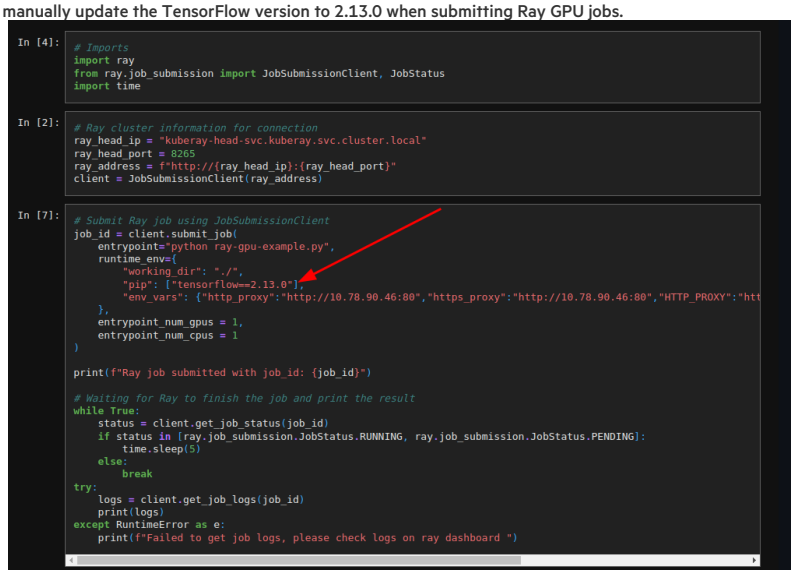
NOTE

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**).



To ensure that Ray jobs run on the GPU, manually update the TensorFlow version to 2.13.0 when submitting Ray GPU jobs.

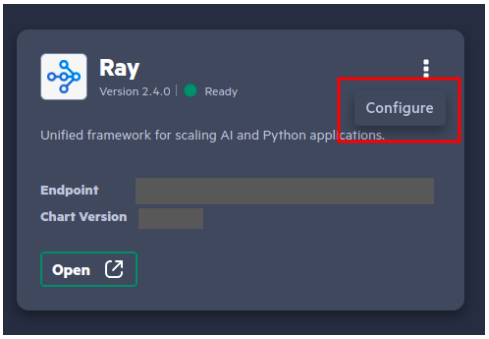


To learn more, see **Using Ray with GPUs**.

Disabling GPU Support for Ray

To disable GPU support for Ray after HPE AI Essentials Software installation, follow these steps:

1. Click the Tools & Frameworks icon on the left navigation bar.
2. Click the three dots menu on the Ray tile and click Configure.



3. Set the value of `gpu.enabled` to `false`.
4. Click Configure.

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 AI Essentials 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 AI Essentials Software.

Release Notes (v1.9.1)

This document summarizes the latest updates and enhancements in HPE AI Essentials Software (version 1.9.1), including new features, improvements, bug fixes, and known issues.

HPE AI Essentials Software delivers a ready-to-run set of AI and data-foundation tools with a unified control plane that provides adaptable solutions, ongoing enterprise support, and trusted AI services. HPE AI Essentials Software includes data and model compliance and extensible features that ensure AI pipelines are in compliance, explainable, and reproducible throughout the AI lifecycle.

HPE AI Essentials Software is the AI power within HPE Private Cloud AI that provides a turnkey solution designed for Generative AI. Leveraging advanced NVIDIA AI computing technology, HPE Private Cloud AI integrates all the necessary hardware, software, and services to accelerate AI productivity and streamline operations.

New Features

This release includes the following new features:
Dynamic NIM Delivery with NGC

HPE AI Essentials Software provides on-demand, run-time model delivery for direct access to NVIDIA's latest inference optimizations, at your convenience. All you need is your NVIDIA NGC key to securely fetch validated NIMs from NVIDIA. Model caching and NGC key integration were previously supported through HPE MLIS, and now this new delivery mechanism builds on those features for runtime fetching and local caching of validated NIMs. You can pre-load NIMs during provisioning or roll them out dynamically during deployment.
For details, see NVIDIA AI Enterprise.

Data Lakehouse Gateway

Data Lakehouse Gateway securely connects HPE AI Essentials Software to data lakehouses through a global namespace. The global namespace unifies data, regardless of where the data is located, making it easy to access through a variety of tools including SQL, Apache Spark, Superset, and AI/ML tools.
For details, see Data Lakehouse Gateway.

Projects

HPE AI Essentials Software includes support for enterprise multi-tenancy through Projects. A project is a collection of resources in an isolated environment that enables collaboration amongst project members. Now, you can work autonomously in your private project or collaboratively with your team members in shared projects. When you use an application, you simply select the project you want to work in through the Project selector. For example, when you open the Spark application, click the Project selector at the top of the page and select a project. The Spark application opens in the selected project and you perform your work there.
For details, see Projects.

Favorite Functionality

The new favorite functionality makes accessing your favorite applications easy. You can select your favorite applications by clicking the star icon in application tiles on the **Tools & Frameworks** page. Applications that you favor appear at the top of the page. Favorite applications also appear on the **Favorite Tools & Frameworks** tile on the **Home** page of HPE AI Essentials Software.

Enhancements

This release includes the following enhancements:
Tools & Frameworks Page in UI

Previously, the **Tools & Frameworks** page in the HPE AI Essentials Software UI organized tools and frameworks by category with tabs (Data Science, Data Engineering, and Analytics). Now, all tools and frameworks display on the page and you either use the **Search** field or the filter to locate an application. You can filter by status (Ready, Unknown, Error) and category (Data Science, Data Engineering, Analytics).

Deprecated Components

As of version 1.9, HPE AI Essentials Software no longer supports the following components:

- Feast
- WhyLogs

Resolved Issues

This release includes the following resolved issues:

Knowledge Base API examples with added models have the MODEL field in the API body

When a model is deployed through the Model Catalog's Add New Model feature via non-NGC MLIS registries (such as HuggingFace or OpenLLM), the Knowledge Base API examples now display the MODEL field in the API body.

No model failures and post-job action errors in Ray tune tasks

Ray tune tasks and model storage no longer result in model failures or errors.

Canary deployments no longer fail due to excessive endpoint name length

Endpoints no longer get stuck in the "Deploying" state after Canary Rollout due to the length of the subdomain of an inferencing endpoint.

Audit logs display for MLIS registries

The Audit Logs page lists the operations performed on MLIS registries, such as create, update, and delete.

Known Issues

This release includes the following known issues:

User no longer has access to knowledge bases after upgrade

After upgrading HPE AI Essentials Software, users that previously had full access to knowledge bases can no longer access the knowledge bases.
Resolution

Grant the users access to the knowledge bases, as described in Knowledge Base Access.

Running GPU workloads cause upgrade failures

The GPU operator cannot delete certain GPU workloads which causes upgrades to fail, resulting in the following error:

```
Warning  GPUDriverUpgrade 25m nvidia-gpu-operator
Failed to delete workload pods on the node for the driver upgrade,
[error when waiting for pod...]
```

Resolution

Before upgrading to HPE AI Essentials Software version 1.9, scale down all GPU workloads, perform the upgrade, and then scale the GPU workloads back up after the upgrade is completed.

1. Scale down GPU workloads:

   ```
   kubectl scale deployment <deployment-name> -n <namespace> --replicas=0
   ```

2. Upgrade to HPE AI Essentials Software version 1.9.

3. Scale workloads back up:

   ```
   kubectl scale deployment <deployment-name> -n <namespace> --replicas=0
   ```

NVIDIA driver pod goes into CrashLoopBackOff state after uploading GPU license

When you upload or delete a CPU license, the ezlicense service sets the nvidia.com/gpu.deploy.operands flag to false which causes the NVIDIA driver pod to terminate. If you upload a new GPU license while inference services are running, the NVIDIA driver pod comes up and goes into a CrashLoopBackOff state. The driver pod enters this state because it cannot unload its kernel modules, which are still in use by the inference services.
Resolution

Scale down the GPU workloads, allow the driver pod to come up, and then scale up the GPU workloads.

1. Scale down GPU workloads:

   ```
   kubectl scale deployment <deployment-name> -n <namespace> --replicas=0
   ```

2. Wait for the nvidia-driver-daemonset pod to come up successfully. To check pod status, run:

   ```
   kubectl get pods -n gpu-operator -l app=nvidia-driver-daemonset
   ```

3. Scale workloads back up:

   ```
   kubectl scale --replicas=<replicacount> deployment/my-app
   ```

Cannot delete table due to missing or inaccessible metadata

If Data Lakehouse Gateway cannot access the `pcai_<uuid>_metadata` directory or its contents at the source table location, the system returns a message stating that the table metadata is missing or inaccessible.
Resolution

See Troubleshooting Data Lakehouse Gateway.

Cannot update deployment due to limited resources

Updating an inference service or knowledge base deployment results in an exceeded quota error message:

```
FailedCreate
Error creating: pods... exceeded quota: project quotas,requested...
```

This message indicates that there are not enough resources to make the update. Updating a deployment requires 2x the number of resources. For example, if the deployment uses 2

GPU, there must be an additional 2 GPU to perform an update.

Resolution

For inference service models, you can pause the deployment, update the deployment, and then resume the deployment. However, you cannot pause knowledge base deployments. You must either provide the additional resources or delete the knowledge base and then recreate it. If you provide additional resources, you must provide 2x the amount used by the deployment.

Spark history server only displays logs for Spark applications created in 1.9

After upgrading to HPE AI Essentials Software 1.9, Spark history server does not show logs for Spark applications created in 1.8. Spark history server automatically deletes logs after 12 hours. Any logs from 1.8 would likely have been deleted during the upgrade period.

Spark applications created before upgrading to 1.9 no longer work when cloned

Due to a change in the pvc name (from $USERNAME-sparkhs-pvc to sparkhs-pvc), Spark applications that ran successfully in the previous version of HPE AI Essentials Software now get stuck in the "submitting" state when cloned, and the system returns the following message:

```
Warning  FailedScheduling  3m45s (x5 over 5m48s)
scheduler-plugins-scheduler  0/3 nodes are available:
persistentvolumeclaim "<user>-gma-5d9f1a8f-sparkhs-pvc" not found.
0/3 nodes are available: 3 Preemption is not helpful for scheduling.
```

Resolution

To change the pvc name in the Spark YAML configuration, complete the following steps:

1. Go to the **Spark Applications** page by clicking **Analytics > Spark Applications** in the left navigation panel.
2. In the **Project** field, select the project that contains the Spark application you need to modify.
3. In the **Actions** column, click the kebab menu (three dots) for the Spark application and select **Edit YAML**.
4. In the **volumes:** section of the YAML, delete the following persistentVolumeClaim:

   ```
   - name: sparkhs-eventlog-storage
     persistentVolumeClaim:
       claimName: $USERNAME-sparkhs-pvc
   ```

5. Click **Update Application**. The application runs automatically.
6. Clone the Spark application by selecting **Clone** in the **Actions** menu.
7. When the cloned application finishes running, click **View YAML** in the **Actions** menu. You can see that the system updated the persistentVolumeClaim:

   ```
   - name: sparkhs-eventlog-storage
     persistentVolumeClaim:
       claimName: sparkhs-pvc
   ```

8. (Optional) Delete the original version of the Spark application.

PrestoDB SQL Lab queries fail in Superset

When running queries from SQL Lab against a Presto database connection, Superset returns a 401 unauthorized error:

```
Presto Error
presto error: Unexpected status code 401
b'Unauthorized'
```

However, the same Presto connection works for creating datasets and charts. For details, see **https://github.com/apache/superset/issues/33554**.

After upgrade, a Knowledge Base with Read Only access in HPE AI Essentials Software shows as No Access

After upgrade, if a member was previously given Read Only access to a Knowledge Base, their access level now displays as No Access when it should display as Endpoint Access. However, the member can still access the Knowledge Base.

Resolution

After upgrading the Knowledge Base, the administrator must give Endpoint Access to the affected member user. See Knowledge Base Access.

After upgrade, a Knowledge Base with Full Access given to a member user in HPE AI Essentials Software shows as Full Access, but does not have the Edit option

If a member user is given Full Access to a Knowledge Base in the previous version of HPE AI Essentials Software, that member user still shows as having Full Access after upgrading, but cannot edit the Knowledge Base.
Resolution

The administrator must give Endpoint Access to the affected user, following the regular steps to provide Knowledge Base access. See Knowledge Base Access.

Auto scaling templates incorrectly accept special characters in the name

Although the system does not support or recognize auto scaling templates with special characters in the name, such as ( ), %, or *, the system does not prevent you from saving an auto scaling template with special characters. The auto scaling template will still appear in the **Auto scaling templates** list on the **Global Administration** page in MLIS.
Resolution

Do not use special characters in the name when you create an auto scaling template. You cannot edit or delete auto scaling templates with special characters.

RDMA does not support cross-switch routing

GPUDirect Remote Direct Memory Access (RDMA) in HPE AI Essentials Software does not currently support cross-switch routing. The NVIDIA nv-ipam does not support static routes between RDMA pod network interfaces, which is required for cross-switch routing.
Resolution

To use RDMA in an application, specify only one RDMA device in the pod spec.

UI displays the wrong vCPU count

An intermittent issue with the license CR can cause the UI to display the wrong vCPU count on the **Activation Key** tab of the **Settings** page.
Resolution

Complete the following steps to forcibly delete a license if the UI delete license or kubectl delete ezlicense operations fail:

1. Edit the ezlicense-validating-webhook-configuration validatingwebhookconfiguration:

   ```
   kubectl edit validatingwebhookconfigurations ezlicense-validating-webhook-
   configuration
   ```

2. Remove the `- DELETE` operation:

   ```
   apiVersion: admissionregistration.k8s.io/v1
   kind: ValidatingWebhookConfiguration
   ...
     rules:
     - apiGroups:
       - ezconfig.hpe.ezaf.com
       apiVersions:
       - v1alpha1
       operations:
       - CREATE
       - UPDATE
       - DELETE   <------- delete line
   ```

3. Delete all the GPU and vCPU licenses.

4. Add the vCPU licenses and then add the GPU licenses.

5. Edit the ezlicense-validating-webhook-configuration validatingwebhookconfiguration:

   ```
   kubectl edit validatingwebhookconfigurations ezlicense-validating-webhook-
   configuration
   ```

6. Add the `- DELETE` line to the rules.operations section:

   ```
   apiVersion: admissionregistration.k8s.io/v1
   kind: ValidatingWebhookConfiguration
   ...
     rules:
     - apiGroups:
       - ezconfig.hpe.ezaf.com
       apiVersions:
       - v1alpha1
       operations:
       - CREATE
       - UPDATE
       - DELETE    <--- add
   ```

CVE-2025-23266 NVIDIA Container Toolkit vulnerability

NVIDIA Container Toolkit for all platforms contains a vulnerability in some hooks used to initialize the container, where an attacker could execute arbitrary code with elevated permissions. A successful exploit of this vulnerability might lead to escalation of privileges, data tampering, information disclosure, and denial of service.
Mitigations for NVIDIA GPU Operator

Opt out of using the `enable-cuda-compat` hook when using the NVIDIA GPU Operator.

In the Toolkit Container, add `disable-cuda-compat-lib-hook` to the `NVIDIA_CONTAINER_TOOLKIT_OPT_IN_FEATURES` environment variable. To change this environment variable, include the following arguments when installing or upgrading the NVIDIA GPU Operator with Helm:

```
--set "toolkit.env[0].name=NVIDIA_CONTAINER_TOOLKIT_OPT_IN_FEATURES" \
--set "toolkit.env[0].value=disable-cuda-compat-lib-hook"
```

Notes:

- Any other required feature flags should be comma-separated in the `--set "toolkit.env[0].value"` flag.
- When using a GPU Operator version prior to 25.3.1, you can deploy NVIDIA Container Toolkit 1.17.8 by including the following argument when installing or upgrading the GPU Operator with Helm:

  ```
  --set "toolkit.version=v1.17.8-ubuntu20.04"
  ```

- For Red Hat Enterprise Linux or Red Hat OpenShift, you must specify the v1.17.8-ubi8 tag.

See **CVE-2025-23266** for more information.

CVE-2025-32424: MLIS vulnerability when loading models

The MLIS OpenLLM support runtime for CUDA relies on PyTorch to load models. However, due to a PyTorch vulnerability in OpenLLM 0.4.44, ensure that you deploy only models that you trust.
See **CVE-2025-32434 Detail** for more information.

CVE-2025-32375: MLIS vulnerability when creating a custom BentoML service layer

The MLIS OpenLLM support runtime relies on OpenLLM version 0.4.44 and BentoML version 1.1.11, which has the following vulnerability:
If you create a custom BentoML service layer that starts the BentoML start-runner-server, build it into a Bento, and then serve it via S3/PFS/PVC using the OpenLLM model format, the runner can be exploited in your inference service to enable a remote-code-execution attack. The attack surface is limited to users authorized to access the inference service, and can only impact the inference service container which runs as a non-privileged user in BentoML.

See **CVE-2025-32375 Detail** for more information.

Additional Resources

- **HPE AI Essentials User Guide**
- HPE AI Essentials Release Note Archives
  - **1.8.0 Release Notes**
  - **1.6.1 Release Notes**
  - **1.6.0 Release Notes**
  - **1.5.2 Release Notes**
- **HPE Private Cloud AI Administration Guide**
- **HPE Private Cloud AI Release Notes**