Red Hat OpenShift AI Self-Managed 2.10

Getting started with Red Hat OpenShift AI Self-Managed

Learn how to work in an OpenShift AI environment

Last Updated: 2024-06-27

Red Hat OpenShift AI Self-Managed 2.10 Getting started with Red Hat OpenShift AI Self-Managed

Learn how to work in an OpenShift AI environment

Legal Notice

Abstract

Learn how to work in an OpenShift AI environment.

Table of Contents

CHAPTER 1. OVERVIEW

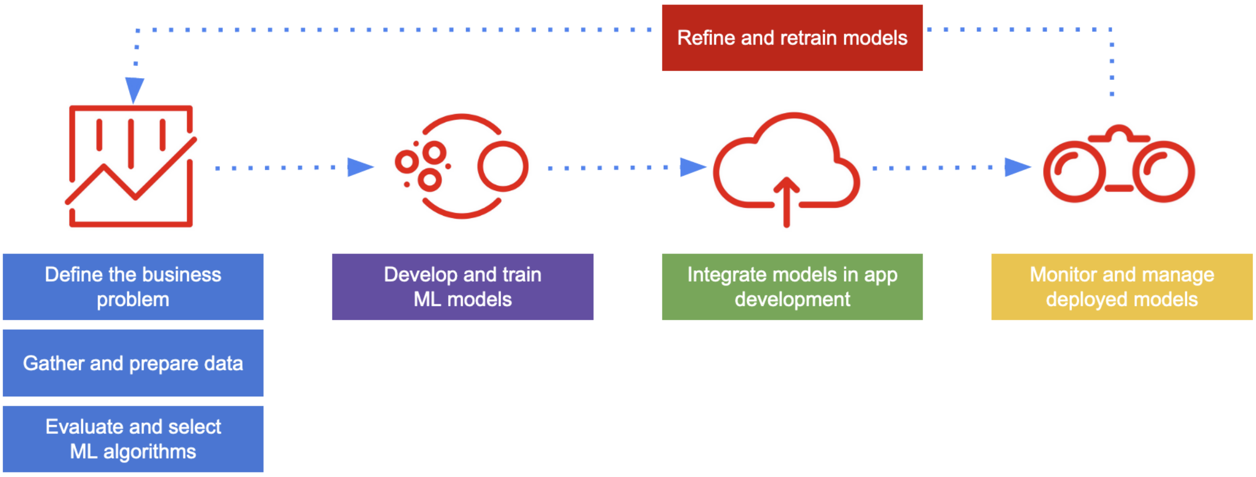
Red Hat OpenShift AI is an artificial intelligence (AI) platform that provides tools to rapidly train, serve, and monitor machine learning (ML) models onsite, in the public cloud, or at the edge.

OpenShift AI provides a powerful AI/ML platform for building AI-enabled applications. Data scientists and MLOps engineers can collaborate to move from experiment to production in a consistent environment quickly.

You can deploy OpenShift AI on any supported version of OpenShift, whether on-premise, in the cloud, or in disconnected environments. For details on supported versions, see Red Hat OpenShift AI: Supported Configurations.

1.1. DATA SCIENCE WORKFLOW

For the purpose of getting you started with OpenShift AI, Figure 1 illustrates a simplified data science workflow. The real world process of developing ML models is an iterative one.


Refine and retrain models
Define the business problem
Gather and prepare data
Evaluate and select ML algorithms
Develop and train ML models
Integrate models in app development
Monitor and manage deployed models

The simplified data science workflow for predictive AI use cases includes the following tasks:

- Defining your business problem and setting goals to solve it.
- Gathering, cleaning, and preparing data. Data often has to be federated from a range of sources, and exploring and understanding data plays a key role in the success of a data science project.
- Evaluating and selecting ML models for your business use case.
- Train models for your business use case by tuning model parameters based on your set of training data. In practice, data scientists train a range of models, and compare performance while considering tradeoffs such as time and memory constraints.
- Integrate models into an application, including deployment and testing. After model training, the next step of the workflow is production. Data scientists are often responsible for putting the model in production and making it accessible so that a developer can integrate the model into an application.

- Monitor and manage deployed models. Depending on the organization, data scientists, data engineers, or ML engineers must monitor the performance of models in production, tracking prediction and performance metrics.

- Refine and retrain models. Data scientists can evaluate model performance results and refine models to improve outcome by excluding or including features, changing the training data, and modifying other configuration parameters.

1.2. ABOUT THIS GUIDE

This guide assumes you are familiar with data science and ML Ops concepts. It describes the following tasks to get you started with using OpenShift AI:

- Log in to the OpenShift AI dashboard
- Create a data science project
- If you have data stored in Object Storage, configure a data connection to more easily access it
- Create a workbench and choose an IDE, such as JupyterLab or code-server, for your data scientist development work
- Learn where to get information about the next steps:
  - Developing and training a model
  - Automating the workflow with pipelines
  - Implementing distributed workloads
  - Testing your model
  - Deploying your model
  - Monitoring and managing your model

See also OpenShift AI tutorial: Fraud detection example . It provides step-by-step guidance for using OpenShift AI to develop and train an example model in JupyterLab, deploy the model, and refine the model by using automated pipelines.

CHAPTER 2. LOGGING IN TO OPENSHIFT AI

After you install OpenShift AI, log in to the OpenShift AI dashboard so that you can set up your development and deployment environment.

Prerequisites

- You know the OpenShift AI identity provider and your login credentials.
  - If you are a data scientist, data engineer, or ML engineer, your administrator must provide you with the OpenShift AI instance URL, for example: **https://rhods-dashboard-redhat-oai-applications.apps.example.abc1.p1.openshiftapps.com/**
- You have the latest version of one of the following supported browsers:
  - Google Chrome
  - Mozilla Firefox
  - Safari

Procedure

1. Browse to the OpenShift AI instance URL and click **Log in with OpenShift**.
   - If you have access to OpenShift Container Platform, you can browse to the OpenShift Container Platform web console and click the **Application Launcher** (▦) → **Red Hat OpenShift AI**.
2. Click the name of your identity provider, for example, **GitHub**,**Google**, or your company’s single sign-on method.
3. Enter your credentials and click **Log in** (or equivalent for your identity provider).

Verification

- The OpenShift AI dashboard opens on the **Home** page.

CHAPTER 3. 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.

CHAPTER 4. 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 accomodate 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.

4.1. 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 4.1. 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.

Image name	Description
code-server (Technology Preview)	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. NOTE Elyra-based pipelines are not available with the code-server notebook image.
RStudio Server (Technology preview)	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. 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. SeeBuilding the RStudio Server notebook images for more information. IMPORTANT **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.

Image name	Description
CUDA - RStudio Server (Technology preview)	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. 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. SeeBuilding the RStudio Server notebook images for more information. **IMPORTANT** **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. 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.

4.2. 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.

CHAPTER 5. NEXT STEPS

The following product documentation provides more information on how to develop, test, and deploy data science solutions with OpenShift AI.

Try the end-to-end tutorial

OpenShift AI tutorial - Fraud detection example

Step-by-step guidance to complete the following tasks with an example fraud detection model:

- Explore a pre-trained fraud detection model by using a Jupyter notebook.
- Deploy the model by using OpenShift AI model serving.
- Refine and train the model by using automated pipelines.

Develop and train a model in your workbench IDE

Working in your data science IDE

Learn how to access your workbench IDE (JupyterLab, code-server, or RStudio Server).

For the JupyterLab IDE, learn about the following tasks:

- Creating and importing notebooks
- Using Git to collaborate on notebooks
- Viewing and installing Python packages
- Troubleshooting common problems

Automate your ML workflow with pipelines

Working with data science pipelines

Enhance your data science projects on OpenShift AI by building portable machine learning (ML) workflows with data science pipelines, by using Docker containers. Use pipelines for continuous retraining and updating of a model based on newly received data.

Deploy and test a model

Serving models

Deploy your ML models on your OpenShift cluster to test and then integrate them into intelligent applications. When you deploy a model, it is available as a service that you can access by using API calls. You can return predictions based on data inputs that you provide through API calls.

Monitor and manage models

Serving models

The Red Hat OpenShift AI service supports model deployment options for hosting the model on Red Hat OpenShift Dedicated or Red Hat Openshift Service on AWS for integration into an external application.

Add accelerators to optimize performance

Working with accelerators

If you work with large data sets, you can use accelerators, such as NVIDIA GPUs and Habana Gaudi devices, to optimize the performance of your data science models in OpenShift AI. With accelerators, you can scale your work, reduce latency, and increase productivity.

Implement distributed workloads for higher performance

Working with distributed workloads
Implement distributed workloads to use multiple cluster nodes in parallel for faster, more efficient data processing and model training.

Explore extensions

Working with connected applications
Extend your core OpenShift AI solution with integrated third-party applications. Several leading AI/ML software technology partners, including Starburst, Intel AI Tools, Anaconda, and IBM are also available through Red Hat Marketplace.

5.1. ADDITIONAL RESOURCES

In addition to product documentation, Red Hat provides a rich set of learning resources for OpenShift AI and supported applications.

On the **Resources** page of the OpenShift AI dashboard, you can use the category links to filter the resources for various stages of your data science workflow. For example, click the **Model serving** category to display resources that describe various methods of deploying models. Click **All items** to show the resources for all categories.

For the selected category, you can apply additional options to filter the available resources. For example, you can filter by type, such as how-to articles, quick starts, or tutorials; these resources provide the answers to common questions.

For information about Red Hat OpenShift AI support requirements and limitations, see Red Hat OpenShift AI: Supported Configurations.