



Red Hat OpenShift AI Self-Managed 2.10

Working on data science projects

Organize your work in projects and workbenches, create and collaborate on notebooks, train and deploy models, configure model servers, and implement pipelines

Red Hat OpenShift AI Self-Managed 2.10 Working on data science projects

Organize your work in projects and workbenches, create and collaborate on notebooks, train and deploy models, configure model servers, and implement pipelines

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

Organize your work in projects and workbenches, create and collaborate on notebooks, train and deploy models, configure model servers, and implement pipelines.

Table of Contents

CHAPTER 1. WORKING ON DATA SCIENCE PROJECTS	4
1.1. USING DATA SCIENCE PROJECTS	4
1.1.1. Creating a data science project	4
1.1.2. Updating a data science project	5
1.1.3. Deleting a data science project	6
1.2. USING PROJECT WORKBENCHES	6
1.2.1. Creating a workbench and selecting an IDE	6
1.2.1.1. About workbench images	7
1.2.1.2. Creating a workbench	9
1.2.2. Starting a workbench	12
1.2.3. Updating a project workbench	12
1.2.4. Deleting a workbench from a data science project	13
1.3. USING DATA CONNECTIONS	14
1.3.1. Adding a data connection to your data science project	14
1.3.2. Deleting a data connection	15
1.3.3. Updating a connected data source	15
1.4. CONFIGURING CLUSTER STORAGE	16
1.4.1. Adding cluster storage to your data science project	16
1.4.2. Updating cluster storage	17
1.4.3. Deleting cluster storage from a data science project	18
1.5. CONFIGURING ACCESS TO DATA SCIENCE PROJECTS	19
1.5.1. Configuring access to data science projects	19
1.5.2. Sharing access to a data science project	19
1.5.3. Updating access to a data science project	20
1.5.4. Removing access to a data science project	22
CHAPTER 2. WORKING IN YOUR DATA SCIENCE IDE	23
2.1. ACCESSING YOUR WORKBENCH IDE	23
2.2. WORKING IN JUPYTERLAB	23
2.2.1. Creating and importing notebooks	23
2.2.1.1. Creating a new notebook	23
2.2.1.2. Uploading an existing notebook file from local storage	24
2.2.2. Collaborating on notebooks by using Git	24
2.2.3. Uploading an existing notebook file from a Git repository by using JupyterLab	25
2.2.3.1. Uploading an existing notebook file from a Git repository by using the command line interface	25
2.2.3.2. Updating your project with changes from a remote Git repository	26
2.2.3.3. Pushing project changes to a Git repository	26
2.2.4. Managing Python packages	27
2.2.4.1. Viewing Python packages installed on your notebook server	27
2.2.4.2. Installing Python packages on your notebook server	28
2.2.5. Troubleshooting common problems in Jupyter for users	29
2.2.5.1. I see a 403: Forbidden error when I log in to Jupyter	29
2.2.5.2. My notebook server does not start	30
2.2.5.3. I see a database or disk is full error or a no space left on device error when I run my notebook cells	30
2.3. WORKING IN CODE-SERVER	30
2.3.1. Installing extensions with code-server	31
CHAPTER 3. WORKING WITH DATA SCIENCE PIPELINES	33
3.1. ENABLING DATA SCIENCE PIPELINES 2.0	33
3.1.1. Installing OpenShift AI with DSP 2.0	34
3.1.2. Upgrading to DSP 2.0	34

3.1.3. Migrating pipelines from DSP 1.0 to 2.0	35
3.1.4. Accessing DSP 1.0 pipelines and history	35
3.1.5. Uninstalling the OpenShift Pipelines Operator	35
3.2. MANAGING DATA SCIENCE PIPELINES	36
3.2.1. Configuring a pipeline server	36
3.2.2. Defining a pipeline	38
3.2.3. Importing a data science pipeline	38
3.2.4. Downloading a data science pipeline	39
3.2.5. Deleting a data science pipeline	40
3.2.6. Deleting a pipeline server	40
3.2.7. Viewing the details of a pipeline server	41
3.2.8. Viewing existing pipelines	41
3.2.9. Overview of pipeline versions	42
3.2.10. Uploading a pipeline version	42
3.2.11. Deleting a pipeline version	43
3.2.12. Viewing pipeline versions	44
3.2.13. Viewing the details of a pipeline version	45
3.3. MANAGING PIPELINE RUNS	45
3.3.1. Overview of pipeline runs	46
3.3.2. Storing data with data science pipelines	46
3.3.3. Viewing active pipeline runs	47
3.3.4. Executing a pipeline run	48
3.3.5. Stopping an active pipeline run	48
3.3.6. Duplicating an active pipeline run	49
3.3.7. Viewing scheduled pipeline runs	50
3.3.8. Scheduling a pipeline run using a cron job	50
3.3.9. Scheduling a pipeline run	51
3.3.10. Duplicating a scheduled pipeline run	52
3.3.11. Deleting a scheduled pipeline run	54
3.3.12. Viewing the details of a pipeline run	54
3.3.13. Viewing archived pipeline runs	55
3.3.14. Archiving a pipeline run	55
3.3.15. Restoring an archived pipeline run	56
3.3.16. Deleting an archived pipeline run	57
3.3.17. Duplicating an archived pipeline run	57
3.4. WORKING WITH PIPELINE LOGS	58
3.4.1. About pipeline logs	58
3.4.2. Viewing pipeline step logs	59
3.4.3. Downloading pipeline step logs	60
3.5. WORKING WITH PIPELINES IN JUPYTERLAB	60
3.5.1. Overview of pipelines in JupyterLab	60
3.5.2. Accessing the pipeline editor	62
3.5.3. Creating a runtime configuration	62
3.5.4. Updating a runtime configuration	64
3.5.5. Deleting a runtime configuration	66
3.5.6. Duplicating a runtime configuration	67
3.5.7. Running a pipeline in JupyterLab	68
3.5.8. Exporting a pipeline in JupyterLab	69
3.6. ADDITIONAL RESOURCES	70

CHAPTER 1. WORKING ON DATA SCIENCE PROJECTS

As a data scientist, you can organize your data science work into a single project. A data science project in OpenShift AI can consist of the following components:

Workbenches

Creating a workbench allows you to add a Jupyter notebook to your project.

Cluster storage

For data science projects that require data to be retained, you can add cluster storage to the project.

Data connections

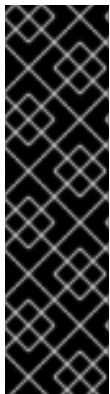
Adding a data connection to your project allows you to connect data inputs to your workbenches.

Pipelines

Standardize and automate machine learning workflows to enable you to further enhance and deploy your data science models.

Models and model servers

Deploy a trained data science model to serve intelligent applications. Your model is deployed with an endpoint that allows applications to send requests to the model.



IMPORTANT

If you create an OpenShift project outside of the OpenShift AI user interface, the project is not shown on the **Data Science Projects** page. In addition, you cannot use features exclusive to OpenShift AI, such as workbenches and model serving, with a standard OpenShift project.

To classify your OpenShift project as a data science project, and to make available features exclusive to OpenShift AI, you must add the label **opendatahub.io/dashboard: 'true'** to the project namespace. After you add this label, your project is subsequently shown on the **Data Science Projects** page.

1.1. USING DATA SCIENCE PROJECTS

1.1.1. Creating a data science project

To implement a data science workflow, you must create a project. In OpenShift, a project is a Kubernetes namespace with additional annotations, and is the main way that you can manage user access to resources. A project organizes your data science work in one place and also allows you to collaborate with other developers and data scientists in your organization.

Within a project, you can add the following functionality:

- Data connections so that you can access data without having to hardcode information like endpoints or credentials.
- Workbenches for working with and processing data, and for developing models.
- Deployed models so that you can test them and then integrate them into intelligent applications. Deploying a model makes it available as a service that you can access by using an API.
- Pipelines for automating your ML workflow.

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.

Procedure

1. From the OpenShift AI dashboard, select **Data Science Projects**.
2. Click **Create data science project**
3. In the **Create a data science project** dialog, enter a display **Name** for your project.
4. Optional: Edit the **Resource name** for your data science project. The resource name must consist of lowercase alphanumeric characters, -, and must start and end with an alphanumeric character.
Note: After you create a project, you can change the project display name but you cannot change the resource name.
5. Enter a **description** for your data science project.
6. Click **Create**.

Verification

- A project details page opens. From this page, you can add data connections, create workbenches, configure pipelines, and deploy models.


1.1.2. Updating a data science project

You can update your data science project's details by changing your project's name and description text.

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.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the action menu () beside the project whose details you want to update and click **Edit project**.
The **Edit data science project** dialog opens.
3. Optional: Update the **name** for your data science project.
4. Optional: Update the **description** for your data science project.

5. Click **Update**.

Verification

- The data science project that you updated is displayed on the **Data Science Projects** page.

1.1.3. Deleting a data science project

You can delete data science projects so that they do not appear on the OpenShift AI **Data Science Projects** page when you no longer want to use 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, **{oai-user-group}**) in OpenShift.
- You have created a data science project.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the action menu (**⋮**) beside the project that you want to delete and then click **Delete project**.
The **Delete project** dialog opens.
3. Enter the project name in the text field to confirm that you intend to delete it.
4. Click **Delete project**.

Verification

- The data science project that you deleted is no longer displayed on the **Data Science Projects** page.
- Deleting a data science project deletes any associated workbenches, data science pipelines, cluster storage, and data connections. This data is permanently deleted and is not recoverable.

1.2. USING PROJECT WORKBENCHES

1.2.1. Creating a workbench and selecting an IDE

A workbench is an isolated area where you can examine and work with ML models. You can also work with data and run programs, for example to prepare and clean data. While a workbench is not required if, for example, you only want to service an existing model, one is needed for most data science workflow tasks, such as writing code to process data, or training a model.

When you create a workbench, you specify an image (an IDE, packages, and other dependencies). Supported IDEs include JupyterLab, code-server (Technology Preview), and RStudio (Technology Preview).

The IDEs are based on a server–client architecture. Each IDE provides a server that runs in a container on the OpenShift cluster, while the user interface (the client) is displayed in your web browser. For example, the Jupyter notebook server runs in a container on the Red Hat OpenShift cluster. The client is the JupyterLab interface that opens in your web browser on your local computer. All of the commands that you enter in JupyterLab are executed by the notebook server. Similarly, other IDEs like code-server or RStudio Server provide a server that runs in a container on the OpenShift cluster, while the user interface is displayed in your web browser. This architecture allows you to interact through your local computer in a browser environment, while all processing occurs on the cluster. The cluster provides the benefits of larger available resources and security because the data being processed never leaves the cluster.

In a workbench, you also configure data connections (to access external data for training models and to save models so that you can deploy them) and cluster storage (for persisting data). Workbenches within the same project can share models and data through object storage with the data science pipelines and model servers.

For data science projects that require data retention, you can add container storage to the workbench you are creating.

Within a project, you can create multiple workbenches. When to create a new workbench depends on considerations, such as the following:

- The workbench configuration (for example, CPU, RAM, or IDE). If you want to avoid editing the configuration of an existing workbench’s configuration to accommodate a new task, you can create a new workbench instead.
- Separation of tasks or activities. For example, you might want to use one workbench for your Large Language Models (LLM) experimentation activities, another workbench dedicated to a demo, and another workbench for testing.

1.2.1.1. About workbench images

A workbench image (sometimes referred to as a notebook image) is optimized with the tools and libraries that you need for model development. You can use the provided workbench images or an OpenShift AI admin user can create custom workbench images adapted to your needs.

To provide a consistent, stable platform for your model development, many provided workbench images contain the same version of Python. Most workbench images available on OpenShift AI are pre-built and ready for you to use immediately after OpenShift AI is installed or upgraded.

For information about Red Hat support of workbench images and packages, see [Red Hat OpenShift AI: Supported Configurations](#).

Red Hat OpenShift AI contains the following notebook images that are available by default.



IMPORTANT

Notebook images denoted with **(Technology Preview)** in this table are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using Technology Preview features in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

Table 1.1. Default notebook images

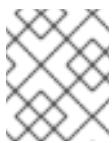


Image name	Description
CUDA	If you are working with compute-intensive data science models that require GPU support, use the Compute Unified Device Architecture (CUDA) notebook image to gain access to the NVIDIA CUDA Toolkit. Using this toolkit, you can optimize your work by using GPU-accelerated libraries and optimization tools.
Standard Data Science	Use the Standard Data Science notebook image for models that do not require TensorFlow or PyTorch. This image contains commonly-used libraries to assist you in developing your machine learning models.
TensorFlow	TensorFlow is an open source platform for machine learning. With TensorFlow, you can build, train and deploy your machine learning models. TensorFlow contains advanced data visualization features, such as computational graph visualizations. It also allows you to easily monitor and track the progress of your models.
PyTorch	PyTorch is an open source machine learning library optimized for deep learning. If you are working with computer vision or natural language processing models, use the Pytorch notebook image.
Minimal Python	If you do not require advanced machine learning features, or additional resources for compute-intensive data science work, you can use the Minimal Python image to develop your models.
TrustyAI	Use the TrustyAI notebook image to leverage your data science work with model explainability, tracing, and accountability, and runtime monitoring.
HabanaAI	The HabanaAI notebook image optimizes high-performance deep learning (DL) with Habana Gaudi devices. Habana Gaudi devices accelerate DL training workloads and maximize training throughput and efficiency.
code-server (Technology Preview)	<p>With the code-server notebook image, you can customize your notebook environment to meet your needs using a variety of extensions to add new languages, themes, debuggers, and connect to additional services. Enhance the efficiency of your data science work with syntax highlighting, auto-indentation, and bracket matching, as well as an automatic task runner for seamless automation. See code-server in GitHub for more information.</p> <div style="display: flex; align-items: flex-start;">  <div> <p>NOTE</p> <p>Elyra-based pipelines are not available with the code-server notebook image.</p> </div> </div>

Image name	Description
RStudio Server (Technology preview)	<p>Use the RStudio Server notebook image to access the RStudio IDE, an integrated development environment for R, a programming language for statistical computing and graphics. See the RStudio Server site for more information.</p> <p>To use the RStudio Server notebook image, you must first build it by creating a secret and triggering the BuildConfig, and then enable it in the OpenShift AI UI by editing the rstudio-rhel9 image stream. See Building the RStudio Server notebook images for more information.</p> <div style="display: flex; align-items: flex-start;">  <div style="margin-left: 10px;"> <p>IMPORTANT</p> <p>Disclaimer: Red Hat supports managing workbenches in OpenShift AI. However, Red Hat does not provide support for the RStudio software. RStudio Server is available through https://rstudio.org/ and is subject to their licensing terms. Review their licensing terms before you use this sample workbench.</p> </div> </div>
CUDA - RStudio Server (Technology preview)	<p>Use the CUDA - RStudio Server notebook image to access the RStudio IDE and NVIDIA CUDA Toolkit. RStudio is an integrated development environment for R, a programming language for statistical computing and graphics. With the NVIDIA CUDA toolkit, you can optimize your work using GPU-accelerated libraries and optimization tools. See the RStudio Server site for more information.</p> <p>To use the CUDA - RStudio Server notebook image, you must first build it by creating a secret and triggering the BuildConfig, and then enable it in the OpenShift AI UI by editing the cuda-rstudio-rhel9 image stream. See Building the RStudio Server notebook images for more information.</p> <div style="display: flex; align-items: flex-start;">  <div style="margin-left: 10px;"> <p>IMPORTANT</p> <p>Disclaimer: Red Hat supports managing workbenches in OpenShift AI. However, Red Hat does not provide support for the RStudio software. RStudio Server is available through https://rstudio.org/ and is subject to their licensing terms. Review their licensing terms before you use this sample workbench.</p> <p>The CUDA - RStudio Server notebook image contains NVIDIA CUDA technology. CUDA licensing information is available at https://docs.nvidia.com/cuda/. Review their licensing terms before you use this sample workbench.</p> </div> </div>

1.2.1.2. Creating a workbench

When you create a workbench, you specify an image (an IDE, packages, and other dependencies). You can also configure data connections, cluster storage, and add container storage.

Prerequisites

- You have logged in to Red Hat OpenShift AI.

- If you use specialized OpenShift AI groups, you are part of the user group or admin group (for example, **rhoai-users** or **rhoai-admins**) in OpenShift.
- You created a project.
- If you created a Simple Storage Service (S3) account outside of Red Hat OpenShift AI and you want to create data connections to your existing S3 storage buckets, you have the following credential information for the storage buckets:
 - Endpoint URL
 - Access key
 - Secret key
 - Region
 - Bucket name

For information about working with data stored in AWS S3, see [Integrating data from Amazon S3](#).

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
2. Click the name of the project that you want to add the workbench to. A **Details** page for the project opens.
3. In the **Workbenches** section, click **Create a workbench**
4. In the **Create workbench** page, configure the properties of the workbench that you are creating.
 - a. In the **Name** field, enter a name for your workbench.
 - b. Optional: In the **Description** field, enter a description to define your workbench.
 - c. In the **Notebook image** section, complete the fields to specify the workbench image to use with your workbench.

From the **Image selection** list, select a workbench image that suits your use case. A workbench image includes an IDE and Python packages (reusable code). Optionally, click the **View package information** option to view a list of packages that are included in the image that you selected.

If the workbench image has multiple versions available, select the workbench image version to use from the **Versions** section. To use the latest package versions, Red Hat recommends that you use the most recently added image.



NOTE

You can change the workbench image after you create the workbench.

- d. In the **Deployment size** section, from the **Container size** list, select a container size for your server. The container size controls the number of CPUs, the amount of memory, and the minimum and maximum request capacity of the container.

- e. Optional: Select and specify values for any environment variables. Setting environment variables during the workbench configuration helps you save time later because you do not need to define them in the body of your notebooks, or with the IDE command line interface.

If you are using S3-compatible storage, add these recommended environment variables:

- **AWS_ACCESS_KEY_ID** specifies your Access Key ID for Amazon Web Services.
- **AWS_SECRET_ACCESS_KEY** specifies your Secret access key for the account specified in **AWS_ACCESS_KEY_ID**.

OpenShift AI stores the credentials as Kubernetes secrets in a protected namespace if you select **Secret** when you add the variable.

- f. Configure the storage for your workbench. Select one of the following options:

- **Create new persistent storage** to create storage that is retained after you shut down your workbench. Complete the relevant fields to define the storage.
- **Use existing persistent storage** to reuse existing storage and select the storage from the **Persistent storage** list.

- g. Optionally, you can add a data connection to your workbench. A data connection is a resource that contains the configuration parameters needed to connect to a data source or an object storage bucket. Currently, only S3-Compatible data connections are supported. You can use storage buckets for storing data, models, and pipeline artifacts. You can also use a data connection to specify the location of a model that you want to deploy. In the **Data connections** section, select the **Use a data connection** checkbox.

- Create a new data connection as follows:
 - i. Select **Create new data connection**
 - ii. In the **Name** field, enter a unique name for the data connection.
 - iii. In the **Access key** field, enter the access key ID for the S3-compatible object storage provider.
 - iv. In the **Secret key** field, enter the secret access key for the S3-compatible object storage account that you specified.
 - v. In the **Endpoint** field, enter the endpoint of your S3-compatible object storage bucket.
 - vi. In the **Region** field, enter the default region of your S3-compatible object storage account.
 - vii. In the **Bucket** field, enter the name of your S3-compatible object storage bucket.
- Use an existing data connection as follows:
 - i. Select **Use existing data connection**
 - ii. From the **Data connection** list, select a data connection that you previously defined.

5. Click **Create workbench**.

Verification

- The workbench that you created appears on the **Workbenches** tab for the project.
- Any cluster storage that you associated with the workbench during the creation process appears on the **Cluster storage** tab for the project.
- The **Status** column on the **Workbenches** tab displays a status of **Starting** when the workbench server is starting, and **Running** when the workbench has successfully started.
- Optionally, click the **Open** link to open the IDE in a new window.

1.2.2. Starting a workbench

You can manually start a data science project's workbench from the **Workbenches** tab on the project details page. By default, workbenches start immediately after you create 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 created a data science project that contains a workbench.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the name of the project whose workbench you want to start.
A project details page opens.
3. Click the **Workbenches** tab.
4. Click the toggle in the **Status** column for the relevant workbench to start a workbench that is not running.
The status of the workbench that you started changes from **Stopped** to **Running**. After the workbench has started, click **Open** to open the workbench's notebook.

Verification

- The workbench that you started appears on the **Workbenches** tab for the project, with the status of **Running**.


1.2.3. Updating a project workbench

If your data science work requires you to change your workbench's notebook image, container size, or identifying information, you can update the properties of your project's workbench. If you require extra power for use with large datasets, you can assign accelerators to your workbench to optimize performance.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- If you use 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 has a workbench.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the name of the project whose workbench you want to update.
A project details page opens.
3. Click the **Workbenches** tab.
4. Click the action menu () beside the workbench that you want to update and then click **Edit workbench**.
The **Edit workbench** page opens.
5. Update any of the workbench properties and then click **Update workbench**.

Verification

- The workbench that you updated appears on the **Workbenches** tab for the project.


1.2.4. Deleting a workbench from a data science project

You can delete workbenches from your data science projects to help you remove Jupyter notebooks that are no longer relevant to your work.

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 with a workbench.

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 delete the workbench from.
A project details page opens.
3. Click the **Workbenches** tab.
4. Click the action menu () beside the workbench that you want to delete and then click **Delete workbench**.
The **Delete workbench** dialog opens.

5. Enter the name of the workbench in the text field to confirm that you intend to delete it.
6. Click **Delete workbench**.

Verification

- The workbench that you deleted is no longer displayed in the **Workbenches** tab for the project.
- The custom resource (CR) associated with the workbench's Jupyter notebook is deleted.

1.3. USING DATA CONNECTIONS

1.3.1. Adding a data connection to your data science project

You can enhance your data science project by adding a connection to a data source. When you want to work with a very large data sets, you can store your data in an S3-compatible object storage bucket, so that you do not fill up your local storage. You also have the option of associating the data connection with an existing workbench that does not already have a connection.

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 data connection to.
- You have access to S3-compatible object storage.
- If you intend to add the data connection to an existing workbench, you have saved any data in the workbench to avoid losing work.

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 add a data connection to.
A project details page opens.
3. Click the **Data connections** tab.
4. Click **Add data connection**.
The **Add data connection** dialog opens.
5. Enter a **name** for the data connection.
6. In the **Access key** field, enter the access key ID for your S3-compatible object storage provider.
7. In the **Secret key** field, enter the secret access key for the S3-compatible object storage account you specified.
8. In the **Endpoint** field, enter the endpoint of your S3-compatible object storage bucket.

9. In the **Region** field, enter the default region of your S3-compatible object storage account.
10. In the **Bucket** field, enter the name of your S3-compatible object storage bucket.
11. Optional: From the **Connected workbench** list, select a workbench to connect.
12. Click **Add data connection**

Verification

- The data connection that you added appears in the **Data connections** tab for the project.
- If you selected a workbench, the workbench is visible in the **Connected workbenches** column in the **Data connections** tab for the project.

1.3.2. Deleting a data connection

You can delete data connections from your data science projects to help you remove connections that are no longer relevant to your work.

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 with a data connection.

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 delete the data connection from.
A project details page opens.
3. Click the **Data connections** tab.
4. Click the action menu (**⋮**) beside the data connection that you want to delete and then click **Delete data connection**.
The **Delete data connection** dialog opens.
5. Enter the name of the data connection in the text field to confirm that you intend to delete it.
6. Click **Delete data connection**

Verification

- The data connection that you deleted is no longer displayed in the **Data connections** tab for the project.


1.3.3. Updating a connected data source

To use an existing data source with a different workbench, you can change the data source that is connected to your project's workbench.

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, created a workbench, and you have defined a data connection.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the name of the project whose data source you want to change.
A project details page opens.
3. Click the **Data connections** tab.
4. Click the action menu () beside the data source that you want to change and then click **Edit data connection**.
The **Edit data connection** dialog opens.
5. In the **Connected workbench** section, select an existing workbench from the list.
6. Click **Update data connection**

Verification

- The updated data connection is displayed in the **Data connections** tab for the project.
- You can access your S3 data source using environment variables in the connected workbench.

1.4. CONFIGURING CLUSTER STORAGE

1.4.1. Adding cluster storage to your data science project

For data science projects that require data to be retained, you can add cluster storage to the project. Additionally, you can also connect cluster storage to a specific project's workbench.

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 cluster storage to.

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 add the cluster storage to.
A project details page opens.
3. Click the **Cluster storage** tab.
4. Click **Add cluster storage**.
The **Add storage** dialog opens.
5. Enter a **name** for the cluster storage.
6. Enter a **description** for the cluster storage.
7. Under **Persistent storage size**, enter a new size in gibibytes. The minimum size is 1 GiB, and the maximum size is 16384 GiB.
8. Optional: Select a **workbench** from the list to connect the cluster storage to an existing workbench.
9. If you selected a workbench to connect the storage to, enter the storage directory in the **Mount folder** field.
10. Click **Add storage**.

Verification

- The cluster storage that you added appears in the **Cluster storage** tab for the project.
- A new persistent volume claim (PVC) is created with the storage size that you defined.
- The persistent volume claim (PVC) is visible as an attached storage in the **Workbenches** tab for the project.

1.4.2. Updating cluster storage


If your data science work requires you to change the identifying information of a project's cluster storage or the workbench that the storage is connected to, you can update your project's cluster storage to change these properties.

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 cluster storage.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the name of the project whose storage you want to update.

- A project details page opens.
3. Click the **Cluster storage** tab.
 4. Click the action menu () beside the storage that you want to update and then click **Edit storage**.
The **Edit storage** page opens.
 5. Update the storage's properties.
 - a. Update the **name** for the storage, if applicable.
 - b. Update the **description** for the storage, if applicable.
 - c. Increase the **Persistent storage size** for the storage, if applicable.
Note that you can only increase the storage size. Updating the storage size restarts the workbench and makes it unavailable for a period of time that is usually proportional to the size change.
 - d. Update the **workbench** that the storage is connected to, if applicable.
 - e. If you selected a new workbench to connect the storage to, enter the storage directory in the **Mount folder** field.
 6. Click **Update storage**.

If you increased the storage size, the workbench restarts and is unavailable for a period of time that is usually proportional to the size change.

Verification

- The storage that you updated appears in the **Cluster storage** tab for the project.

1.4.3. Deleting cluster storage from a data science project


You can delete cluster storage from your data science projects to help you free up resources and delete unwanted storage space.

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 with cluster storage.

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 delete the storage from.
A project details page opens.
3. Click the **Cluster storage** tab.

4. Click the action menu () beside the storage that you want to delete and then click **Delete storage**.
The **Delete storage** dialog opens.
5. Enter the name of the storage in the text field to confirm that you intend to delete it.
6. Click **Delete storage**.

Verification

- The storage that you deleted is no longer displayed in the **Cluster storage** tab for the project.
- The persistent volume (PV) and persistent volume claim (PVC) associated with the cluster storage are both permanently deleted. This data is not recoverable.

1.5. CONFIGURING ACCESS TO DATA SCIENCE PROJECTS

1.5.1. Configuring access to data science projects

To enable you to work collaboratively on your data science projects with other users, you can share access to your project. After creating your project, you can then set the appropriate access permissions from the OpenShift AI user interface.

You can assign the following access permission levels to your data science projects:

- **Admin** - Users can modify all areas of a project, including its details (project name and description), components, and access permissions.
- **Edit** - Users can modify a project's components, such as its workbench, but they cannot edit a project's access permissions or its details (project name and description).

1.5.2. Sharing access to a data science project


To enable your organization to work collaboratively, you can share access to your data science project with other users and groups.

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.


Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. From the list of data science projects, click the name of the data science project that you want to share access to.
A project details page opens.

3. Click the **Permissions** tab.
The **Permissions** page for the project opens.
4. Provide one or more users with access to the project.
 - a. In the **Users** section, click **Add user**.
 - b. In the **Name** field, enter the user name of the user whom you want to provide access to the project.
 - c. From the **Permissions** list, select one of the following access permission levels:
 - Admin: Users with this access level can edit project details and manage access to the project.
 - Edit: Users with this access level can view and edit project components, such as its workbenches, data connections, and storage.
 - d. To confirm your entry, click **Confirm** ().
 - e. Optional: To add an additional user, click **Add user** and repeat the process.
5. Provide one or more OpenShift groups with access to the project.
 - a. In the **Groups** section, click **Add group**.
 - b. From the **Name** list, select a group to provide access to the project.

**NOTE**

If you do not have **cluster-admin** permissions, the **Name** list is not visible. Instead, an input field is displayed enabling you to configure group permissions.

- c. From the **Permissions** list, select one of the following access permission levels:
 - Admin: Groups with this access permission level can edit project details and manage access to the project.
 - Edit: Groups with this access permission level can view and edit project components, such as its workbenches, data connections, and storage.
- d. To confirm your entry, click **Confirm** ().
- e. Optional: To add an additional group, click **Add group** and repeat the process.

Verification

- Users to whom you provided access to the project can perform only the actions permitted by their access permission level.
- The **Users** and **Groups** sections on the **Permissions** tab show the respective users and groups that you provided with access to the project.


1.5.3. Updating access to a data science project

To change the level of collaboration on your data science project, you can update the access permissions of users and groups who have access to your project.

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 previously shared access to your project with other users or groups.
- You have administrator permissions or you are the project owner.

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 change the access permissions of.
A project details page opens.
3. Click the **Permissions** tab.
The **Permissions** page for the project opens.
4. Update the user access permissions to the project.
 - a. In the **Name** field, update the user name of the user whom you want to provide access to the project.
 - b. From the **Permissions** list, update the user access permissions by selecting one of the following:
 - **Admin**: Users with this access level can edit project details and manage access to the project.
 - **Edit**: Users with this access level can view and edit project components, such as its workbenches, data connections, and storage.
 - c. To confirm the update to the entry, click **Confirm** ().
5. Update the OpenShift groups access permissions to the project.
 - a. From the **Name** list, update the group that has access to the project by selecting another group from the list.




NOTE

If you do not have **cluster-admin** permissions, the **Name** list is not visible. Instead, you can configure group permissions in the input field that appears.

- b. From the **Permissions** list, update the group access permissions by selecting one of the following:

- **Admin:** Groups with this access permission level can edit project details and manage access to the project.
- **Edit:** Groups with this access permission level can view and edit project components, such as its workbenches, data connections, and storage.

c. To confirm the update to the entry, click **Confirm** ().

Verification

- The **Users** and **Groups** sections on the **Permissions** tab show the respective users and groups whose project access permissions you changed.


1.5.4. Removing access to a data science project

If you no longer want to work collaboratively on your data science project, you can restrict access to your project by removing users and groups that you previously provided access to your project.

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 previously shared access to your project with other users or groups.
- You have administrator permissions or you are the project owner.

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 change the access permissions of.
A project details page opens.
3. Click the **Permissions** tab.
The **Permissions** page for the project opens.
4. Click the action menu () beside the user or group whose access permissions you want to revoke and click **Delete**.

Verification

- Users whose access you have revoked can no longer perform the actions that were permitted by their access permission level.

CHAPTER 2. WORKING IN YOUR DATA SCIENCE IDE

In Red Hat OpenShift AI, when you create a workbench, you select a workbench image that includes an integrated development environment (IDE) for developing your machine learning models.

OpenShift AI supports the following data science IDEs for developing ML models:

- JupyterLab
- code-server (Technology Preview feature)
- RStudio Server (Technology Preview feature)



NOTE

The RStudio workbench images are currently unavailable for disconnected environments.

2.1. ACCESSING YOUR WORKBENCH IDE

To access a workbench IDE, use the link provided in the OpenShift AI interface.

Prerequisite

- You have created a data science project and a workbench.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
2. Click the name of the project that contains the workbench.
3. If the status of the workbench is **Running**, skip to Step 4. Otherwise, click the toggle in the **Status** column for the relevant workbench and wait until the status changes from **Stopped** to **Running**.
4. Click **Open**.

Verification

- A new browser window opens for the workbench IDE.

2.2. WORKING IN JUPYTERLAB

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. You can configure and arrange workflows in data science and machine learning. JupyterLab is an open-source web application that supports over 40 programming languages, including Python and R.

2.2.1. Creating and importing notebooks

You can create a blank notebook or import a notebook from several different sources.

2.2.1.1. Creating a new notebook

You can create a new Jupyter notebook from an existing notebook container image to access its resources and properties. The **Notebook server control panel** contains a list of available container images that you can run as a single-user notebook server.

Prerequisites

- Ensure that you have logged in to Red Hat OpenShift AI.
- Ensure that you have launched your notebook server and logged in to JupyterLab.
- The notebook image exists in a registry, image stream, and is accessible.

Procedure

1. Click **File** → **New** → **Notebook**.
2. If prompted, select a kernel for your notebook from the list.
If you want to use a kernel, click **Select**. If you do not want to use a kernel, click **No Kernel**.

Verification

- Check that the notebook file is visible in the JupyterLab interface.


2.2.1.2. Uploading an existing notebook file from local storage

You can load an existing notebook from local storage into JupyterLab to continue work, or adapt a project for a new use case.

Prerequisites

- Credentials for logging in to JupyterLab.
- A launched and running Jupyter notebook server.
- A notebook file exists in your local storage.

Procedure

1. In the **File Browser** in the left sidebar of the JupyterLab interface, click **Upload Files** ().
2. Locate and select the notebook file and click **Open**.
The file is displayed in the **File Browser**.

Verification

- The notebook file displays in the **File Browser** in the left sidebar of the JupyterLab interface.
- You can open the notebook file in JupyterLab.

2.2.2. Collaborating on notebooks by using Git

If your notebooks or other files are stored in Git version control, you can import them from a Git repository onto your notebook server to work with them in JupyterLab. When you are ready, you can push your changes back to the Git repository so that others can review or use your models.

2.2.3. Uploading an existing notebook file from a Git repository by using JupyterLab

You can use the JupyterLab user interface to clone a Git repository into your workspace to continue your work or integrate files from an external project.


Prerequisites

- A launched and running Jupyter notebook server.
- Read access for the Git repository you want to clone.

Procedure

1. Copy the HTTPS URL for the Git repository.
 - On GitHub, click **Code** → **HTTPS** and click the Clipboard button.
 - On GitLab, click **Clone** and click the Clipboard button under **Clone with HTTPS**.

2. In the JupyterLab interface, click the **Git Clone** button ().

You can also click **Git** → **Clone a repository** in the menu, or click the Git icon () and click the **Clone a repository** button.

The *Clone a repo* dialog appears.

3. Enter the HTTPS URL of the repository that contains your notebook.
4. Click **CLONE**.
5. If prompted, enter your username and password for the Git repository.

Verification

- Check that the contents of the repository are visible in the file browser in JupyterLab, or run the **ls** command in the terminal to verify that the repository is shown as a directory.

2.2.3.1. Uploading an existing notebook file from a Git repository by using the command line interface

You can use the command line interface to clone a Git repository into your workspace to continue your work or integrate files from an external project.

Prerequisites

- A launched and running Jupyter notebook server.

Procedure

1. Copy the HTTPS URL for the Git repository.
 - On GitHub, click **Code** → **HTTPS** and click the Clipboard button.
 - On GitLab, click **Clone** and click the Clipboard button under **Clone with HTTPS**.

2. In JupyterLab, click **File** → **New** → **Terminal** to open a terminal window.
3. Enter the **git clone** command.

```
git clone <git-clone-URL>
```

Replace `*git-clone-URL*` with the HTTPS URL, for example:

```
[1234567890@jupyter-nb-jdoe ~]$ git clone https://github.com/example/myrepo.git
Cloning into myrepo...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 2821 (delta 1), reused 5 (delta 1), pack-reused 2810
Receiving objects: 100% (2821/2821), 39.17 MiB | 23.89 MiB/s, done.
Resolving deltas: 100% (1416/1416), done.
```

Verification

- Check that the contents of the repository are visible in the file browser in JupyterLab, or run the **ls** command in the terminal to verify that the repository is shown as a directory.



2.2.3.2. Updating your project with changes from a remote Git repository

You can pull changes made by other users into your data science project from a remote Git repository.

Prerequisites

- You have configured the remote Git repository.
- You have already imported the Git repository into JupyterLab, and the contents of the repository are visible in the file browser in JupyterLab.
- You have permissions to pull files from the remote Git repository to your local repository.
- You have credentials for logging in to Jupyter.
- You have a launched and running Jupyter server.

Procedure

1. In the JupyterLab interface, click the **Git** button ().
2. Click the **Pull latest changes** button ().

Verification

- You can view the changes pulled from the remote repository in the **History** tab of the Git pane.


2.2.3.3. Pushing project changes to a Git repository

To build and deploy your application in a production environment, upload your work to a remote Git repository.

Prerequisites

- You have opened a notebook in the JupyterLab interface.
- You have already added the relevant Git repository to your notebook server.
- You have permission to push changes to the relevant Git repository.
- You have installed the Git version control extension.

Procedure

1. Click **File** → **Save All** to save any unsaved changes.
2. Click the Git icon () to open the Git pane in the JupyterLab interface.
3. Confirm that your changed files appear under **Changed**.
If your changed files appear under **Untracked**, click **Git** → **Simple Staging** to enable a simplified Git process.
4. Commit your changes.
 - a. Ensure that all files under **Changed** have a blue checkmark beside them.
 - b. In the **Summary** field, enter a brief description of the changes you made.
 - c. Click **Commit**.
5. Click **Git** → **Push to Remote** to push your changes to the remote repository.
6. When prompted, enter your Git credentials and click **OK**.

Verification

- Your most recently pushed changes are visible in the remote Git repository.

2.2.4. Managing Python packages

In JupyterLab, you can view the Python packages that are installed on your notebook image and install additional packages.

2.2.4.1. Viewing Python packages installed on your notebook server

You can check which Python packages are installed on your notebook server and which version of the package you have by running the **pip** tool in a notebook cell.

Prerequisites

- Log in to JupyterLab and open a notebook.

Procedure

1. Enter the following in a new cell in your notebook:

```
!pip list
```

2. Run the cell.

Verification

- The output shows an alphabetical list of all installed Python packages and their versions. For example, if you use this command immediately after creating a notebook server that uses the **Minimal** image, the first packages shown are similar to the following:

Package	Version

aiohttp	3.7.3
alembic	1.5.2
appdirs	1.4.4
argo-workflows	3.6.1
argon2-cffi	20.1.0
async-generator	1.10
async-timeout	3.0.1
attrdict	2.0.1
attrs	20.3.0
backcall	0.2.0

Additional resources

- [Installing Python packages on your notebook server](#)

2.2.4.2. Installing Python packages on your notebook server

You can install Python packages that are not part of the default notebook server by adding the package and the version to a **requirements.txt** file and then running the **pip install** command in a notebook cell.



NOTE

You can also install packages directly, but Red Hat recommends using a **requirements.txt** file so that the packages stated in the file can be easily re-used across different notebooks. In addition, using a **requirements.txt** file is also useful when using a S2I build to deploy a model.

Prerequisites

- Log in to JupyterLab and open a notebook.

Procedure

1. Create a new text file using one of the following methods:
 - Click **+** to open a new launcher and click **Text file**.
 - Click **File** → **New** → **Text File**.
2. Rename the text file to **requirements.txt**.

- a. Right-click on the name of the file and click **Rename Text**. The **Rename File** dialog opens.
 - b. Enter **requirements.txt** in the **New Name** field and click **Rename**.
3. Add the packages to install to the **requirements.txt** file.

```
altair
```

You can specify the exact version to install by using the **==** (equal to) operator, for example:

```
altair==4.1.0
```



NOTE

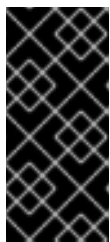
Red Hat recommends specifying exact package versions to enhance the stability of your notebook server over time. New package versions can introduce undesirable or unexpected changes in your environment's behavior.

To install multiple packages at the same time, place each package on a separate line.

4. Install the packages in **requirements.txt** to your server using a notebook cell.
 - a. Create a new cell in your notebook and enter the following command:

```
!pip install -r requirements.txt
```

- b. Run the cell by pressing Shift and Enter.



IMPORTANT

This command installs the package on your notebook server, but you must still run the **import** directive in a code cell to use the package in your code.

```
import altair
```

Verification

- Confirm that the packages in **requirements.txt** appear in the list of packages installed on the notebook server. See [Viewing Python packages installed on your notebook server](#) for details.

2.2.5. Troubleshooting common problems in Jupyter for users

If you are seeing errors in Red Hat OpenShift AI related to Jupyter, your notebooks, or your notebook server, read this section to understand what could be causing the problem.

If you cannot see your problem here or in the release notes, contact Red Hat Support.

2.2.5.1. I see a 403: Forbidden error when I log in to Jupyter

Problem

If your administrator has configured specialized user groups for OpenShift AI, your user name might not be added to the default user group or the default administrator group for OpenShift AI.

Resolution

Contact your administrator so that they can add you to the correct group/s.

2.2.5.2. My notebook server does not start

Problem

The OpenShift Container Platform cluster that hosts your notebook server might not have access to enough resources, or the Jupyter pod may have failed.

Resolution

Check the logs in the **Events** section in OpenShift for error messages associated with the problem. For example:

```
Server requested
2021-10-28T13:31:29.830991Z [Warning] 0/7 nodes are available: 2 Insufficient memory,
2 node(s) had taint {node-role.kubernetes.io/infra: }, that the pod didn't tolerate, 3 node(s) had taint
{node-role.kubernetes.io/master: },
that the pod didn't tolerate.
```

Contact your administrator with details of any relevant error messages so that they can perform further checks.

2.2.5.3. I see a database or disk is full error or a no space left on device error when I run my notebook cells

Problem

You might have run out of storage space on your notebook server.

Resolution

Contact your administrator so that they can perform further checks.

2.3. WORKING IN CODE-SERVER

Red Hat OpenShift AI includes the code-server workbench image.

For more information on code-server, see [code-server in GitHub](#).



NOTE

Elyra-based pipelines are not available with the code-server workbench image.



IMPORTANT

The code-server notebook image is currently available in Red Hat OpenShift AI 2.10 as a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

2.3.1. Installing extensions with code-server

With the code-server workbench image, you can customize your code-server environment by using extensions to add new languages, themes, and debuggers, and to connect to additional services. You can also enhance the efficiency of your data science work with extensions for syntax highlighting, auto-indentation, and bracket matching.

For details about the third-party extensions that you can install with code-server, see the [Open VSX Registry](#).

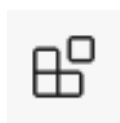
Prerequisites

- You are logged in to Red Hat OpenShift AI.
- If you use 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 has a code-server workbench.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Projects**.
The **Data Science Projects** page opens.
2. Click the name of the project containing the code-server workbench you want to start.
A project details page opens.
3. Click the **Workbenches** tab.
4. Click the toggle in the **Status** column for the relevant workbench to start a workbench that is not running.
The status of the workbench that you started changes from **Stopped** to **Running**.
5. After the workbench has started, click **Open** to open the workbench notebook.



6. In the Activity Bar, click the **Extensions** icon ().
7. Search for the name of the extension you want to install.
8. Click **Install** to add the extension to your code-server environment.

Verification

Verification

- In the **Browser - Installed** list on the **Extensions** panel, you see the extension that you installed.

CHAPTER 3. WORKING WITH DATA SCIENCE PIPELINES

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 run: An execution of your pipeline.
 - Active run: A pipeline run that is in its execution phase, or is stopped.
 - Scheduled run: A pipeline run scheduled to execute at least once.
 - Archived run: A pipeline run that resides in the run archive and is no longer required.

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.

3.1. ENABLING DATA SCIENCE PIPELINES 2.0

From OpenShift AI version 2.9, data science pipelines are based on [KubeFlow Pipelines \(KFP\) version 2.0](#). DSP 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 (DSP) 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 with DSP, 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'
```

3.1.1. Installing OpenShift AI with DSP 2.0

To install OpenShift AI 2.9, ensure that there is no installation of Argo Workflows that is not installed by DSP 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 DSP on your cluster, DSP 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.

3.1.2. Upgrading to DSP 2.0**IMPORTANT**

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

To upgrade to DSP 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 DSP enabled, and there is an existing installation of Argo Workflows that is not installed by DSP on your cluster, OpenShift AI components will not be upgraded. To complete the component upgrade, disable DSP 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 DSP 1.0 to 2.0. For more information, see [Migrating pipelines from DSP 1.0 to 2.0](#).

3.1.3. Migrating pipelines from DSP 1.0 to 2.0

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

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

DSP 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 DSP 2.0-based data science project.
5. (Optional) Remove your DSP 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 DSP 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](#).

3.1.4. Accessing DSP 1.0 pipelines and history

You can view historical DSP 1.0 pipeline run information in the OpenShift Container Platform 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](#).

3.1.5. Uninstalling the OpenShift Pipelines Operator

When your migration to DSP 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 DSP 1.0 pipelines to 2.0 is complete, and that there are no remaining DSP 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](#)

3.2. MANAGING DATA SCIENCE PIPELINES

3.2.1. 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, the database is migrated to DSP 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.

3.2.2. 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](#)

3.2.3. 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.


3.2.4. Downloading a data science pipeline

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

Prerequisites

- You have installed the OpenShift Pipelines operator.
- 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**.
The **Pipelines** page opens.
2. From the **Project** list, select the project whose pipeline that you want to download.
3. In the **Pipeline name** column, click the name of the pipeline that you want to download.
The **Pipeline details** page opens displaying the **Graph** tab.
4. Click the **YAML** tab.
The page reloads to display an embedded YAML editor showing the pipeline code.
5. Click the **Download** button () to download the YAML file containing your pipeline's code to your local machine.

Verification

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

3.2.5. 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.

3.2.6. 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.

3.2.7. 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.


3.2.8. 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.

3.2.9. 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.

3.2.10. 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.

3.2.11. 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.

3.2.12. Viewing pipeline versions


You can view all versions for a pipeline on the **Pipelines** page.

Prerequisites

- You have installed the OpenShift Pipelines operator.
- 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.
3. Click **Expand** () on the row containing the pipeline that you want to view versions for.

Verification

- You can view the versions of the pipeline on the **Pipelines** page.


3.2.13. 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.

3.3. MANAGING PIPELINE RUNS

3.3.1. 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. You can clone your pipeline runs to reproduce and scale them accordingly, 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 run:** A pipeline run scheduled to execute at least once.
- **Active run:** A pipeline run that is in its execution phase, or is stopped.
- **Archived run:** A pipeline run that resides in the run archive and is no longer required.

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.

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.

3.3.2. 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 3.1. Pipeline configuration file and artifacts storage locations

Pipeline run source	Pipeline storage directory	Run artifacts storage directory
---------------------	----------------------------	---------------------------------

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

3.3.3. Viewing active pipeline runs

You can view a list of pipeline runs that were previously executed in OpenShift AI. 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.

Procedure

1. From the OpenShift AI dashboard, click **Data Science Pipelines** → **Runs**.
2. On the **Runs** page, select the **project** for the active pipeline runs that you want to view.
3. On the **Run details** page, click the **Active** 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 **Active** tab on the **Runs** page.

3.3.4. Executing a pipeline run

You can instantiate a single execution of a pipeline by creating an active pipeline run that executes immediately after creation.

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 **Data Science Pipelines** → **Runs**.
2. On the **Runs** page, select the **project** that you want to create a run for.
3. Click **Create run**.
4. On the **Create run** page, configure the run:
 - a. In the **Name** field, enter a name for the run.
 - b. In the **Description** field, enter a description for the run.
 - c. 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 complete the relevant fields in the **Import pipeline** dialog.
 - d. 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 complete the relevant fields in the **Upload new version** dialog.
 - e. Configure the input parameters for the run by selecting the parameters from the list.
 - f. Click **Create run**.

Verification

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


3.3.5. Stopping an active pipeline run

If you no longer require an active pipeline run to continue executing, 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 **Data Science Pipelines** → **Runs**.
2. On the **Runs** page, select the **project** that contains the pipeline whose active run you want to stop.
3. In the **Active** tab, click the action menu () beside the active run that you want to delete and 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".


3.3.6. Duplicating an active pipeline run

To make it easier to quickly execute pipeline runs with the same configuration, 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 **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 action menu () beside the relevant active run and click **Duplicate**.
4. In 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.

3.3.7. Viewing scheduled pipeline runs

You can view a list of pipeline runs that are scheduled for execution in OpenShift AI. 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

- 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 **Data Science Pipelines → Runs**.
The **Runs** page opens.
2. From the **Project** list, select the project whose scheduled pipeline runs you want to view.
3. Click the **Schedules** tab.
4. 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.

3.3.8. 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 * * * *
On Monday and Tuesday at 15:40 UTC	0 40 15 * * MON,TUE

Additional resources

- [Cron Expression Format](#)

3.3.9. 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 **Data Science Pipelines** → **Runs**. The **Runs** page opens.
2. From the **Project** list, select the project that you want to schedule a run for.
3. Click the **Schedules** tab.
4. Click **Schedule run**.
5. On the **Schedule run** page, configure the run that you are scheduling:
 - a. In the **Name** field, enter a name for the run.

- b. In the **Description** field, enter a description for the run.
- c. 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. 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](#).
- d. In the **Maximum concurrent runs** field, specify the number of runs that can execute concurrently, from a range of one to ten.
- e. 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.
- f. 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.
- g. 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.
- h. 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 complete the relevant fields in the **Import pipeline** dialog.
- i. 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 complete the relevant fields in the **Upload new version** dialog.
- j. Configure the input parameters for the run by selecting the parameters from the list.
- k. Click **Schedule run**.

Verification

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

3.3.10. Duplicating a scheduled pipeline run



To make it easier to schedule runs to execute as part of your pipeline configuration, 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 **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 **Schedules** tab.
4. Click the action menu () beside the run that you want to duplicate and click **Duplicate**.
5. On the **Duplicate schedule** 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 **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. 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](#).
 - d. For **Maximum concurrent runs**, specify the number of runs that can execute concurrently, from a range of one to ten.
 - e. 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.
 - f. 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.
 - g. 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.
 - h. 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 complete the relevant fields in the **Import pipeline** dialog.
 - i. 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 complete the relevant fields in the **Upload new version** dialog.
 - j. Configure the input parameters for the run by selecting the parameters from the list.
 - k. Click **Schedule run**.

Verification

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

3.3.11. 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 **Runs** 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 **Data Science Pipelines** → **Runs**.
The **Runs** page opens.
2. From the **Project** list, select the project that contains the pipeline whose scheduled run you want to delete.
The page refreshes to show the pipeline's scheduled runs on the **Schedules** tab.
3. Click the action menu (**⋮**) beside the scheduled run that you want to delete and click **Delete**.
The **Delete schedule** dialog opens.
4. Enter the run's name in the text field to confirm that you intend to delete it.
5. Click **Delete**.

Verification

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

3.3.12. 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.

3.3.13. 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 **Data Science Pipelines → Runs**.
2. On the **Runs** page, select the **project** for the archived pipeline runs that you want to view.
3. Click the **Archived** tab.

Verification

- A list of archived runs appears in the **Archived** tab on the **Runs** page.

3.3.14. 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 **Data Science Pipelines → Runs**.
2. On the **Runs** page, select the project for the pipeline run that you want to archive from the **Project** list.
3. On the **Run details** page, click the action menu (**⋮**) beside the run that you want to archive and then click **Archive**.
4. In the **Archiving run** dialog, enter the run name in the text field to confirm that you intend to archive it.
5. Click **Archive**.

Verification

- The archived run does not appear in the **Active** tab and instead appears in the **Archived** tab on the **Runs** page.

3.3.15. 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 **Data Science Pipelines → Runs**.
2. On the **Runs** page, select the **project** for the archived pipeline run that you want to restore.

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

Verification

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


3.3.16. 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.

3.3.17. 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.

3.4. WORKING WITH PIPELINE LOGS

3.4.1. 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).

3.4.2. 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.

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.



3.4.3. 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 stop 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.

3.5. WORKING WITH PIPELINES IN JUPYTERLAB

3.5.1. 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](#).

3.5.2. 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.

3.5.3. 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** 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.



3.5.4. 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.



3.5.5. 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.

3.5.6. 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


- 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.


3.5.7. 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.

3.5.8. 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 a 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** as 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.

3.6. ADDITIONAL RESOURCES

- [Kubeflow Pipelines 2.0 Documentation](#)
- [Working with pipelines in JupyterLab](#).