



Red Hat OpenShift AI Self-Managed 2.11

Working with data science pipelines

Work with data science pipelines from Red Hat OpenShift AI Self-Managed

Red Hat OpenShift AI Self-Managed 2.11 Working with data science pipelines

Work with data science pipelines from Red Hat OpenShift AI Self-Managed

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Enhance your data science projects on OpenShift AI by building portable machine learning (ML) workflows with data science pipelines.

Table of Contents

PREFACE	4
CHAPTER 1. ENABLING DATA SCIENCE PIPELINES 2.0	6
1.1. INSTALLING OPENSIFT AI WITH DATA SCIENCE PIPELINES 2.0	6
1.2. UPGRADING TO DATA SCIENCE PIPELINES 2.0	6
1.3. MIGRATING PIPELINES FROM DATA SCIENCE PIPELINES 1.0 TO 2.0	7
1.4. ACCESSING DATA SCIENCE PIPELINES 1.0 PIPELINES AND HISTORY	7
1.5. UNINSTALLING THE OPENSIFT PIPELINES OPERATOR	8
CHAPTER 2. MANAGING DATA SCIENCE PIPELINES	9
2.1. CONFIGURING A PIPELINE SERVER	9
2.2. DEFINING A PIPELINE	10
2.3. IMPORTING A DATA SCIENCE PIPELINE	11
2.4. DELETING A DATA SCIENCE PIPELINE	12
2.5. DELETING A PIPELINE SERVER	12
2.6. VIEWING THE DETAILS OF A PIPELINE SERVER	13
2.7. VIEWING EXISTING PIPELINES	14
2.8. OVERVIEW OF PIPELINE VERSIONS	14
2.9. UPLOADING A PIPELINE VERSION	14
2.10. DELETING A PIPELINE VERSION	15
2.11. VIEWING THE DETAILS OF A PIPELINE VERSION	16
2.12. DOWNLOADING A DATA SCIENCE PIPELINE VERSION	17
CHAPTER 3. MANAGING PIPELINE EXPERIMENTS	19
3.1. OVERVIEW OF PIPELINE EXPERIMENTS	19
3.2. CREATING A PIPELINE EXPERIMENT	19
3.3. ARCHIVING A PIPELINE EXPERIMENT	20
3.4. DELETING AN ARCHIVED PIPELINE EXPERIMENT	20
3.5. RESTORING AN ARCHIVED PIPELINE EXPERIMENT	21
3.6. VIEWING PIPELINE TASK EXECUTIONS	22
3.7. VIEWING PIPELINE ARTIFACTS	22
3.8. COMPARING RUNS	23
CHAPTER 4. MANAGING PIPELINE RUNS	25
4.1. OVERVIEW OF PIPELINE RUNS	25
4.2. STORING DATA WITH DATA SCIENCE PIPELINES	25
4.3. VIEWING ACTIVE PIPELINE RUNS	26
4.4. EXECUTING A PIPELINE RUN	27
4.5. STOPPING AN ACTIVE PIPELINE RUN	28
4.6. DUPLICATING AN ACTIVE PIPELINE RUN	28
4.7. VIEWING SCHEDULED PIPELINE RUNS	29
4.8. SCHEDULING A PIPELINE RUN USING A CRON JOB	30
4.9. SCHEDULING A PIPELINE RUN	31
4.10. DUPLICATING A SCHEDULED PIPELINE RUN	32
4.11. DELETING A SCHEDULED PIPELINE RUN	34
4.12. VIEWING THE DETAILS OF A PIPELINE RUN	34
4.13. VIEWING ARCHIVED PIPELINE RUNS	35
4.14. ARCHIVING A PIPELINE RUN	36
4.15. RESTORING AN ARCHIVED PIPELINE RUN	36
4.16. DELETING AN ARCHIVED PIPELINE RUN	37
4.17. DUPLICATING AN ARCHIVED PIPELINE RUN	38

CHAPTER 5. WORKING WITH PIPELINE LOGS	40
5.1. ABOUT PIPELINE LOGS	40
5.2. VIEWING PIPELINE STEP LOGS	40
5.3. DOWNLOADING PIPELINE STEP LOGS	41
CHAPTER 6. WORKING WITH PIPELINES IN JUPYTERLAB	43
6.1. OVERVIEW OF PIPELINES IN JUPYTERLAB	43
6.2. ACCESSING THE PIPELINE EDITOR	44
6.3. CREATING A RUNTIME CONFIGURATION	44
6.4. UPDATING A RUNTIME CONFIGURATION	47
6.5. DELETING A RUNTIME CONFIGURATION	49
6.6. DUPLICATING A RUNTIME CONFIGURATION	49
6.7. RUNNING A PIPELINE IN JUPYTERLAB	50
6.8. EXPORTING A PIPELINE IN JUPYTERLAB	51
CHAPTER 7. ADDITIONAL RESOURCES	53

PREFACE

As a data scientist, you can enhance your data science projects on OpenShift AI by building portable machine learning (ML) workflows with data science pipelines, using Docker containers. This enables you to standardize and automate machine learning workflows to enable you to develop and deploy your data science models.

For example, the steps in a machine learning workflow might include items such as data extraction, data processing, feature extraction, model training, model validation, and model serving. Automating these activities enables your organization to develop a continuous process of retraining and updating a model based on newly received data. This can help address challenges related to building an integrated machine learning deployment and continuously operating it in production.

You can also use the Elyra JupyterLab extension to create and run data science pipelines within JupyterLab. For more information, see [Working with pipelines in JupyterLab](#).

From OpenShift AI version 2.9, data science pipelines are based on [KubeFlow Pipelines \(KFP\) version 2.0](#). For more information, see [Enabling data science pipelines 2.0](#).

A data science pipeline in OpenShift AI consists of the following components:

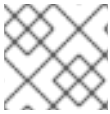
- Pipeline server: A server that is attached to your data science project and hosts your data science pipeline.
- Pipeline: A pipeline defines the configuration of your machine learning workflow and the relationship between each component in the workflow.
 - Pipeline code: A definition of your pipeline in a YAML file.
 - Pipeline graph: A graphical illustration of the steps executed in a pipeline run and the relationship between them.
- Pipeline experiment: A workspace where you can try different configurations of your pipelines. You can use experiments to organize your runs into logical groups.
 - Archived pipeline experiment: An archived pipeline experiment.
 - Pipeline artifact: An output artifact produced by a pipeline component.
 - Pipeline execution: The execution of a task in a pipeline.
- Pipeline run: An execution of your pipeline.
 - Active run: A pipeline run that is executing, or stopped.
 - Scheduled run: A pipeline run that is scheduled to execute at least once.
 - Archived run: An archived pipeline run.

This feature is based on KubeFlow Pipelines 2.0. Use the latest KubeFlow Pipelines 2.0 SDK to build your data science pipeline in Python code. After you have built your pipeline, use the SDK to compile it into an Intermediate Representation (IR) YAML file. The OpenShift AI user interface enables you to track and manage pipelines and pipeline runs. You can manage incremental changes to pipelines in OpenShift AI by using versioning. This allows you to develop and deploy pipelines iteratively, preserving a record of your changes.

You can store your pipeline artifacts in an S3-compatible object storage bucket so that you do not consume local storage. To do this, you must first configure write access to your S3 bucket on your storage account.

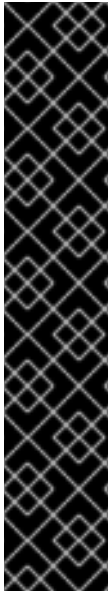
CHAPTER 1. ENABLING DATA SCIENCE PIPELINES 2.0

From OpenShift AI version 2.9, data science pipelines are based on [KubeFlow Pipelines \(KFP\) version 2.0](#). Data science pipelines 2.0 is enabled and deployed by default in OpenShift AI.



NOTE

The **PipelineConf** class is deprecated, and there is no KFP 2.0 equivalent.



IMPORTANT

Data science pipelines 2.0 contains an installation of Argo Workflows. OpenShift AI does not support direct customer usage of this installation of Argo Workflows.

To install or upgrade to OpenShift AI 2.9 or later with data science pipelines, ensure that your cluster does not have an existing installation of Argo Workflows that is not installed by OpenShift AI.

Argo Workflows resources that are created by OpenShift AI have the following labels in the OpenShift Console under **Administration > CustomResourceDefinitions** in the **argoproj.io** group:

```
labels:
  app.kubernetes.io/part-of: data-science-pipelines-operator
  app.opendatahub.io/data-science-pipelines-operator: 'true'
```

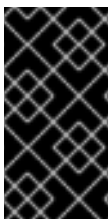
1.1. INSTALLING OPENSIFT AI WITH DATA SCIENCE PIPELINES 2.0

To install OpenShift AI 2.9 or later with data science pipelines, ensure that there is no installation of Argo Workflows that is not installed by data science pipelines on your cluster, and follow the installation steps described in [Installing and uninstalling OpenShift AI Self-Managed](#), or for disconnected environments, see [Installing and uninstalling Red Hat OpenShift AI in a disconnected environment](#).

If there is an existing installation of Argo Workflows that is not installed by data science pipelines on your cluster, data science pipelines will be disabled after you install OpenShift AI 2.9 or later.

To enable data science pipelines, remove the separate installation of Argo Workflows from your cluster. Data science pipelines will be enabled automatically.

1.2. UPGRADING TO DATA SCIENCE PIPELINES 2.0



IMPORTANT

After you upgrade to OpenShift AI 2.9 or later, pipelines created with data science pipelines 1.0 continue to run, but are inaccessible from the OpenShift AI dashboard. If you are a current data science pipelines user, do not upgrade to OpenShift AI with data science pipelines 2.0 until you are ready to migrate to the new pipelines solution.

To upgrade to data science pipelines 2.0, follow these steps:

1. Ensure that your cluster does not have an existing installation of Argo Workflows that is not installed by OpenShift AI, and then follow the upgrade steps described in [Upgrading OpenShift AI Self-Managed](#), or for disconnected environments, [Upgrading Red Hat OpenShift AI in a](#)

[disconnected environment](#).

If you upgrade to OpenShift AI 2.9 or later with data science pipelines enabled, and there is an existing installation of Argo Workflows that is not installed by data science pipelines on your cluster, OpenShift AI components will not be upgraded. To complete the component upgrade, disable data science pipelines or remove the separate installation of Argo Workflows from your cluster. The component upgrade will then complete automatically.

2. Update your workbenches to use the notebook image version 2024.1 or later. For more information, see [Updating a project workbench](#).
3. Manually migrate your pipelines from data science pipelines 1.0 to 2.0. For more information, see [Migrating pipelines from data science pipelines 1.0 to 2.0](#).

1.3. MIGRATING PIPELINES FROM DATA SCIENCE PIPELINES 1.0 TO 2.0

OpenShift AI does not automatically migrate existing data science pipelines 1.0 instances to 2.0. To use existing pipelines with data science pipelines 2.0, you must manually migrate them.

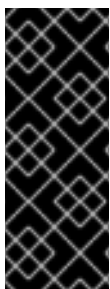
1. On OpenShift AI 2.9 or later, create a new data science project.
2. Configure a new pipeline server.
3. Update and recompile your data science pipelines 1.0 pipelines as described in [Migrate from KFP SDK v1: v1 to v2 migration instructions and breaking changes](#).



NOTE

Data science pipelines 2.0 does not use the **kfp-tekton** library. In most cases, you can replace usage of **kfp-tekton** with the **kfp** library.

4. Import your updated pipelines to your new data science pipelines 2.0-based data science project.
5. (Optional) Remove your data science pipelines 1.0 pipeline server.



IMPORTANT

Data science pipelines 1.0 used the **kfp-tekton** Python library. Data science pipelines 2.0 does not use **kfp-tekton**. You can uninstall **kfp-tekton** when there are no remaining data science pipelines 1.0 pipeline servers in use on your cluster.

For Data science pipelines 2.0, use the latest version of the KFP SDK. For more information, see the [Kubeflow Pipelines SDK API Reference](#).

1.4. ACCESSING DATA SCIENCE PIPELINES 1.0 PIPELINES AND HISTORY

You can view historical data science pipelines 1.0 pipeline run information in the OpenShift Console under **Pipelines > Project > PipelineRuns**

You can still connect to the KFP API server by using the **kfp-tekton** SDK for programmatic access to your pipelines and pipeline run history. For more information, see [Kubeflow Pipelines SDK for Tekton](#).

1.5. UNINSTALLING THE OPENSIFT PIPELINES OPERATOR

When your migration to data science pipelines 2.0 is complete, and if you are not using OpenShift Pipelines for any purpose other than data science pipelines 1.0, you can remove the OpenShift Pipelines Operator.



IMPORTANT

Before removing the OpenShift Pipelines Operator, ensure that migration of your data science pipelines 1.0 pipelines to 2.0 is complete, and that there are no remaining data science pipelines 1.0 pipeline servers in use on your cluster.

Additional resources

- [PyPI: kfp](#)
- [Kubeflow Pipelines SDK API Reference](#) .
- [Creating a data science project](#)
- [Configuring a pipeline server](#)
- [Importing a data science pipeline](#)
- [Deleting a pipeline server](#)

CHAPTER 2. MANAGING DATA SCIENCE PIPELINES

2.1. CONFIGURING A PIPELINE SERVER

Before you can successfully create a pipeline in OpenShift AI, you must configure a pipeline server. This task includes configuring where your pipeline artifacts and data are stored.



NOTE

You are not required to specify any storage directories when configuring a data connection for your pipeline server. When you import a pipeline, the **/pipelines** folder is created in the **root** folder of the bucket, containing a YAML file for the pipeline. If you upload a new version of the same pipeline, a new YAML file with a different ID is added to the **/pipelines** folder.

When you run a pipeline, the artifacts are stored in the **/pipeline-name** folder in the **root** folder of the bucket.



IMPORTANT

If you use an external MySQL database and upgrade to OpenShift AI 2.9 or later, the database is migrated to data science pipelines 2.0 format, making it incompatible with earlier versions of OpenShift AI.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have created a data science project that you can add a pipeline server to.
- You have an existing S3-compatible object storage bucket and you have configured write access to your S3 bucket on your storage account.
- If you are configuring a pipeline server with an external MySQL database, your database must use at least MySQL version 5.x. However, Red Hat recommends that you use MySQL version 8.x.
- If you are configuring a pipeline server with a MariaDB database, your database must use MariaDB version 10.3 or later. However, Red Hat recommends that you use at least MariaDB version 10.5.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the name of the project that you want to configure a pipeline server for.
A project details page opens.
3. Click the **Pipelines** tab.

4. Click **Configure pipeline server**.
The **Configure pipeline server** dialog appears.
5. In the **Object storage connection** section, provide values for the mandatory fields:
 - a. In the **Access key** field, enter the access key ID for the S3-compatible object storage provider.
 - b. In the **Secret key** field, enter the secret access key for the S3-compatible object storage account that you specified.
 - c. In the **Endpoint** field, enter the endpoint of your S3-compatible object storage bucket.
 - d. In the **Region** field, enter the default region of your S3-compatible object storage account.
 - e. In the **Bucket** field, enter the name of your S3-compatible object storage bucket.




IMPORTANT

If you specify incorrect data connection settings, you cannot update these settings on the same pipeline server. Therefore, you must delete the pipeline server and configure another one.

6. In the **Database** section, click **Show advanced database options** to specify the database to store your pipeline data and select one of the following sets of actions:
 - Select **Use default database stored on your cluster** to deploy a MariaDB database in your project.
 - Select **Connect to external MySQL database** to add a new connection to an external database that your pipeline server can access.
 - i. In the **Host** field, enter the database's host name.
 - ii. In the **Port** field, enter the database's port.
 - iii. In the **Username** field, enter the default user name that is connected to the database.
 - iv. In the **Password** field, enter the password for the default user account.
 - v. In the **Database** field, enter the database name.
7. Click **Configure pipeline server**.

Verification

In the **Pipelines** tab for the project:

- The **Import pipeline** button is available.
- When you click the action menu () and then click **View pipeline server configuration**, the pipeline server details are displayed.

2.2. DEFINING A PIPELINE

The KubeFlow Pipelines SDK enables you to define end-to-end machine learning and data pipelines. Use the latest KubeFlow Pipelines 2.0 SDK to build your data science pipeline in Python code. After you have

built your pipeline, use the SDK to compile it into an Intermediate Representation (IR) YAML file. After defining the pipeline, you can import the YAML file to the OpenShift AI dashboard to enable you to configure its execution settings.

You can also use the Elyra JupyterLab extension to create and run data science pipelines within JupyterLab. For more information about creating pipelines in JupyterLab, see [Working with pipelines in JupyterLab](#). For more information about the Elyra JupyterLab extension, see [Elyra Documentation](#).

Additional resources

- [Kubeflow Pipelines 2.0 Documentation](#)
- [Elyra Documentation](#)

2.3. IMPORTING A DATA SCIENCE PIPELINE

To help you begin working with data science pipelines in OpenShift AI, you can import a YAML file containing your pipeline's code to an active pipeline server, or you can import the YAML file from a URL. This file contains a Kubeflow pipeline compiled by using the Kubeflow compiler. After you have imported the pipeline to a pipeline server, you can execute the pipeline by creating a pipeline run.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have compiled your pipeline with the Kubeflow compiler and you have access to the resulting YAML file.
- If you are uploading your pipeline from a URL, the URL is publicly accessible.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
2. On the **Pipelines** page, select the **project** that you want to import a pipeline to.
3. Click **Import pipeline**.
4. In the **Import pipeline** dialog, enter the details for the pipeline that you are importing.
 - a. In the **Pipeline name** field, enter a name for the pipeline that you are importing.
 - b. In the **Pipeline description** field, enter a description for the pipeline that you are importing.
 - c. Select where you want to import your pipeline from by performing one of the following actions:
 - Select **Upload a file** to upload your pipeline from your local machine's file system. Import your pipeline by clicking **upload** or by dragging and dropping a file.

- Select **Import by url** to upload your pipeline from a URL and then enter the URL into the text box.
- d. Click **Import pipeline**.

Verification

- The pipeline that you imported appears on the **Pipelines** page and on the **Pipelines** tab on the project details page.


2.4. DELETING A DATA SCIENCE PIPELINE

If you no longer require access to your data science pipeline on the dashboard, you can delete it so that it does not appear on the **Data Science Pipelines** page.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- There are active pipelines available on the **Pipelines** page.
- The pipeline that you want to delete does not contain any pipeline versions.
- The pipeline that you want to delete does not contain any pipeline versions. For more information, see [Deleting a pipeline version](#).

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
2. On the **Pipelines** page, select the project that contains the pipeline that you want to delete from the **Project** list.
3. Click the action menu () beside the pipeline that you want to delete and click **Delete pipeline**.
4. In the **Delete pipeline** dialog, enter the pipeline name in the text field to confirm that you intend to delete it.
5. Click **Delete pipeline**.

Verification

- The data science pipeline that you deleted no longer appears on the **Pipelines** page.

2.5. DELETING A PIPELINE SERVER

After you have finished running your data science pipelines, you can delete the pipeline server. Deleting a pipeline server automatically deletes all of its associated pipelines, pipeline versions, and runs. If your pipeline data is stored in a database, the database is also deleted along with its meta-data. In addition, after deleting a pipeline server, you cannot create new pipelines or pipeline runs until you create another pipeline server.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
2. On the **Pipelines** page, select the **project** for the pipeline server that you want to delete.
3. From the **Pipeline server actions** list, select **Delete pipeline server**.
4. In the **Delete pipeline server** dialog, enter the pipeline server's name in the text field to confirm that you intend to delete it.
5. Click **Delete**.

Verification

- Pipelines previously assigned to the deleted pipeline server no longer appears on the **Pipelines** page for the relevant data science project.
- Pipeline runs previously assigned to the deleted pipeline server no longer appears on the **Runs** page for the relevant data science project.

2.6. VIEWING THE DETAILS OF A PIPELINE SERVER

You can view the details of pipeline servers configured in OpenShift AI, such as the pipeline's data connection details and where its data is stored.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- You have previously created a data science project that contains an active and available pipeline server.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
2. On the **Pipelines** page opens, select the **project** whose pipeline server you want to view.
3. From the **Pipeline server actions** list, select **View pipeline server configuration**.

Verification

- You can view the relevant pipeline server details in the **View pipeline server** dialog.


2.7. VIEWING EXISTING PIPELINES

You can view the details of pipelines that you have imported to Red Hat OpenShift AI, such as the pipeline's last run, when it was created, the pipeline's executed runs, and details of any associated pipeline versions.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.
- Existing pipelines are available.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
2. On the **Pipelines** page, select the relevant **project** for the pipelines you want to view.
3. Study the pipelines on the list.
4. Optional: Click **Expand** () on the relevant row to view details of any pipeline versions associated with the pipeline.

Verification

- A list of previously created data science pipelines appears on the **Pipelines** page.

2.8. OVERVIEW OF PIPELINE VERSIONS

You can manage incremental changes to pipelines in OpenShift AI by using versioning. This allows you to develop and deploy pipelines iteratively, preserving a record of your changes. You can track and manage your changes on the OpenShift AI dashboard, allowing you to schedule and execute runs against all available versions of your pipeline.

2.9. UPLOADING A PIPELINE VERSION

You can upload a YAML file to an active pipeline server that contains the latest version of your pipeline, or you can upload the YAML file from a URL. The YAML file must consist of a Kubeflow pipeline compiled by using the Kubeflow compiler. After you upload a pipeline version to a pipeline server, you can execute it by creating a pipeline run.

Prerequisites


- You have logged in to Red Hat OpenShift AI.

- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have a pipeline version available and ready to upload.
- If you are uploading your pipeline version from a URL, the URL is publicly accessible.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
2. On the **Pipelines** page, select the **project** that you want to upload a pipeline version to.
3. Click the **Import pipeline** dropdown list and select **Upload new version**.
4. In the **Upload new version** dialog, enter the details for the pipeline version that you are uploading.
 - a. From the **Pipeline** list, select the pipeline that you want to upload your pipeline version to.
 - b. In the **Pipeline version name** field, confirm the name for the pipeline version, and change it if necessary.
 - c. In the **Pipeline version description** field, enter a description for the pipeline version.
 - d. Select where you want to upload your pipeline version from by performing one of the following actions:
 - Select **Upload a file** to upload your pipeline version from your local machine's file system. Import your pipeline version by clicking **upload** or by dragging and dropping a file.
 - Select **Import by url** to upload your pipeline version from a URL and then enter the URL into the text box.
 - e. Click **Upload**.

Verification

- The pipeline version that you uploaded is displayed on the **Pipelines** page. Click **Expand** () on the row containing the pipeline to view its versions.
- The **Version** column on the row containing the pipeline version that you uploaded on the **Pipelines** page increments by one.




2.10. DELETING A PIPELINE VERSION

You can delete specific versions of a pipeline when you no longer require them. Deleting a default pipeline version automatically changes the default pipeline version to the next most recent version. If no pipeline versions exist, the pipeline persists without a default version.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines** → **Pipelines**.
The **Pipelines** page opens.
2. Delete the pipeline versions that you no longer require:
 - To delete a single pipeline version:
 - a. From the **Project** list, select the project that contains a version of a pipeline that you want to delete.
 - b. On the row containing the pipeline, click **Expand** ().
 - c. Click the action menu () beside the project version that you want to delete and click **Delete pipeline version**.
The **Delete pipeline version** dialog opens.
 - d. Enter the name of the pipeline version in the text field to confirm that you intend to delete it.
 - e. Click **Delete**.
 - To delete multiple pipeline versions:
 - a. On the row containing each pipeline version that you want to delete, select the checkbox.
 - b. Click the action menu () next to the **Import pipeline** dropdown, and select **Delete** from the list.

Verification

- The pipeline version that you deleted no longer appears on the **Pipelines** page or on the **Pipelines** tab for the data science project.

2.11. VIEWING THE DETAILS OF A PIPELINE VERSION


You can view the details of a pipeline version that you have uploaded to Red Hat OpenShift AI, such as its graph and YAML code.

Prerequisites

- You have logged in to Red Hat OpenShift AI.

- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.
- You have a pipeline available on an active and available pipeline server.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
The **Pipelines** page opens.
2. From the **Project** list, select the project containing the pipeline versions that you want to view details for.
3. Click **Expand** () on the row containing the pipeline that you want to view versions for.
4. Click the pipeline version that you want to view the details of.
The **Pipeline details** page opens, displaying the **Graph** and **YAML** tabs.

Verification

- On the **Pipeline details** page, you can view the pipeline graph and YAML code.


2.12. DOWNLOADING A DATA SCIENCE PIPELINE VERSION


To make further changes to a data science pipeline version that you previously uploaded to OpenShift AI, you can download pipeline version code from the user interface.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have created and imported a pipeline to an active pipeline server that is available to download.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Pipelines**.
2. On the **Pipelines** page, select the **project** that contains the version that you want to download.
3. For a pipeline that contains the version that you want to download, click **Expand** ().
4. Click the pipeline version that you want to download.
5. On the **Pipeline details** page, click the **YAML** tab.

6. Click the **Download** button () to download the YAML file containing your pipeline version code to your local machine.

Verification

- The pipeline version code downloads to your browser's default directory for downloaded files.

CHAPTER 3. MANAGING PIPELINE EXPERIMENTS

3.1. OVERVIEW OF PIPELINE EXPERIMENTS

A pipeline experiment is a workspace where you can try different configurations of your pipelines. You can use experiments to organize your runs into logical groups. As a data scientist, you can use OpenShift AI to define, manage, and track pipeline experiments. You can view a record of previously created and archived experiments from the **Experiments** page in the OpenShift AI user interface. Pipeline experiments contain pipeline runs, including recurring runs. This allows you to try different configurations of your pipelines.

When you work with data science pipelines, it is important to monitor and record your pipeline experiments to track the performance of your data science pipelines. You can compare the results of up to 10 pipeline runs at one time, and view available parameter, scalar metric, confusion matrix, and receiver operating characteristic (ROC) curve data for all selected runs.

You can view artifacts for an executed pipeline run from the OpenShift AI dashboard. Pipeline artifacts can help you to evaluate the performance of your pipeline runs and make it easier to understand your pipeline components. Pipeline artifacts can range from plain text data to detailed, interactive data visualizations.

3.2. CREATING A PIPELINE EXPERIMENT

Pipeline experiments are workspaces where you can try different configurations of your pipelines. You can also use experiments to organize your pipeline runs into logical groups. Pipeline experiments contain pipeline runs, including recurring runs.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project to create the pipeline experiment in.
3. Click **Create experiment**
4. In the **Create experiment** dialog, configure the pipeline experiment:
 - a. In the **Experiment name** field, enter a name for the pipeline experiment.
 - b. In the **Description** field, enter a description for the pipeline experiment.
 - c. Click **Create experiment**

Verification

- The pipeline experiment that you created appears on the **Experiments** tab.

3.3. ARCHIVING A PIPELINE EXPERIMENT

You can retain records of your pipeline experiments by archiving them. If required, you can restore pipeline experiments from your archive to reuse, or delete pipeline experiments that are no longer required.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- You have imported a pipeline to an active pipeline server.
- A pipeline experiment is available to archive.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment that you want to archive.
3. Click the action menu (**⋮**) beside the pipeline experiment that you want to archive, and then click **Archive**.
4. In the **Archiving experiment** dialog, enter the pipeline experiment name in the text field to confirm that you intend to archive it.
5. Click **Archive**.

Verification

- The archived pipeline experiment does not appear in the **Runs** tab, and instead appears in the **Archive** tab on the **Experiments** page for the pipeline experiment.

3.4. DELETING AN ARCHIVED PIPELINE EXPERIMENT


You can delete pipeline experiments from the OpenShift AI experiment archive.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.

- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.
- A pipeline experiment is available in the pipeline archive.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the archived pipeline experiment that you want to delete.
3. Click the **Archive** tab.
4. Click the action menu () beside the pipeline experiment that you want to delete, and then click **Delete**.
5. In the **Delete experiment** dialog, enter the pipeline experiment name in the text field to confirm that you intend to delete it.
6. Click **Delete**.

Verification

- The pipeline experiment that you deleted no longer appears on the **Archive** tab on the **Experiments** page.


3.5. RESTORING AN ARCHIVED PIPELINE EXPERIMENT

You can restore an archived pipeline experiment to the active state.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- An archived pipeline experiment exists in your project.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the archived pipeline experiment that you want to restore.
3. Click the **Archive** tab.
4. Click the action menu () beside the pipeline experiment that you want to restore, and then click **Restore**.

5. In the **Restore experiment** dialog, click **Restore**.

Verification

- The restored pipeline experiment appears in the **Experiments** tab on the **Experiments** page.

3.6. VIEWING PIPELINE TASK EXECUTIONS

When a pipeline run executes, you can view details of executed tasks in each step in a pipeline run from the OpenShift AI dashboard. A step forms part of a task in a pipeline.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously triggered a pipeline run.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Executions**.
2. On the **Executions** page, from the **Project** drop-down list, select the project that contains the experiment for the pipeline task executions that you want to view.

Verification

- On the **Executions** page, you can view the execution details of each pipeline task execution, such as its name, status, unique ID, and execution type. The execution status indicates whether the pipeline task has successfully executed. For further information about the details of the task execution, click the execution name.

3.7. VIEWING PIPELINE ARTIFACTS

After a pipeline run executes, you can view its pipeline artifacts from the OpenShift AI dashboard. Pipeline artifacts can help you to evaluate the performance of your pipeline runs and make it easier to understand your pipeline components. Pipeline artifacts can range from plain text data to detailed, interactive data visualizations.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.

- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously triggered a pipeline run.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Artifacts**.
2. On the **Artifacts** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the pipeline artifacts that you want to view.

Verification

- On the **Artifacts** page, you can view the details of each pipeline artifact, such as its name, unique ID, type, and URI.

3.8. COMPARING RUNS

You can compare up to 10 pipeline runs at one time, and view available parameter, scalar metric, confusion matrix, and receiver operating characteristic (ROC) curve data for all selected runs.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have created at least 2 pipeline runs.

Procedure

1. In the OpenShift AI dashboard, select **Experiments** > **Experiments and runs**
The **Experiments** page opens.
2. From the **Project** drop-down list, select the project that contains the runs that you want to compare.
3. In the **Experiments** column, click the experiment that you want to compare runs for. To select runs that are not in an experiment, click **Default**. All runs that are created without specifying an experiment will appear in the **Default** group.
The **Runs** page opens.
4. Select the checkbox next to each run that you want to compare, and then click **Compare runs**.
You can compare a maximum of 10 runs at one time.
The **Compare runs** page opens and displays available parameter, scalar metric, confusion matrix, and receiver operating characteristic (ROC) curve data for the runs that you selected.

- a. The **Run list** section displays a list of selected runs. You can filter the list by run name, experiment, pipeline version start date, duration, and status.
 - b. The **Parameters** section displays parameter information for each selected run. Set the **Hide parameters with no differences** switch to **On** to hide parameters that have the same values.
 - c. The **Metrics** section displays scalar metric, confusion matrix, and ROC curve data for all selected runs.
 - i. On the **Scalar metrics** tab, set the **Hide parameters with no differences** switch to **On** to hide parameters that have the same values.
 - ii. On the **ROC curve** tab, in the artifacts list, adjust the ROC curve chart by deselecting the checkbox next to artifacts that you want to remove from the chart.
5. To select different runs for comparison, click **Manage runs**.
The **Manage runs** dialog opens.
- a. From the **Search** filter drop-down list, select **Run**, **Experiment**, **Pipeline version**, **Start date**, or **Status** to filter the run list by each value.
 - b. Deselect the checkbox next to each run that you want to remove from your comparison.
 - c. Select the checkbox next to each run that you want to add to your comparison.
6. Click **Update**.

Verification

- The **Compare runs** page opens and displays data for the runs that you selected.

CHAPTER 4. MANAGING PIPELINE RUNS

4.1. OVERVIEW OF PIPELINE RUNS

A pipeline run is a single execution of a data science pipeline. As data scientist, you can use OpenShift AI to define, manage, and track executions of a data science pipeline. You can view a record of previously executed, scheduled, and archived runs from the **Runs** page in the OpenShift AI user interface.

You can optimize your use of pipeline runs for portability and repeatability by using pipeline experiments. With experiments, you can logically group pipeline runs and try different configurations of your pipelines. You can also clone your pipeline runs to reproduce and scale them, or archive them when you want to retain a record of their execution, but no longer require them. You can delete archived runs that you no longer want to retain, or you can restore them to their former state.

You can execute a run once, that is, immediately after its creation, or on a recurring basis. Recurring runs consist of a copy of a pipeline with all of its parameter values and a run trigger. A run trigger indicates when a recurring run executes. You can define the following run triggers:

- **Periodic:** used for scheduling runs to execute in intervals.
- **Cron:** used for scheduling runs as a cron job.

You can also configure multiple instances of the same run to execute concurrently, from a range of one to ten. When executed, you can track the run's progress from the run **Details** page on the OpenShift AI user interface. From here, you can view the run's graph, and output artifacts. A pipeline run can be in one of the following states:

- **Scheduled:** A pipeline run that is scheduled to execute at least once
- **Active:** A pipeline run that is executing, or stopped.
- **Archived:** An archived pipeline run.

You can use catch up runs to ensure your pipeline runs do not permanently fall behind schedule when paused. For example, if you re-enable a paused recurring run, the run scheduler backfills each missed run interval. If you disable catch up runs, and you have a scheduled run interval ready to execute, the run scheduler only schedules the run execution for the latest run interval. Catch up runs are enabled by default. However, if your pipeline handles backfill internally, Red Hat recommends that you disable catch up runs to avoid duplicate backfill.

After a pipeline run executes, you can view details of its executed tasks on the **Executions** page, along with its artifacts, on the **Artifacts** page. From the **Executions** page, you can view the execution status of each task, which indicates whether it completed successfully. You can also view further information about each executed task by clicking the execution name in the list. From the **Artifacts** page, you can view the details of each pipeline artifact, such as its name, unique ID, type, and URI. Pipeline artifacts can help you to evaluate the performance of your pipeline runs and make it easier to understand your pipeline components. Pipeline artifacts can range from plain text data or detailed, interactive data visualizations.

You can review and analyze logs for each step in an active pipeline run. With the log viewer, you can search for specific log messages, view the log for each step, and download the step logs to your local machine.

4.2. STORING DATA WITH DATA SCIENCE PIPELINES

When you run a data science pipeline, OpenShift AI stores the pipeline YAML configuration file and resulting pipeline run artifacts in the **root** directory of your storage bucket. The directories that contain pipeline run artifacts can differ depending on where you executed the pipeline run from. See the following table for further information:

Table 4.1. Pipeline configuration file and artifacts storage locations

Pipeline run source	Pipeline storage directory	Run artifacts storage directory
OpenShift AI dashboard	<p>/pipelines/<pipeline_version_id></p> <p>Example: /pipelines/1d01c4eb-d2ab-4916-9935-a73a5580f1fb</p>	<p>/<pipeline_name>/<pipeline_run_id></p> <p>Example: iris-training-pipeline/2g48k8pw-a8ib-4884-9145-h41j7599h3ds</p>
JupyterLab Elyra extension	<p>/pipelines/<pipeline_version_id></p>	<p>/<pipeline_name_timestamp></p> <p>Example: /hello-generic-world-0523161704</p> <p>With the JupyterLab Elyra extension, you can also set an object storage path prefix</p> <p>Example: /iris-project/hello-generic-world-0523161704</p>

4.3. VIEWING ACTIVE PIPELINE RUNS

You can view a list of pipeline runs that were previously executed in a pipeline experiment. From this list, you can view details relating to your pipeline runs, such as the pipeline version that the run belongs to, along with the run status, duration, and execution start time.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously executed a pipeline run that is available.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the active pipeline runs that you want to view.

3. From the list of experiments, click the experiment that contains the active pipeline runs that you want to view.
4. On the **Runs** page, click the **Runs** tab.
After a run has completed its execution, the run's status appears in the **Status** column in the table, indicating whether the run has succeeded or failed.

Verification

- A list of active runs appears in the **Runs** tab on the **Runs** page for the pipeline experiment.

4.4. EXECUTING A PIPELINE RUN

By default, a pipeline run executes once immediately after it is created.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**.
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment that you want to create a run for.
3. From the list of pipeline experiments, click the experiment that you want to create a run for.
4. Click **Create run**.
5. On the **Create run** page, configure the run:
 - a. From the **Experiment** list, select the pipeline experiment that you want to create a run for. Alternatively, to create a new pipeline experiment, click **Create new experiment**, and then complete the relevant fields in the **Create experiment** dialog.
 - b. In the **Name** field, enter a name for the run.
 - c. In the **Description** field, enter a description for the run.
 - d. From the **Pipeline** list, select the pipeline that you want to create a run for. Alternatively, to create a new pipeline, click **Create new pipeline**, and then complete the relevant fields in the **Import pipeline** dialog.
 - e. From the **Pipeline version** list, select the pipeline version to create a run for. Alternatively, to upload a new version, click **Upload new version**, and then complete the relevant fields in the **Upload new version** dialog.

- f. Configure the input parameters for the run by selecting the parameters from the list.
- g. Click **Create run**.

Verification

- The pipeline run that you created appears in the **Runs** tab on the **Runs** page for the pipeline experiment.

4.5. STOPPING AN ACTIVE PIPELINE RUN

If you no longer require an active pipeline run to continue executing in a pipeline experiment, you can stop the run before its defined end date.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- There is a previously created data science project available that contains a pipeline server.
- You have imported a pipeline to an active pipeline server.
- An active pipeline run is currently executing.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the active run that you want to stop.
3. From the list of pipeline experiments, click the pipeline experiment that contains the run that you want to stop.
4. In the **Runs** tab, click the action menu (**⋮**) beside the active run that you want to stop, and then click **Stop**.
There might be a short delay while the run stops.

Verification

- In the list of active runs, the status of the run is "stopped".

4.6. DUPLICATING AN ACTIVE PIPELINE RUN

To make it easier to quickly execute pipeline runs with the same configuration in a pipeline experiment, you can duplicate them.

Prerequisites

- You have logged in to Red Hat OpenShift AI.

- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.
- An active run is available to duplicate in the **Active** tab on the **Runs** page.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the pipeline run that you want to duplicate.
3. From the list of pipeline experiments, click the experiment that contains the pipeline run that you want to duplicate.
4. Click the action menu (**⋮**) beside the relevant active run and click **Duplicate**.
5. In the **Duplicate run** page, configure the duplicate run:
 - a. From the **Experiment** list, select the pipeline experiment that contains the pipeline run that you want to duplicate. Alternatively, to create a new pipeline experiment, click **Create new experiment**, and then complete the relevant fields in the **Create experiment** dialog.
 - b. In the **Name** field, enter a name for the duplicate run.
 - c. In the **Description** field, enter a description for the duplicate run.
 - d. From the **Pipeline** list, select the pipeline to contain the duplicate run. Alternatively, to create a new pipeline, click **Create new pipeline**, and then complete the relevant fields in the **Import pipeline** dialog.
 - e. From the **Pipeline version** list, select the pipeline version to contain the duplicate run. Alternatively, to upload a new version, click **Upload new version**, and then complete the relevant fields in the **Upload new version** dialog.
 - f. In the **Parameters** section, configure the input parameters for the run that you are duplicating by selecting the appropriate parameters from the list.
 - g. Click **Create run**.

Verification

- The duplicate pipeline run appears in the **Runs** tab on the **Runs** page for the pipeline experiment.

4.7. VIEWING SCHEDULED PIPELINE RUNS

You can view a list of pipeline runs that are scheduled for execution in a pipeline experiment. From this list, you can view details relating to your pipeline runs, such as the pipeline version that the run belongs to. You can also view the run status, execution frequency, and schedule.

Prerequisites

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have scheduled a pipeline run that is available to view.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the scheduled pipeline runs that you want to view.
3. From the list of pipeline experiments, click the experiment that contains the pipeline runs that you want to view.
4. On the **Runs** page, click the **Schedules** tab.
5. Study the table showing a list of scheduled runs.
After a run has been scheduled, the run's status indicates whether the run is ready for execution or unavailable for execution. To change its execution availability, click the run's Status icon.

Verification

- A list of scheduled runs appears in the **Schedules** tab on the **Runs** page for the pipeline experiment.

4.8. SCHEDULING A PIPELINE RUN USING A CRON JOB

You can use a cron job to schedule a pipeline run to execute at a specific time. Cron jobs are useful for creating periodic and recurring tasks, and can also schedule individual tasks for a specific time, such as if you want to schedule a run for a low activity period. To successfully execute runs in OpenShift AI, you must use the supported format. See [Cron Expression Format](#) for more information.

The following examples show the correct format:

Run occurrence	Cron format
Every five minutes	@every 5m
Every 10 minutes	0 */10 * * * *
Daily at 16:16 UTC	0 16 16 * * *
Daily every quarter of the hour	0 0,15,30,45 * * * *

Run occurrence	Cron format
On Monday and Tuesday at 15:40 UTC	0 40 15 * * MON,TUE

Additional resources

- [Cron Expression Format](#)

4.9. SCHEDULING A PIPELINE RUN


To repeatedly run a pipeline, you can create a scheduled pipeline run.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**.
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the run that you want to schedule.
3. From the list of pipeline experiments, click the experiment that contains the run that you want to schedule.
4. Click the **Schedules** tab.
5. Click **Create schedule**.
6. On the **Schedule run** page, configure the run that you are scheduling:
 - a. From the **Experiment** list, select the pipeline experiment that you want to contain the scheduled run. Alternatively, to create a new pipeline experiment, click **Create new experiment**, and then complete the relevant fields in the **Create experiment** dialog.
 - b. In the **Name** field, enter a name for the run.
 - c. In the **Description** field, enter a description for the run.
 - d. From the **Trigger type** list, select one of the following options:
 - Select **Periodic** to specify an execution frequency. In the **Run every** field, enter a numerical value and select an execution frequency from the list.
 - Select **Cron** to specify the execution schedule in **cron** format in the **Cron string** field.

This creates a cron job to execute the run. Click the **Copy** button () to copy the cron job schedule to the clipboard. The field furthest to the left represents seconds. For more information about scheduling tasks using the supported **cron** format, see [Cron Expression Format](#).

- e. In the **Maximum concurrent runs** field, specify the number of runs that can execute concurrently, from a range of one to ten.
- f. For **Start date**, specify a start date for the run. Select a start date using the **Calendar** and the start time from the list of times.
- g. For **End date**, specify an end date for the run. Select an end date using the **Calendar** and the end time from the list of times.
- h. For **Catch up**, enable or disable catch up runs. You can use catch up runs to ensure your pipeline runs do not permanently fall behind schedule when paused. For example, if you re-enable a paused recurring run, the run scheduler backfills each missed run interval.
- i. From the **Pipeline** list, select the pipeline that you want to create a run for. Alternatively, to create a new pipeline, click **Create new pipeline**, and then complete the relevant fields in the **Import pipeline** dialog.
- j. From the **Pipeline version** list, select the pipeline version to create a run for. Alternatively, to upload a new version, click **Upload new version**, and then complete the relevant fields in the **Upload new version** dialog.
- k. Configure the input parameters for the run by selecting the parameters from the list.
- l. Click **Schedule run**.

Verification

- The pipeline run that you scheduled appears in the **Schedules** tab on the **Runs** page for the pipeline experiment.


4.10. DUPLICATING A SCHEDULED PIPELINE RUN


To make it easier to schedule runs to execute as part of your pipeline experiment, you can duplicate existing scheduled runs.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.
- A scheduled run is available to duplicate in the **Schedules** tab on the **Runs** page.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the scheduled run that you want to duplicate.
3. From the list of pipeline experiments, click the experiment that contains the pipeline run that you want to duplicate.
4. On the **Runs** page, click the **Schedules** tab.
5. Click the action menu () beside the run that you want to duplicate and click **Duplicate**.
6. On the **Duplicate schedule** page, configure the duplicate run:
 - a. From the **Experiment** list, select the pipeline experiment that contains the scheduled pipeline run that you want to duplicate. Alternatively, to create a new pipeline experiment, click **Create new experiment**, and then complete the relevant fields in the **Create experiment** dialog.
 - b. In the **Name** field, enter a name for the duplicate run.
 - c. In the **Description** field, enter a description for the duplicate run.
 - d. From the **Trigger type** list, select one of the following options:
 - Select **Periodic** to specify an execution frequency. In the **Run every** field, enter a numerical value and select an execution frequency from the list.
 - Select **Cron** to specify the execution schedule in **cron** format in the **Cron string** field.

This creates a cron job to execute the run. Click the **Copy** button () to copy the cron job schedule to the clipboard. The field furthest to the left represents seconds. For more information about scheduling tasks using the supported **cron** format, see [Cron Expression Format](#).
 - e. For **Maximum concurrent runs**, specify the number of runs that can execute concurrently, from a range of one to ten.
 - f. For **Start date**, specify a start date for the duplicate run. Select a start date using the **Calendar** and the start time from the list of times.
 - g. For **End date**, specify an end date for the duplicate run. Select an end date using the **Calendar** and the end time from the list of times.
 - h. For **Catch up**, enable or disable catch up runs. You can use catch up runs to ensure your pipeline runs do not permanently fall behind schedule when paused. For example, if you re-enable a paused recurring run, the run scheduler backfills each missed run interval.
 - i. From the **Pipeline** list, select the pipeline that you want to create a duplicate run for. Alternatively, to create a new pipeline, click **Create new pipeline**, and then complete the relevant fields in the **Import pipeline** dialog.
 - j. From the **Pipeline version** list, select the pipeline version to create a duplicate run for. Alternatively, to upload a new version, click **Upload new version**, and then complete the relevant fields in the **Upload new version** dialog.
 - k. Configure the input parameters for the run by selecting the parameters from the list.

- I. Click **Schedule run**.

Verification

- The pipeline run that you duplicated appears in the **Schedules** tab on the **Runs** page for the pipeline experiment.

4.11. DELETING A SCHEDULED PIPELINE RUN

To discard pipeline runs that you previously scheduled, but no longer require, you can delete them so that they do not appear on the **Schedules** page.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously scheduled a run that is available to delete.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the scheduled pipeline run that you want to delete.
3. From the list of pipeline experiments, click the experiment that contains the scheduled pipeline run that you want to delete.
4. On the **Runs** page, click the **Schedules** tab.
5. Click the action menu (**⋮**) beside the scheduled pipeline run that you want to delete, and then click **Delete**.
6. In the **Delete schedule** dialog, enter the run name in the text field to confirm that you intend to delete it.
7. Click **Delete**.

Verification

- The run that you deleted no longer appears on the **Schedules** tab for the pipeline experiment.



4.12. VIEWING THE DETAILS OF A PIPELINE RUN

To gain a clearer understanding of your pipeline runs, you can view the details of a previously triggered pipeline run, such as its graph, execution details, and run output.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously triggered a pipeline run.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines** → **Pipelines**.
2. On the **Pipelines** page, select the **project** that you want to view run details for.
3. For a pipeline that you want to view run details for, click **Expand** ().
4. Click the action menu () for the pipeline version and then click **View runs**.
5. On the **Runs** page, click the name of the run that you want to view the details of.

Verification

- On the **Run details** page, you can view the run's graph, execution details, input parameters, step logs, and run output.

4.13. VIEWING ARCHIVED PIPELINE RUNS

You can view a list of pipeline runs that you have archived. You can view details for your archived pipeline runs, such as the pipeline version, run status, duration, and execution start date.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- You have imported a pipeline to an active pipeline server.
- An archived pipeline run exists.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the archived pipeline runs that you want to view.

3. From the list of pipeline experiments, click the experiment that contains the archived pipeline runs that you want to view.
4. On the **Runs** page, click the **Archive** tab.

Verification

- A list of archived runs appears in the **Archive** tab on the **Runs** page for the pipeline experiment.

4.14. ARCHIVING A PIPELINE RUN

You can retain records of your pipeline runs by archiving them. If required, you can restore runs from your archive to reuse, or delete runs that are no longer required.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously executed a pipeline run that is available.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment for the run that you want to archive.
3. From the list of pipeline experiments, click the experiment that contains the pipeline run that you want to archive.
4. On the **Runs** page, click the **Runs** tab.
5. Click the action menu (**⋮**) beside the pipeline run that you want to archive, and then click **Archive**.
6. In the **Archiving run** dialog, enter the run name in the text field to confirm that you intend to archive it.
7. Click **Archive**.

Verification

- The archived run does not appear in the **Runs** tab, and instead appears in the **Archive** tab on the **Runs** page for the pipeline experiment.


4.15. RESTORING AN ARCHIVED PIPELINE RUN

You can restore an archived run to the active state.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- You have imported a pipeline to an active pipeline server.
- An archived run exists in your project.

Procedure

1. From the OpenShift AI dashboard, click **Experiments** → **Experiments and runs**
2. On the **Experiments** page, from the **Project** drop-down list, select the project that contains the pipeline experiment that you want to restore.
3. From the list of pipeline experiments, click the experiment that contains the archived pipeline run that you want to restore.
4. On the **Runs** page, click the **Archive** tab.
5. Click the action menu () beside the pipeline run that you want to restore, and then click **Restore**.
6. In the **Restore run** dialog, click **Restore**.

Verification

- The restored run appears in the **Runs** tab on the **Runs** page for the pipeline experiment.

4.16. DELETING AN ARCHIVED PIPELINE RUN


You can delete pipeline runs from the OpenShift AI run archive.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and has a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously archived a pipeline run.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines** → **Runs**.
2. On the **Runs** page, select the **project** for the archived pipeline run you want to delete.

3. In the **Run details** page, click **Archived**.
4. Click the action menu () beside the run that you want to delete and click **Delete**.
5. In the **Delete run** dialog, enter the run name in the text field to confirm that you intend to delete it.
6. Click **Delete**.

Verification

- The archived run that you deleted no longer appears in the **Archived** tab on the **Runs** page.


4.17. DUPLICATING AN ARCHIVED PIPELINE RUN

To make it easier to reproduce runs with the same configuration as runs in your archive, you can duplicate them.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a configured pipeline server.
- You have imported a pipeline to an active pipeline server.
- An archived run is available to duplicate in the **Archived** tab on the **Runs** page.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines** → **Runs**.
2. On the **Runs** page, select the **project** that has the pipeline run that you want to duplicate.
3. Click the **Archived** tab.
4. Click the action menu () beside the relevant archived run and click **Duplicate**.
5. On the **Duplicate run** page, configure the duplicate run:
 - a. In the **Name** field, enter a name for the duplicate run.
 - b. In the **Description** field, enter a description for the duplicate run.
 - c. From the **Pipeline** list, select the pipeline to contain the duplicate run.
 - d. From the **Pipeline version** list, select the pipeline version to contain the duplicate run.
 - e. In the **Parameters** section, configure the input parameters for the run that you are duplicating by selecting the appropriate parameters from the list.
 - f. Click **Create run**.

Verification

- The duplicate pipeline run appears in the **Active** tab on the **Runs** page.

CHAPTER 5. WORKING WITH PIPELINE LOGS

5.1. ABOUT PIPELINE LOGS

You can review and analyze step logs for each step in a triggered pipeline run.

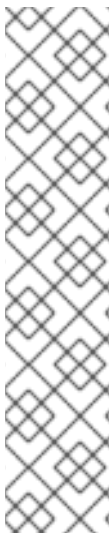
To help you troubleshoot and audit your pipelines, you can review and analyze these step logs by using the log viewer in the OpenShift AI dashboard. From here, you can search for specific log messages, view the log for each step, and download the step logs to your local machine.

If the step log file exceeds its capacity, a warning appears above the log viewer stating that the log window displays partial content. Expanding the warning displays further information, such as how the log viewer refreshes every three seconds, and that each step log displays the last 500 lines of log messages received. In addition, you can click **download all step logs** to download all step logs to your local machine.

Each step has a set of container logs. You can view these container logs by selecting a container from the **Steps** list in the log viewer. The **Step-main** container log consists of the log output for the step. The **step-copy-artifact** container log consists of output relating to artifact data sent to s3-compatible storage. If the data transferred between the steps in your pipeline is larger than 3 KB, five container logs are typically available. These logs contain output relating to data transferred between your persistent volume claims (PVCs).

5.2. VIEWING PIPELINE STEP LOGS

To help you troubleshoot and audit your pipelines, you can review and analyze the log of each pipeline step using the log viewer. From here, you can search for specific log messages and download the logs for each step in your pipeline. If the pipeline is running, you can also pause and resume the log from the log viewer.



NOTE

Logs are no longer stored in S3-compatible storage for Python scripts which are running in Elyra pipelines. From OpenShift AI version 2.11, you can view these logs in the pipeline step log viewer.

For this change to take effect, you must be using the latest runtime images for Elyra, which are provided in the 2024.1 workbench images.

If you have an older workbench image version, update the **Version selection** field to **2024.1**, as described in [Updating a project workbench](#).





Updating your workbench image version will clear any existing runtime image selections for your pipeline. After you update your workbench version, open your workbench IDE and update the properties of your pipeline to select a runtime image.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.

- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.
- You have previously triggered a pipeline run.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines** → **Pipelines**.
2. On the **Pipelines** page, select the **project** that you want to view logs for.
3. For the pipeline that you want to view logs for, click **Expand** ().
4. Click the action menu () on the row containing the project version that you want to view pipeline logs for and click **View runs**.
5. On the **Runs** page, click the name of the run that you want to view logs for.
6. On the graph on the **Run details** page, click the pipeline step that you want to view logs for.
7. Click the **Logs** tab.
8. To view the logs of another pipeline step, from the **Steps** list, select the step that you want to view logs for.
9. Analyze the log using the log viewer.
 - To search for a specific log message, enter at least part of the message in the search bar.
 - To view the full log in a separate browser window, click the action menu () and select **View raw logs**. Alternatively, to expand the size of the log viewer, click the action menu () and select **Expand**.

Verification

- You can view the logs for each step in your pipeline.

5.3. DOWNLOADING PIPELINE STEP LOGS



Instead of viewing the step logs of a pipeline run using the log viewer on the OpenShift AI dashboard, you can download them for further analysis. You can choose to download the logs belonging to all steps in your pipeline, or you can download the log only for the step log displayed in the log viewer.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have previously created a data science project that is available and contains a pipeline server.
- You have imported a pipeline to an active pipeline server.

- You have previously triggered a pipeline run.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines** → **Pipelines**.
2. On the **Pipelines** page, select the **project** that you want to download logs for.
3. For the pipeline that you want to download logs for, click **Expand** ().
4. Click **View runs** on the row containing the pipeline version that you want to download logs for.
5. On the **Runs** page, click the name of the run that you want to download logs for.
6. On the graph on the **Run details** page, click the pipeline step that you want to download logs for.
7. Click the **Logs** tab.
8. In the log viewer, click the **Download** button ().
 - a. Select **Download current step log** to download the log for the current pipeline step.
 - b. Select **Download all step logs** to download the logs for all steps in your pipeline run.

Verification

- The step logs download to your browser's default directory for downloaded files.

CHAPTER 6. WORKING WITH PIPELINES IN JUPYTERLAB

6.1. OVERVIEW OF PIPELINES IN JUPYTERLAB

You can use Elyra to create visual end-to-end pipeline workflows in JupyterLab. Elyra is an extension for JupyterLab that provides you with a Pipeline Editor to create pipeline workflows that can be executed in OpenShift AI.

You can access the Elyra extension within JupyterLab when you create the most recent version of one of the following notebook images:

- Standard Data Science
- PyTorch
- TensorFlow
- TrustyAI
- HabanaAI

When you use the Pipeline Editor to visually design your pipelines, minimal coding is required to create and run pipelines. For more information about Elyra, see [Elyra Documentation](#). For more information about the Pipeline Editor, see [Visual Pipeline Editor](#). After you have created your pipeline, you can run it locally in JupyterLab, or remotely using data science pipelines in OpenShift AI.

The pipeline creation process consists of the following tasks:

- Create a data science project that contains a workbench.
- Create a pipeline server.
- Create a new pipeline in the Pipeline Editor in JupyterLab.
- Develop your pipeline by adding Python notebooks or Python scripts and defining their runtime properties.
- Define execution dependencies.
- Run or export your pipeline.

Before you can run a pipeline in JupyterLab, your pipeline instance must contain a runtime configuration. A runtime configuration defines connectivity information for your pipeline instance and S3-compatible cloud storage.

If you create a workbench as part of a data science project, a default runtime configuration is created automatically. However, if you create a notebook from the Jupyter tile in the OpenShift AI dashboard, you must create a runtime configuration before you can run your pipeline in JupyterLab. For more information about runtime configurations, see [Runtime Configuration](#). As a prerequisite, before you create a workbench, ensure that you have created and configured a pipeline server within the same data science project as your workbench.

You can use S3-compatible cloud storage to make data available to your notebooks and scripts while they are executed. Your cloud storage must be accessible from the machine in your deployment that runs JupyterLab and from the cluster that hosts data science pipelines. Before you create and run pipelines in JupyterLab, ensure that you have your s3-compatible storage credentials readily available.

Additional resources

- [Elyra Documentation](#)
- [Visual Pipeline Editor](#)
- [Runtime Configuration](#).

6.2. ACCESSING THE PIPELINE EDITOR

You can use Elyra to create visual end-to-end pipeline workflows in JupyterLab. Elyra is an extension for JupyterLab that provides you with a Pipeline Editor to create pipeline workflows that can execute in OpenShift AI.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have created a data science project.
- You have created a workbench with the **Standard Data Science** notebook image.
- You have created and configured a pipeline server within the data science project that contains your workbench.
- You have created and launched a Jupyter server from a notebook image that contains the Elyra extension (Standard data science, TensorFlow, TrustyAI, PyTorch, or HabanaAI).
- You have access to S3-compatible storage.

Procedure

1. After you open JupyterLab, confirm that the JupyterLab launcher is automatically displayed.
2. In the **Elyra** section of the JupyterLab launcher, click the **Pipeline Editor** tile. The Pipeline Editor opens.

Verification

- You can view the Pipeline Editor in JupyterLab.

6.3. CREATING A RUNTIME CONFIGURATION



If you create a workbench as part of a data science project, a default runtime configuration is created automatically. However, if you create a notebook from the Jupyter tile in the OpenShift AI dashboard, you must create a runtime configuration before you can run your pipeline in JupyterLab. This enables you to specify connectivity information for your pipeline instance and S3-compatible cloud storage.

Prerequisites

- You have logged in to Red Hat OpenShift AI.

- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have access to S3-compatible cloud storage.
- You have created a data science project that contains a workbench.
- You have created and configured a pipeline server within the data science project that contains your workbench.
- You have created and launched a Jupyter server from a notebook image that contains the Elyra extension (Standard data science, TensorFlow, TrustyAI, PyTorch, or HabanaAI).

Procedure

1. In the left sidebar of JupyterLab, click **Runtimes** ().
2. Click the **Create new runtime configuration** button ().
The **Add new Data Science Pipelines runtime configuration** page opens.
3. Complete the relevant fields to define your runtime configuration.
 - a. In the **Display Name** field, enter a name for your runtime configuration.
 - b. Optional: In the **Description** field, enter a description to define your runtime configuration.
 - c. Optional: In the **Tags** field, click **Add Tag** to define a category for your pipeline instance. Enter a name for the tag and press Enter.
 - d. Define the credentials of your data science pipeline:
 - i. In the **Data Science Pipelines API Endpoint** field, enter the API endpoint of your data science pipeline. Do not specify the pipelines namespace in this field.
 - ii. In the **Public Data Science Pipelines API Endpoint** field, enter the public API endpoint of your data science pipeline.



IMPORTANT

You can obtain the data science pipelines API endpoint from the **Data Science Pipelines → Runs** page in the dashboard. Copy the relevant endpoint and enter it in the **Public Data Science Pipelines API Endpoint** field.

- iii. Optional: In the **Data Science Pipelines User Namespace** field, enter the relevant user namespace to run pipelines.
- iv. From the **Authentication Type** list, select the authentication type required to authenticate your pipeline.

**IMPORTANT**

If you created a notebook directly from the Jupyter tile on the dashboard, select **EXISTING_BEARER_TOKEN** from the **Authentication Type** list.


- v. In the **Data Science Pipelines API Endpoint Username** field, enter the user name required for the authentication type.
- vi. In the **Data Science Pipelines API Endpoint Password Or Token** field, enter the password or token required for the authentication type.

**IMPORTANT**

To obtain the data science pipelines API endpoint token, in the upper-right corner of the OpenShift web console, click your user name and select **Copy login command**. After you have logged in, click **Display token** and copy the value of **--token=** from the **Log in with this token** command.

- e. Define the connectivity information of your S3-compatible storage:
 - i. In the **Cloud Object Storage Endpoint** field, enter the endpoint of your S3-compatible storage. For more information about Amazon s3 endpoints, see [Amazon Simple Storage Service endpoints and quotas](#).
 - ii. Optional: In the **Public Cloud Object Storage Endpoint** field, enter the URL of your S3-compatible storage.
 - iii. In the **Cloud Object Storage Bucket Name** field, enter the name of the bucket where your pipeline artifacts are stored. If the bucket name does not exist, it is created automatically.
 - iv. From the **Cloud Object Storage Authentication Type** list, select the authentication type required to access to your S3-compatible cloud storage. If you use AWS S3 buckets, select **KUBERNETES_SECRET** from the list.
 - v. In the **Cloud Object Storage Credentials Secret** field, enter the secret that contains the storage user name and password. This secret is defined in the relevant user namespace, if applicable. In addition, it must be stored on the cluster that hosts your pipeline runtime.
 - vi. In the **Cloud Object Storage Username** field, enter the user name to connect to your S3-compatible cloud storage, if applicable. If you use AWS S3 buckets, enter your AWS Secret Access Key ID.
 - vii. In the **Cloud Object Storage Password** field, enter the password to connect to your S3-compatible cloud storage, if applicable. If you use AWS S3 buckets, enter your AWS Secret Access Key.
- f. Click **Save & Close**.

Verification

- The runtime configuration that you created is shown in the **Runtimes** tab () in the left sidebar of JupyterLab.



6.4. UPDATING A RUNTIME CONFIGURATION

To ensure that your runtime configuration is accurate and updated, you can change the settings of an existing runtime configuration.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have access to S3-compatible storage.
- You have created a data science project that contains a workbench.
- You have created and configured a pipeline server within the data science project that contains your workbench.
- A previously created runtime configuration is available in the JupyterLab interface.
- You have created and launched a Jupyter server from a notebook image that contains the Elyra extension (Standard data science, TensorFlow, TrustyAI, PyTorch, or HabanaAI).

Procedure

1. In the left sidebar of JupyterLab, click **Runtimes** ().
2. Hover the cursor over the runtime configuration that you want to update and click the **Edit** button ().
The **Data Science Pipelines runtime configuration** page opens.
3. Fill in the relevant fields to update your runtime configuration.
 - a. In the **Display Name** field, update name for your runtime configuration, if applicable.
 - b. Optional: In the **Description** field, update the description of your runtime configuration, if applicable.
 - c. Optional: In the **Tags** field, click **Add Tag** to define a category for your pipeline instance. Enter a name for the tag and press Enter.
 - d. Define the credentials of your data science pipeline:
 - i. In the **Data Science Pipelines API Endpoint** field, update the API endpoint of your data science pipeline, if applicable. Do not specify the pipelines namespace in this field.
 - ii. In the **Public Data Science Pipelines API Endpoint** field, update the API endpoint of your data science pipeline, if applicable.

- iii. Optional: In the **Data Science Pipelines User Namespace** field, update the relevant user namespace to run pipelines, if applicable.
- iv. From the **Authentication Type** list, select a new authentication type required to authenticate your pipeline, if applicable.



IMPORTANT

If you created a notebook directly from the Jupyter tile on the dashboard, select **EXISTING_BEARER_TOKEN** from the **Authentication Type** list.

- v. In the **Data Science Pipelines API Endpoint Username** field, update the user name required for the authentication type, if applicable.
- vi. In the **Data Science Pipelines API Endpoint Password Or Token** update the password or token required for the authentication type, if applicable.




IMPORTANT

To obtain the data science pipelines API endpoint token, in the upper-right corner of the OpenShift web console, click your user name and select **Copy login command**. After you have logged in, click **Display token** and copy the value of **--token=** from the **Log in with this token** command.

- e. Define the connectivity information of your S3-compatible storage:
 - i. In the **Cloud Object Storage Endpoint** field, update the endpoint of your S3-compatible storage, if applicable. For more information about Amazon s3 endpoints, see [Amazon Simple Storage Service endpoints and quotas](#) .
 - ii. Optional: In the **Public Cloud Object Storage Endpoint** field, update the URL of your S3-compatible storage, if applicable.
 - iii. In the **Cloud Object Storage Bucket Name** field, update the name of the bucket where your pipeline artifacts are stored, if applicable. If the bucket name does not exist, it is created automatically.
 - iv. From the **Cloud Object Storage Authentication Type** list, update the authentication type required to access to your S3-compatible cloud storage, if applicable. If you use AWS S3 buckets, you must select **USER_CREDENTIALS** from the list.
 - v. Optional: In the **Cloud Object Storage Credentials Secret** field, update the secret that contains the storage user name and password, if applicable. This secret is defined in the relevant user namespace. You must save the secret on the cluster that hosts your pipeline runtime.
 - vi. Optional: In the **Cloud Object Storage Username** field, update the user name to connect to your S3-compatible cloud storage, if applicable. If you use AWS S3 buckets, update your AWS Secret Access Key ID.
 - vii. Optional: In the **Cloud Object Storage Password** field, update the password to connect to your S3-compatible cloud storage, if applicable. If you use AWS S3 buckets, update your AWS Secret Access Key.

- f. Click **Save & Close**.

Verification

- The runtime configuration that you updated is shown in the **Runtimes** tab () in the left sidebar of JupyterLab.



6.5. DELETING A RUNTIME CONFIGURATION

After you have finished using your runtime configuration, you can delete it from the JupyterLab interface. After deleting a runtime configuration, you cannot run pipelines in JupyterLab until you create another runtime configuration.


Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have created a data science project that contains a workbench.
- You have created and configured a pipeline server within the data science project that contains your workbench.
- A previously created runtime configuration is visible in the JupyterLab interface.
- You have created and launched a Jupyter server from a notebook image that contains the Elyra extension (Standard data science, TensorFlow, TrustyAI, PyTorch, or HabanaAI).

Procedure

1. In the left sidebar of JupyterLab, click **Runtimes** ().
2. Hover the cursor over the runtime configuration that you want to delete and click the **Delete Item** button ().
A dialog box appears prompting you to confirm the deletion of your runtime configuration.
3. Click **OK**.

Verification

- The runtime configuration that you deleted is no longer shown in the **Runtimes** tab () in the left sidebar of JupyterLab.

6.6. DUPLICATING A RUNTIME CONFIGURATION



To prevent you from re-creating runtime configurations with similar values in their entirety, you can duplicate an existing runtime configuration in the JupyterLab interface.

Prerequisites


Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have created a data science project that contains a workbench.
- You have created and configured a pipeline server within the data science project that contains your workbench.
- A previously created runtime configuration is visible in the JupyterLab interface.
- You have created and launched a Jupyter server from a notebook image that contains the Elyra extension (Standard data science, TensorFlow, TrustyAI, PyTorch, or HabanaAI).

Procedure

1. In the left sidebar of JupyterLab, click **Runtimes** ().
2. Hover the cursor over the runtime configuration that you want to duplicate and click the **Duplicate** button ().

Verification

- The runtime configuration that you duplicated is shown in the **Runtimes** tab () in the left sidebar of JupyterLab.

6.7. RUNNING A PIPELINE IN JUPYTERLAB


You can run pipelines that you have created in JupyterLab from the Pipeline Editor user interface. Before you can run a pipeline, you must create a data science project and a pipeline server. After you create a pipeline server, you must create a workbench within the same project as your pipeline server. Your pipeline instance in JupyterLab must contain a runtime configuration. If you create a workbench as part of a data science project, a default runtime configuration is created automatically. However, if you create a notebook from the Jupyter tile in the OpenShift AI dashboard, you must create a runtime configuration before you can run your pipeline in JupyterLab. A runtime configuration defines connectivity information for your pipeline instance and S3-compatible cloud storage.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have access to S3-compatible storage.
- You have created a pipeline in JupyterLab.
- You have opened your pipeline in the Pipeline Editor in JupyterLab.

- Your pipeline instance contains a runtime configuration.
- You have created and configured a pipeline server within the data science project that contains your workbench.
- You have created and launched a Jupyter server from a notebook image that contains the Elyra extension (Standard data science, TensorFlow, TrustyAI, PyTorch, or HabanaAI).

Procedure

1. In the Pipeline Editor user interface, click **Run Pipeline** ().
The **Run Pipeline** dialog appears. The **Pipeline Name** field is automatically populated with the pipeline file name.
2. Define the settings for your pipeline run.
 - a. From the **Runtime Configuration** list, select the relevant runtime configuration to run your pipeline.
 - b. Optional: Configure your pipeline parameters, if applicable. If your pipeline contains nodes that reference pipeline parameters, you can change the default parameter values. If a parameter is required and has no default value, you must enter a value.
3. Click **OK**.

Verification

- You can view the output artifacts of your pipeline run. The artifacts are stored in your designated object storage bucket.

6.8. EXPORTING A PIPELINE IN JUPYTERLAB

You can export pipelines that you have created in JupyterLab. When you export a pipeline, the pipeline is prepared for later execution, but is not uploaded or executed immediately. During the export process, any package dependencies are uploaded to S3-compatible storage. Also, pipeline code is generated for the target runtime.


Before you can export a pipeline, you must create a data science project and a pipeline server. After you create a pipeline server, you must create a workbench within the same project as your pipeline server. In addition, your pipeline instance in JupyterLab must contain a runtime configuration. If you create a workbench as part of a data science project, a default runtime configuration is created automatically. However, if you create a notebook from the Jupyter tile in the OpenShift AI dashboard, you must create a runtime configuration before you can export your pipeline in JupyterLab. A runtime configuration defines connectivity information for your pipeline instance and S3-compatible cloud storage.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you are using specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You have created a data science project that contains a workbench.

- You have created and configured a pipeline server within the data science project that contains your workbench.
- You have access to S3-compatible storage.
- You have created a pipeline in JupyterLab.
- You have opened your pipeline in the Pipeline Editor in JupyterLab.
- Your pipeline instance contains a runtime configuration.
- You have created and launched a Jupyter server from a notebook image that contains the Elyra extension (Standard data science, TensorFlow, TrustyAI, PyTorch, or HabanaAI).

Procedure

1. In the Pipeline Editor user interface, click **Export Pipeline** ().
The **Export Pipeline** dialog appears. The **Pipeline Name** field is automatically populated with the pipeline file name.
2. Define the settings to export your pipeline.
 - a. From the **Runtime Configuration** list, select the relevant runtime configuration to export your pipeline.
 - b. From the **Export Pipeline** dialog, select an appropriate file format.
 - c. In the **Export Filename** field, enter a file name for the exported pipeline.
 - d. Select the **Replace if file already exists** check box to replace an existing file of the same name as the pipeline you are exporting.
 - e. Optional: Configure your pipeline parameters, if applicable. If your pipeline contains nodes that reference pipeline parameters, you can change the default parameter values. If a parameter is required and has no default value, you must enter a value.
3. Click **OK**.

Verification

- You can view the file containing the pipeline that you exported in your designated object storage bucket.

CHAPTER 7. ADDITIONAL RESOURCES

- [Kubeflow Pipelines 2.0 Documentation](#)