



Red Hat OpenShift AI Self-Managed 2- latest

Managing resources

Manage cluster resources, Jupyter notebooks, and data backup in OpenShift AI

Red Hat OpenShift AI Self-Managed 2-latest Managing resources

Manage cluster resources, Jupyter notebooks, and data backup in OpenShift AI

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

Manage cluster resources, Jupyter notebooks, and data backup in OpenShift AI.

Table of Contents

PREFACE	4
CHAPTER 1. CUSTOMIZING THE DASHBOARD	5
1.1. EDITING THE DASHBOARD CONFIGURATION FILE	5
1.2. DASHBOARD CONFIGURATION OPTIONS	7
CHAPTER 2. MANAGING APPLICATIONS THAT SHOW IN THE DASHBOARD	10
2.1. ADDING AN APPLICATION TO THE DASHBOARD	10
2.2. PREVENTING USERS FROM ADDING APPLICATIONS TO THE DASHBOARD	11
2.3. DISABLING APPLICATIONS CONNECTED TO OPENSIFT AI	12
2.4. SHOWING OR HIDING INFORMATION ABOUT ENABLED APPLICATIONS	13
2.5. HIDING THE DEFAULT JUPYTER APPLICATION	15
2.6. TROUBLESHOOTING COMMON PROBLEMS IN JUPYTER FOR ADMINISTRATORS	16
2.6.1. A user receives a 404: Page not found error when logging in to Jupyter	16
2.6.2. A user's notebook server does not start	16
2.6.3. The user receives a database or disk is full error or a no space left on device error when they run notebook cells	17
CHAPTER 3. MANAGING CLUSTER RESOURCES	19
3.1. CONFIGURING THE DEFAULT PVC SIZE FOR YOUR CLUSTER	19
3.2. RESTORING THE DEFAULT PVC SIZE FOR YOUR CLUSTER	19
3.3. OVERVIEW OF ACCELERATORS	20
3.3.1. Enabling NVIDIA GPUs	21
3.3.2. Enabling Intel Gaudi AI accelerators	22
3.4. ALLOCATING ADDITIONAL RESOURCES TO OPENSIFT AI USERS	23
3.5. TROUBLESHOOTING COMMON PROBLEMS WITH DISTRIBUTED WORKLOADS FOR ADMINISTRATORS	23
3.5.1. A user's Ray cluster is in a suspended state	23
3.5.2. A user's Ray cluster is in a failed state	24
3.5.3. A user receives a failed to call webhook error message for the CodeFlare Operator	24
3.5.4. A user receives a failed to call webhook error message for Kueue	25
3.5.5. A user's Ray cluster does not start	25
3.5.6. A user receives a Default Local Queue ... not found error message	26
3.5.7. A user receives a local_queue provided does not exist error message	26
3.5.8. A user cannot create a Ray cluster or submit jobs	27
3.5.9. The user's pod provisioned by Kueue is terminated before the user's image is pulled	28
CHAPTER 4. CUSTOMIZING COMPONENT DEPLOYMENT RESOURCES	29
4.1. OVERVIEW OF COMPONENT RESOURCE CUSTOMIZATION	29
4.2. CUSTOMIZING COMPONENT RESOURCES	29
4.3. DISABLING COMPONENT RESOURCE CUSTOMIZATION	30
4.4. RE-ENABLING COMPONENT RESOURCE CUSTOMIZATION	31
CHAPTER 5. MANAGING JUPYTER NOTEBOOK SERVERS	33
5.1. ACCESSING THE JUPYTER ADMINISTRATION INTERFACE	33
5.2. STARTING NOTEBOOK SERVERS OWNED BY OTHER USERS	33
5.3. ACCESSING NOTEBOOK SERVERS OWNED BY OTHER USERS	34
5.4. STOPPING NOTEBOOK SERVERS OWNED BY OTHER USERS	34
5.5. STOPPING IDLE NOTEBOOKS	35
5.6. CONFIGURING A CUSTOM NOTEBOOK IMAGE	36
CHAPTER 6. BACKING UP DATA	39
6.1. BACKING UP STORAGE DATA	39

CHAPTER 7. USAGE DATA COLLECTION	40
7.1. USAGE DATA COLLECTION NOTICE FOR OPENSIFT AI	40
7.2. ENABLING USAGE DATA COLLECTION	40
7.3. DISABLING USAGE DATA COLLECTION	41

PREFACE

As an OpenShift AI administrator, you can manage the following resources:

- The dashboard interface, including the visibility of navigation menu options
- Applications that show in the dashboard
- Cluster resources to support compute-intensive data science work
- Jupyter notebook servers
- Data storage backup

You can also specify whether to allow Red Hat to collect data about OpenShift AI usage in your cluster.

CHAPTER 1. CUSTOMIZING THE DASHBOARD

The OpenShift AI dashboard provides features that are designed to work for most scenarios. These features are configured in the **OdhDashboardConfig** custom resource (CR) file.

To see a description of the options in the OpenShift AI dashboard configuration file, see [Dashboard configuration options](#).

As an administrator, you can customize the interface of the dashboard, for example to show or hide some of the dashboard navigation menu options. To change the default settings of the dashboard, edit the **OdhDashboardConfig** custom resource (CR) file as described in [Editing the dashboard configuration file](#).

1.1. EDITING THE DASHBOARD CONFIGURATION FILE

As an administrator, you can customize the interface of the dashboard by editing the dashboard configuration file.

Prerequisites

- You have cluster administrator privileges for your OpenShift cluster.

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. In the **Administrator** perspective, click **Home** → **API Explorer**.
3. In the search bar, enter **OdhDashboardConfig** to filter by kind.
4. Click the **OdhDashboardConfig** custom resource (CR) to open the resource details page.
5. Select the **redhat-ods-applications** project from the **Project** list.
6. Click the **Instances** tab.
7. Click the **odh-dashboard-config** instance to open the details page.
8. Click the **YAML** tab. Here is an example **OdhDashboardConfig** file showing default values:

```
apiVersion: opendatahub.io/v1alpha
kind: OdhDashboardConfig
metadata:
  name: odh-dashboard-config
spec:
  dashboardConfig:
    enablement: true
    disableBYONImageStream: false
    disableClusterManager: false
    disableISVBadges: false
    disableInfo: false
    disableSupport: false
    disableTracking: true
    disableProjects: true
    disablePipelines: true
```

```
disableModelServing: true
disableProjectSharing: true
disableCustomServingRuntimes: false
disableAcceleratorProfiles: true
modelMetricsNamespace: ""
disablePerformanceMetrics: false
notebookController:
  enabled: true
notebookSizes:
- name: Small
  resources:
    limits:
      cpu: '2'
      memory: 2Gi
    requests:
      cpu: '1'
      memory: 1Gi
- name: Medium
  resources:
    limits:
      cpu: '4'
      memory: 4Gi
    requests:
      cpu: '2'
      memory: 2Gi
- name: Large
  resources:
    limits:
      cpu: '8'
      memory: 8Gi
    requests:
      cpu: '4'
      memory: 4Gi
modelServerSizes:
- name: Small
  resources:
    limits:
      cpu: '2'
      memory: 8Gi
    requests:
      cpu: '1'
      memory: 4Gi
- name: Medium
  resources:
    limits:
      cpu: '8'
      memory: 10Gi
    requests:
      cpu: '4'
      memory: 8Gi
- name: Large
  resources:
    limits:
      cpu: '10'
      memory: 20Gi
    requests:
```

```

cpu: '6'
memory: 16Gi
groupsConfig:
  adminGroups: 'odh-admins'
  allowedGroups: 'system:authenticated'
templateOrder:
  - 'ovms'
templateDisablement:
  - 'ovms'

```

9. Edit the values of the options that you want to change.
10. Click **Save** to apply your changes and then click **Reload** to make sure that your changes are synced to the cluster.

Verification

Log in to OpenShift AI and verify that your dashboard configurations apply.

1.2. DASHBOARD CONFIGURATION OPTIONS

The OpenShift AI dashboard includes a set of core features enabled by default that are designed to work for most scenarios. Administrators can configure the OpenShift AI dashboard from the **OdhDashboardConfig** custom resource (CR) in OpenShift.

Table 1.1. Dashboard feature configuration options

Feature	Default	Description
dashboardConfig: enablement	true	Enables admin users to add applications to the OpenShift AI dashboard Application → Enabled page. To disable this ability, set the value to false .
dashboardConfig: disableInfo	false	On the Applications → Explore page, when a user clicks on an application tile, an information panel opens with more details about the application. To disable the information panel for all applications on the Applications → Explore page, set the value to true .
dashboardConfig: disableSupport	false	Shows the Support menu option when a user clicks the Help icon in the dashboard toolbar. To hide this menu option, set the value to true .
dashboardConfig: disableClusterManager	false	Shows the Settings → Cluster settings option in the dashboard navigation menu. To hide this menu option, set the value to true .
dashboardConfig: disableTracking	true	Allows Red Hat to collect data about OpenShift AI usage in your cluster. To enable data collection, set the value to false . You can also set this option in the OpenShift AI dashboard interface from the Settings → Cluster settings navigation menu.

dashboardConfig: disableBYONImageStream	false	Shows the Settings → Notebook images option in the dashboard navigation menu. To hide this menu option, set the value to false .
dashboardConfig: disableISVBadges	false	Shows the label on a tile that indicates whether the application is “Red Hat managed”, “Partner managed”, or “Self-managed”. To hide these labels, set the value to true .
dashboardConfig: disableUserManagement	false	Shows the Settings → User management option in the dashboard navigation menu. To hide this menu option, set the value to true .
dashboardConfig: disableProjects	false	Shows the Data Science Projects option in the dashboard navigation menu. To hide this menu option, set the value to true .
dashboardConfig: disablePipelines	false	Shows the Data Science Pipelines option in the dashboard navigation menu. To hide this menu option, set the value to true .
dashboardConfig: disableModelServing	false	Shows the Model Serving option in the dashboard navigation menu and in the list of components for the data science projects. To hide Model Serving from the dashboard navigation menu and from the list of components for data science projects, set the value to true .
dashboardConfig: disableProjectSharing	false	Allows users to share access to their data science projects with other users. To prevent users from sharing data science projects, set the value to true .
dashboardConfig: disableCustomServingRuntimes	false	Shows the Serving runtimes option in the dashboard navigation menu. To hide this menu option, set the value to true .
dashboardConfig: disableKServe	false	Enables the ability to select KServe as a Serving Platform. To disable this ability, set the value to true .
dashboardConfig: disableModelMesh	false	Enables the ability to select ModelMesh as a Serving Platform. To disable this ability, set the value to true .
dashboardConfig: disableAcceleratorProfiles	false	Shows the Accelerator profiles option in the dashboard navigation menu. To hide this menu option, set the value to true .
dashboardConfig: modelMetricsNamespace	false	Enables the namespace in which the Model Serving Metrics' Prometheus Operator is installed.
dashboardConfig: disablePerformanceMetrics	false	Shows the Endpoint Performance tab on the Model Serving page. To hide this tab, set the value to true .

notebookController: enabled	true	Controls the Notebook Controller options, such as whether it is enabled in the dashboard and which parts are visible.
notebookSizes		Allows you to customize names and resources for notebooks. The Kubernetes-style sizes are shown in the drop-down menu that appears when spawning notebooks with the Notebook Controller. Note: These sizes must follow conventions. For example, requests must be smaller than limits.
ModelServerSizes		Allows you to customize names and resources for model servers.
groupsConfig		Controls access to dashboard features, such as the spawner for allowed users and the cluster settings UI for admin users.
templateOrder		Specifies the order of custom Serving Runtime templates. When the user creates a new template, it is added to this list.

CHAPTER 2. MANAGING APPLICATIONS THAT SHOW IN THE DASHBOARD

2.1. ADDING AN APPLICATION TO THE DASHBOARD

If you have installed an application in your OpenShift cluster, you can add a tile for that application to the OpenShift AI dashboard (the **Applications** → **Enabled** page) to make it accessible for OpenShift AI users.

Prerequisites

- You have cluster administrator privileges for your OpenShift cluster.
- The dashboard configuration enablement option is set to **true** (the default). Note that an admin user can disable this ability as described in [Preventing users from adding applications to the dashboard](#).

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. In the **Administrator** perspective, click **Home** → **API Explorer**.
3. On the **API Explorer** page, search for the **OdhApplication** kind.
4. Click the **OdhApplication** kind to open the resource details page.
5. On the **OdhApplication** details page, select the **redhat-ods-applications** project from the **Project** list.
6. Click the **Instances** tab.
7. Click **Create OdhApplication**.
8. On the **Create OdhApplication** page, copy the following code and paste it into the YAML editor.

```
apiVersion: dashboard.opendatahub.io/v1
kind: OdhApplication
metadata:
  name: exemplename
  namespace: redhat-ods-applications
labels:
  app: odh-dashboard
  app.kubernetes.io/part-of: odh-dashboard
spec:
  enable:
    validationConfigMap: exemplename-enable
  img: >-
    <svg width="24" height="25" viewBox="0 0 24 25" fill="none"
xmlns="http://www.w3.org/2000/svg">
  <path d="path data" fill="#ee0000"/>
</svg>
  getStartedLink: 'https://example.org/docs/quickstart.html'
```

```

route: exemplerroutename
routeNamespace: examplnamespace
displayName: Example Name
kfdefApplications: []
support: third party support
csvName: "
provider: example
docsLink: 'https://example.org/docs/index.html'
quickStart: "
getStartedMarkDown: >-
  # Example

Enter text for the information panel.

description: >-
  Enter summary text for the tile.
category: Self-managed | Partner managed | {org-name} managed

```

9. Modify the parameters in the code for your application.

TIP

To see example YAML files, click **Home** → **API Explorer**, select **OdHApplication**, click the **Instances** tab, select an instance, and then click the **YAML** tab.

10. Click **Create**. The application details page appears.
11. Log in to OpenShift AI.
12. In the left menu, click **Applications** → **Explore**.
13. Locate the new tile for your application and click it.
14. In the information pane for the application, click **Enable**.

Verification

- In the left menu of the OpenShift AI dashboard, click **Applications** → **Enabled** and verify that your application is available.

2.2. PREVENTING USERS FROM ADDING APPLICATIONS TO THE DASHBOARD

By default, admin users are allowed to add applications to the OpenShift AI dashboard **Application** → **Enabled** page.

You can disable the ability for admin users to add applications to the dashboard.

Note: The Jupyter tile is enabled by default. To disable it, see [Hiding the default Jupyter application](#).

Prerequisite

- You have cluster administrator privileges for your OpenShift cluster.

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. Open the dashboard configuration file:
 - a. In the **Administrator** perspective, click **Home** → **API Explorer**.
 - b. In the search bar, enter **OdhDashboardConfig** to filter by kind.
 - c. Click the **OdhDashboardConfig** custom resource (CR) to open the resource details page.
 - d. Select the **redhat-ods-applications** project from the **Project** list.
 - e. Click the **Instances** tab.
 - f. Click the **odh-dashboard-config** instance to open the details page.
 - g. Click the **YAML** tab.
3. In the **spec:dashboardConfig** section, set the value of **enablement** to **false** to disable the ability for dashboard users to add applications to the dashboard.
4. Click **Save** to apply your changes and then click **Reload** to make sure that your changes are synced to the cluster.

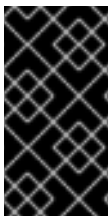
Verification

Open the OpenShift AI dashboard **Application** → **Enabled** page.

2.3. DISABLING APPLICATIONS CONNECTED TO OPENSIFT AI

You can disable applications and components so that they do not appear on the OpenShift AI dashboard when you no longer want to use them, for example, when data scientists no longer use an application or when the application license expires.

Disabling unused applications allows your data scientists to manually remove these application tiles from their OpenShift AI dashboard so that they can focus on the applications that they are most likely to use. See [Removing disabled applications from the dashboard](#) for more information about manually removing application tiles.



IMPORTANT

Do not follow this procedure when disabling the following applications:

- Anaconda Professional Edition. You cannot manually disable Anaconda Professional Edition. It is automatically disabled only when its license expires.

Prerequisites

- You have logged in to the OpenShift web console.
- You are part of the **cluster-admins** user group in OpenShift.
- You have installed or configured the service on your OpenShift cluster.

- The application or component that you want to disable is enabled and appears on the **Enabled** page.

Procedure

1. In the OpenShift web console, switch to the **Administrator** perspective.
2. Switch to the **redhat-ods-applications** project.
3. Click **Operators** → **Installed Operators**.
4. Click on the Operator that you want to uninstall. You can enter a keyword into the **Filter by name** field to help you find the Operator faster.
5. Delete any Operator resources or instances by using the tabs in the Operator interface. During installation, some Operators require the administrator to create resources or start process instances using tabs in the Operator interface. These must be deleted before the Operator can uninstall correctly.
6. On the **Operator Details** page, click the **Actions** drop-down menu and select **Uninstall Operator**.
An **Uninstall Operator?** dialog box is displayed.
7. Select **Uninstall** to uninstall the Operator, Operator deployments, and pods. After this is complete, the Operator stops running and no longer receives updates.



IMPORTANT

Removing an Operator does not remove any custom resource definitions or managed resources for the Operator. Custom resource definitions and managed resources still exist and must be cleaned up manually. Any applications deployed by your Operator and any configured off-cluster resources continue to run and must be cleaned up manually.

Verification

- The Operator is uninstalled from its target clusters.
- The Operator no longer appears on the **Installed Operators** page.
- The disabled application is no longer available for your data scientists to use, and is marked as **Disabled** on the **Enabled** page of the OpenShift AI dashboard. This action may take a few minutes to occur following the removal of the Operator.

2.4. SHOWING OR HIDING INFORMATION ABOUT ENABLED APPLICATIONS

If you have installed another application in your OpenShift cluster, you can add a tile for that application to the OpenShift AI dashboard (the **Applications** → **Enabled** page) to make it accessible for OpenShift AI users.

Prerequisites

- You have cluster administrator privileges for your OpenShift cluster.

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. In the **Administrator** perspective, click **Home** → **API Explorer**.
3. On the **API Explorer** page, search for the **OdhApplication** kind.
4. Click the **OdhApplication** kind to open the resource details page.
5. On the **OdhApplication** details page, select the **redhat-ods-applications** project from the **Project** list.
6. Click the **Instances** tab.
7. Click **Create OdhApplication**.
8. On the **Create OdhApplication** page, copy the following code and paste it into the YAML editor.

```

apiVersion: dashboard.opendatahub.io/v1
kind: OdhApplication
metadata:
  name: examplename
  namespace: redhat-ods-applications
  labels:
    app: odh-dashboard
    app.kubernetes.io/part-of: odh-dashboard
spec:
  enable:
    validationConfigMap: examplename-enable
  img: >-
    <svg width="24" height="25" viewBox="0 0 24 25" fill="none"
xmlns="http://www.w3.org/2000/svg">
    <path d="path data" fill="#ee0000"/>
    </svg>
  getStartedLink: 'https://example.org/docs/quickstart.html'
  route: exampleroutename
  routeNamespace: examplenamespace
  displayName: Example Name
  kfdefApplications: []
  support: third party support
  csvName: "
  provider: example
  docsLink: 'https://example.org/docs/index.html'
  quickStart: "
  getStartedMarkDown: >-
    # Example

```

Enter text for the information panel.

```
description: >-
```

Enter summary text for the tile.

```
category: Self-managed | Partner managed | Red Hat managed
```

9. Modify the parameters in the code for your application.

TIP

To see example YAML files, click **Home** → **API Explorer**, select **OdhApplication**, click the **Instances** tab, select an instance, and then click the **YAML** tab.

10. Click **Create**. The application details page appears.
11. Log in to OpenShift AI.
12. In the left menu, click **Applications** → **Explore**.
13. Locate the new tile for your application and click it.
14. In the information pane for the application, click **Enable**.

Verification

- In the left menu of the OpenShift AI dashboard, click **Applications** → **Enabled** and verify that your application is available.

2.5. HIDING THE DEFAULT JUPYTER APPLICATION

The OpenShift AI dashboard includes Jupyter as an enabled application by default.

To hide the Jupyter tile from the list of Enabled applications, edit the dashboard configuration file.

Prerequisite

- You have cluster administrator privileges for your OpenShift cluster.

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. Open the dashboard configuration file:
 - a. In the **Administrator** perspective, click **Home** → **API Explorer**.
 - b. In the search bar, enter **OdhDashboardConfig** to filter by kind.
 - c. Click the **OdhDashboardConfig** custom resource (CR) to open the resource details page.
 - d. Select the **redhat-ods-applications** project from the **Project** list.
 - e. Click the **Instances** tab.
 - f. Click the **odh-dashboard-config** instance to open the details page.
 - g. Click the **YAML** tab.
3. In the **spec:notebookController** section, set the value of **enabled** to **false** to hide the Jupyter tile from the list of Enabled applications.
4. Click **Save** to apply your changes and then click **Reload** to make sure that your changes are synced to the cluster.

Verification

In the OpenShift AI dashboard, select **Applications** > **Enabled**. You should not see the Jupyter tile.

2.6. TROUBLESHOOTING COMMON PROBLEMS IN JUPYTER FOR ADMINISTRATORS

If your users are experiencing errors in Red Hat OpenShift AI relating to Jupyter, their notebooks, or their notebook server, read this section to understand what could be causing the problem, and how to resolve the problem.

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

2.6.1. A user receives a 404: Page not found error when logging in to Jupyter

Problem

If you have configured specialized user groups for OpenShift AI, the user name might not be added to the default user group for OpenShift AI.

Diagnosis

Check whether the user is part of the default user group.

1. Find the names of groups allowed access to Jupyter.
 - a. Log in to the OpenShift web console.
 - b. Click **User Management** → **Groups**.
 - c. Click the name of your user group, for example, **rhoai-users**.
The **Group details** page for that group appears.
2. Click the **Details** tab for the group and confirm that the **Users** section for the relevant group contains the users who have permission to access Jupyter.

Resolution

- If the user is not added to any of the groups with permission access to Jupyter, follow [Adding users](#) to add them.
- If the user is already added to a group with permission to access Jupyter, contact Red Hat Support.

2.6.2. A user's notebook server does not start

Problem

The OpenShift cluster that hosts the user's notebook server might not have access to enough resources, or the Jupyter pod may have failed.

Diagnosis

1. Log in to the OpenShift web console.
2. Delete and restart the notebook server pod for this user.

- a. Click **Workloads** → **Pods** and set the **Project** to **rhods-notebooks**.
- b. Search for the notebook server pod that belongs to this user, for example, **jupyter-nb-`<username>-*`**.
If the notebook server pod exists, an intermittent failure may have occurred in the notebook server pod.

If the notebook server pod for the user does not exist, continue with diagnosis.
3. Check the resources currently available in the OpenShift cluster against the resources required by the selected notebook server image.
If worker nodes with sufficient CPU and RAM are available for scheduling in the cluster, continue with diagnosis.
4. Check the state of the Jupyter pod.

Resolution

- If there was an intermittent failure of the notebook server pod:
 - a. Delete the notebook server pod that belongs to the user.
 - b. Ask the user to start their notebook server again.
- If the notebook server does not have sufficient resources to run the selected notebook server image, either add more resources to the OpenShift cluster, or choose a smaller image size.
- If the Jupyter pod is in a **FAILED** state:
 - a. Retrieve the logs for the **jupyter-nb-*** pod and send them to Red Hat Support for further evaluation.
 - b. Delete the **jupyter-nb-*** pod.
- If none of the previous resolutions apply, contact Red Hat Support.

2.6.3. The user receives a database or disk is full error or a no space left on device error when they run notebook cells

Problem

The user might have run out of storage space on their notebook server.

Diagnosis

1. Log in to Jupyter and start the notebook server that belongs to the user having problems. If the notebook server does not start, follow these steps to check whether the user has run out of storage space:
 - a. Log in to the OpenShift web console.
 - b. Click **Workloads** → **Pods** and set the **Project** to **rhods-notebooks**.
 - c. Click the notebook server pod that belongs to this user, for example, **jupyter-nb-`<idp>-<username>-*`**.

- d. Click **Logs**. The user has exceeded their available capacity if you see lines similar to the following:

```
Unexpected error while saving file: XXXX database or disk is full
```

Resolution

- Increase the user's available storage by expanding their persistent volume: [Expanding persistent volumes](#)
- Work with the user to identify files that can be deleted from the **/opt/app-root/src** directory on their notebook server to free up their existing storage space.



NOTE

When you delete files using the JupyterLab file explorer, the files move to the hidden **/opt/app-root/src/.local/share/Trash/files** folder in the persistent storage for the notebook. To free up storage space for notebooks, you must permanently delete these files.

CHAPTER 3. MANAGING CLUSTER RESOURCES

3.1. CONFIGURING THE DEFAULT PVC SIZE FOR YOUR CLUSTER

To configure how resources are claimed within your OpenShift AI cluster, you can change the default size of the cluster's persistent volume claim (PVC) ensuring that the storage requested matches your common storage workflow. PVCs are requests for resources in your cluster and also act as claim checks to the resource.

Prerequisites

- You have logged in to Red Hat OpenShift AI.



NOTE

Changing the PVC setting restarts the Jupyter pod and makes Jupyter unavailable for up to 30 seconds. As a workaround, it is recommended that you perform this action outside of your organization's typical working day.

Procedure

1. From the OpenShift AI dashboard, click **Settings** → **Cluster settings**.
2. Under **PVC size**, enter a new size in gibibytes. The minimum size is 1 GiB, and the maximum size is 16384 GiB.
3. Click **Save changes**.

Verification

- New PVCs are created with the default storage size that you configured.

Additional resources

- [Understanding persistent storage](#)

3.2. RESTORING THE DEFAULT PVC SIZE FOR YOUR CLUSTER

To change the size of resources utilized within your OpenShift AI cluster, you can restore the default size of your cluster's persistent volume claim (PVC).

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- You are part of the administrator group for OpenShift AI in OpenShift.

Procedure

1. From the OpenShift AI dashboard, click **Settings** → **Cluster settings**.
2. Click **Restore Default** to restore the default PVC size of 20GiB.

3. Click **Save changes**.

Verification

- New PVCs are created with the default storage size of 20 GiB.

Additional resources

- [Understanding persistent storage](#)

3.3. OVERVIEW OF ACCELERATORS

If you work with large data sets, you can use accelerators to optimize the performance of your data science models in OpenShift AI. With accelerators, you can scale your work, reduce latency, and increase productivity. You can use accelerators in OpenShift AI to assist your data scientists in the following tasks:

- Natural language processing (NLP)
- Inference
- Training deep neural networks
- Data cleansing and data processing

OpenShift AI supports the following accelerators:

- NVIDIA graphics processing units (GPUs)
 - To use compute-heavy workloads in your models, you can enable NVIDIA graphics processing units (GPUs) in OpenShift AI.
 - To enable GPUs on OpenShift, you must install the [NVIDIA GPU Operator](#).
- Intel Gaudi AI accelerators
 - Intel provides hardware accelerators intended for deep learning workloads. You can use the Habana libraries and software associated with Intel Gaudi AI accelerators available from your notebook.
 - Before you can enable Intel Gaudi AI accelerators in OpenShift AI, you must install the necessary dependencies and the version of the HabanaAI Operator that matches the Habana version of the HabanaAI workbench image in your deployment. For more information about how to enable your OpenShift environment for Intel Gaudi AI accelerators, see [HabanaAI Operator v1.10 for OpenShift](#) and [HabanaAI Operator v1.13 for OpenShift](#).
 - You can enable Intel Gaudi AI accelerators on-premises or with AWS DL1 compute nodes on an AWS instance.

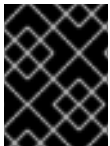
Before you can use an accelerator in OpenShift AI, your OpenShift instance must contain an associated accelerator profile. For accelerators that are new to your deployment, you must configure an accelerator profile for the accelerator in context. You can create an accelerator profile from the **Settings** → **Accelerator profiles** page on the OpenShift AI dashboard. If your deployment contains existing accelerators that had associated accelerator profiles already configured, an accelerator profile is automatically created after you upgrade to the latest version of OpenShift AI.

Additional resources

- [HabanaAI Operator v1.10 for OpenShift](#)
- [HabanaAI Operator v1.13 for OpenShift](#)
- [Habana, an Intel Company](#)
- [Amazon EC2 DL1 Instances](#)
- [lspci\(8\) - Linux man page](#)

3.3.1. Enabling NVIDIA GPUs

Before you can use NVIDIA GPUs in OpenShift AI, you must install the NVIDIA GPU Operator.



IMPORTANT

If you are using OpenShift AI in a disconnected self-managed environment, see [Enabling NVIDIA GPUs](#) instead.

Prerequisites

- You have logged in to your OpenShift cluster.
- You have the **cluster-admin** role in your OpenShift cluster.

Procedure

1. To enable GPU support on an OpenShift cluster, follow the instructions here: [NVIDIA GPU Operator on Red Hat OpenShift Container Platform](#) in the NVIDIA documentation.
2. Delete the **migration-gpu-status** ConfigMap.
 - a. In the OpenShift web console, switch to the **Administrator** perspective.
 - b. Set the **Project** to **All Projects** or **redhat-ods-applications** to ensure you can see the appropriate ConfigMap.
 - c. Search for the **migration-gpu-status** ConfigMap.
 - d. Click the action menu (**:**) and select **Delete ConfigMap** from the list. The **Delete ConfigMap** dialog appears.
 - e. Inspect the dialog and confirm that you are deleting the correct ConfigMap.
 - f. Click **Delete**.
3. Restart the dashboard replicaset.
 - a. In the OpenShift web console, switch to the **Administrator** perspective.
 - b. Click **Workloads** → **Deployments**.
 - c. Set the **Project** to **All Projects** or **redhat-ods-applications** to ensure you can see the appropriate deployment.

- d. Search for the **rhods-dashboard** deployment.
- e. Click the action menu (:) and select **Restart Rollout** from the list.
- f. Wait until the **Status** column indicates that all pods in the rollout have fully restarted.

Verification

- The NVIDIA GPU Operator appears on the **Operators → Installed Operators** page in the OpenShift web console.
- The reset **migration-gpu-status** instance is present on the **Instances** tab on the **AcceleratorProfile** custom resource definition (CRD) details page.



NOTE

In OpenShift AI 2-latest, Red Hat supports the use of accelerators within the same cluster only. Red Hat does not support remote direct memory access (RDMA) between accelerators, or the use of accelerators across a network, for example, by using technology such as NVIDIA GPUDirect or NVLink.

After installing the NVIDIA GPU Operator, create an accelerator profile as described in [Working with accelerator profiles](#).

3.3.2. Enabling Intel Gaudi AI accelerators

Before you can use Intel Gaudi AI accelerators in OpenShift AI, you must install the necessary dependencies and deploy the HabanaAI Operator.

Prerequisites

- You have logged in to OpenShift.
- You have the **cluster-admin** role in OpenShift.

Procedure

1. To enable Intel Gaudi AI accelerators in OpenShift AI, follow the instructions at [HabanaAI Operator for OpenShift](#).

Verification

- From the **Administrator** perspective, the following Operators appear on the **Operators → Installed Operators** page.
 - HabanaAI
 - Node Feature Discovery (NFD)
 - Kernel Module Management (KMM)

After installing the HabanaAI Operator, create an accelerator profile as described in [Working with accelerator profiles](#).

Additional resources

- [HabanaAI Operator v1.10 for OpenShift.](#)
- [HabanaAI Operator v1.13 for OpenShift.](#)

3.4. ALLOCATING ADDITIONAL RESOURCES TO OPENSIFT AI USERS

As a cluster administrator, you can allocate additional resources to a cluster to support compute-intensive data science work. This support includes increasing the number of nodes in the cluster and changing the cluster's allocated machine pool.

For more information about allocating additional resources to an OpenShift cluster, see [Manually scaling a compute machine set](#).

3.5. TROUBLESHOOTING COMMON PROBLEMS WITH DISTRIBUTED WORKLOADS FOR ADMINISTRATORS

If your users are experiencing errors in Red Hat OpenShift AI relating to distributed workloads, read this section to understand what could be causing the problem, and how to resolve the problem.

If the problem is not documented here or in the release notes, contact Red Hat Support.

3.5.1. A user's Ray cluster is in a suspended state

Problem

The resource quota specified in the cluster queue configuration might be insufficient, or the resource flavor might not yet be created.

Diagnosis

The user's Ray cluster head pod or worker pods remain in a suspended state. Check the status of the **Workloads** resource that is created with the **RayCluster** resource. The **status.conditions.message** field provides the reason for the suspended state, as shown in the following example:

```
status:
  conditions:
    - lastTransitionTime: '2024-05-29T13:05:09Z'
      message: 'couldn't assign flavors to pod set small-group-jobtest12: insufficient quota for
nvidia.com/gpu in flavor default-flavor in ClusterQueue'
```

Resolution

1. Check whether the resource flavor is created, as follows:
 - a. In the OpenShift console, select the user's project from the **Project** list.
 - b. Click **Home** → **Search**, and from the **Resources** list, select **ResourceFlavor**.
 - c. If necessary, create the resource flavor.
2. Check the cluster queue configuration in the user's code, to ensure that the resources that they requested are within the limits defined for the project.

3. If necessary, increase the resource quota.

For information about configuring resource flavors and quotas, see [Configuring quota management for distributed workloads](#).

3.5.2. A user's Ray cluster is in a failed state

Problem

The user might have insufficient resources.

Diagnosis

The user's Ray cluster head pod or worker pods are not running. When a Ray cluster is created, it initially enters a **failed** state. This failed state usually resolves after the reconciliation process completes and the Ray cluster pods are running.

Resolution

If the failed state persists, complete the following steps:

1. In the OpenShift console, select the user's project from the **Project** list.
2. Click **Workloads → Pods**.
3. Click the user's pod name to open the pod details page.
4. Click the **Events** tab, and review the pod events to identify the cause of the problem.
5. Check the status of the **Workloads** resource that is created with the **RayCluster** resource. The **status.conditions.message** field provides the reason for the failed state.

3.5.3. A user receives a failed to call webhook error message for the CodeFlare Operator

Problem

After the user runs the **cluster.up()** command, the following error is shown:

```
ApiException: (500)
Reason: Internal Server Error
HTTP response body: {"kind":"Status","apiVersion":"v1","metadata":
{},"status":"Failure","message":"Internal error occurred: failed calling webhook
\"mraycluster.ray.openshift.ai\": failed to call webhook: Post \"https://codeflare-operator-webhook-
service.redhat-ods-applications.svc:443/mutate-ray-io-v1-raycluster?timeout=10s\": no endpoints
available for service \"codeflare-operator-webhook-service\"\",\"reason\":\"InternalError\",\"details\":
{\"causes\":[{\"message\":\"failed calling webhook \"mraycluster.ray.openshift.ai\": failed to call webhook:
Post \"https://codeflare-operator-webhook-service.redhat-ods-applications.svc:443/mutate-ray-io-v1-
raycluster?timeout=10s\": no endpoints available for service \"codeflare-operator-webhook-
service\""}]}},\"code\":500}
```

Diagnosis

The CodeFlare Operator pod might not be running.

Resolution

1. In the OpenShift console, select the user's project from the **Project** list.
2. Click **Workloads → Pods**
3. Verify that the CodeFlare Operator pod is running. If necessary, restart the CodeFlare Operator pod.
4. Review the logs for the CodeFlare Operator pod to verify that the webhook server is serving, as shown in the following example:

```
INFO controller-runtime.webhook Serving webhook server {"host": "", "port": 9443}
```

3.5.4. A user receives a failed to call webhook error message for Kueue

Problem

After the user runs the **cluster.up()** command, the following error is shown:

```
ApiException: (500)
Reason: Internal Server Error
HTTP response body: {"kind":"Status","apiVersion":"v1","metadata":
{},"status":"Failure","message":"Internal error occurred: failed calling webhook \"mraycluster.kb.io\":
failed to call webhook: Post \"https://kueue-webhook-service.redhat-ods-applications.svc:443/mutate-
ray-io-v1-raycluster?timeout=10s\": no endpoints available for service \"kueue-webhook-
service\"","reason":"InternalError","details":{"causes":[{"message":"failed calling webhook
\"mraycluster.kb.io\": failed to call webhook: Post \"https://kueue-webhook-service.redhat-ods-
applications.svc:443/mutate-ray-io-v1-raycluster?timeout=10s\": no endpoints available for service
\"kueue-webhook-service\""}]},"code":500}
```

Diagnosis

The Kueue pod might not be running.

Resolution

1. In the OpenShift console, select the user's project from the **Project** list.
2. Click **Workloads → Pods**
3. Verify that the Kueue pod is running. If necessary, restart the Kueue pod.
4. Review the logs for the Kueue pod to verify that the webhook server is serving, as shown in the following example:

```
{"level":"info","ts":"2024-06-24T14:36:24.255137871Z","logger":"controller-
runtime.webhook","caller":"webhook/server.go:242","msg":"Serving webhook
server","host":"","port":9443}
```

3.5.5. A user's Ray cluster does not start

Problem

After the user runs the **cluster.up()** command, when they run either the **cluster.details()** command or the **cluster.status()** command, the Ray cluster status remains as **Starting** instead of changing to **Ready**. No pods are created.

Diagnosis

Check the status of the **Workloads** resource that is created with the **RayCluster** resource. The **status.conditions.message** field provides the reason for remaining in the **Starting** state. Similarly, check the **status.conditions.message** field for the **RayCluster** resource.

Resolution

1. In the OpenShift console, select the user's project from the **Project** list.
2. Click **Workloads** → **Pods**
3. Verify that the KubeRay pod is running. If necessary, restart the KubeRay pod.
4. Review the logs for the KubeRay pod to identify errors.

3.5.6. A user receives a Default Local Queue ... not found error message

Problem

After the user runs the **cluster.up()** command, the following error is shown:

```
Default Local Queue with kueue.x-k8s.io/default-queue: true annotation not found please create a default Local Queue or provide the local_queue name in Cluster Configuration.
```

Diagnosis

No default local queue is defined, and a local queue is not specified in the cluster configuration.

Resolution

1. Check whether a local queue exists in the user's project, as follows:
 - a. In the OpenShift console, select the user's project from the **Project** list.
 - b. Click **Home** → **Search**, and from the **Resources** list, select **LocalQueue**.
 - c. If no local queues are found, create a local queue.
 - d. Provide the user with the details of the local queues in their project, and advise them to add a local queue to their cluster configuration.
2. Define a default local queue.
For information about creating a local queue and defining a default local queue, see [Configuring quota management for distributed workloads](#).

3.5.7. A user receives a local_queue provided does not exist error message

Problem

After the user runs the **cluster.up()** command, the following error is shown:

```
local_queue provided does not exist or is not in this namespace. Please provide the correct local_queue name in Cluster Configuration.
```

Diagnosis

An incorrect value is specified for the local queue in the cluster configuration, or an incorrect default local queue is defined. The specified local queue either does not exist, or exists in a different namespace.

Resolution

- a. In the OpenShift console, select the user's project from the **Project** list.
 1. Click **Search**, and from the **Resources** list, select **LocalQueue**.
 2. Resolve the problem in one of the following ways:
 - If no local queues are found, create a local queue.
 - If one or more local queues are found, provide the user with the details of the local queues in their project. Advise the user to ensure that they spelled the local queue name correctly in their cluster configuration, and that the **namespace** value in the cluster configuration matches their project name. If the user does not specify a **namespace** value in the cluster configuration, the Ray cluster is created in the current project.
 3. Define a default local queue.
For information about creating a local queue and defining a default local queue, see [Configuring quota management for distributed workloads](#).

3.5.8. A user cannot create a Ray cluster or submit jobs

Problem

After the user runs the **cluster.up()** command, an error similar to the following text is shown:

```

RuntimeError: Failed to get RayCluster CustomResourceDefinition: (403)
Reason: Forbidden
HTTP response body: {"kind":"Status","apiVersion":"v1","metadata":
 {}, "status":"Failure", "message":"rayclusters.ray.io is forbidden: User
 \"system:serviceaccount:regularuser-project:regularuser-workbench\" cannot list resource
 \"rayclusters\" in API group \"ray.io\" in the namespace \"regularuser-
 project\", \"reason\":\"Forbidden\", \"details\":{\"group\":\"ray.io\", \"kind\":\"rayclusters\"}, \"code\":403}

```

Diagnosis

The correct OpenShift login credentials are not specified in the **TokenAuthentication** section of the user's notebook code.

Resolution

1. Advise the user to identify and specify the correct OpenShift login credentials as follows:
 - a. In the OpenShift console header, click your username and click **Copy login command**.
 - b. In the new tab that opens, log in as the user whose credentials you want to use.
 - c. Click **Display Token**.
 - d. From the **Log in with this token** section, copy the **token** and **server** values.

- e. Specify the copied **token** and **server** values in your notebook code as follows:

```
auth = TokenAuthentication(  
    token = "<token>",  
    server = "<server>",  
    skip_tls=False  
)  
auth.login()
```

2. Verify that the user has the correct permissions and is part of the **rhoai-users** group.

3.5.9. The user's pod provisioned by Kueue is terminated before the user's image is pulled

Problem

Kueue waits for a period of time before marking a workload as ready, to enable all of the workload pods to become provisioned and running. By default, Kueue waits for 5 minutes. If the pod image is very large and is still being pulled after the 5-minute waiting period elapses, Kueue fails the workload and terminates the related pods.

Diagnosis

1. In the OpenShift console, select the user's project from the **Project** list.
2. Click **Workloads** → **Pods**
3. Click the user's pod name to open the pod details page.
4. Click the **Events** tab, and review the pod events to check whether the image pull completed successfully.

Resolution

If the pod takes more than 5 minutes to pull the image, resolve the problem in one of the following ways:

- Add an **OnFailure** restart policy for resources that are managed by Kueue.
- In the **redhat-ods-applications** namespace, edit the **kueue-manager-config** ConfigMap to set a custom timeout for the **waitForPodsReady** property. For more information about this configuration option, see [Enabling waitForPodsReady](#) in the Kueue documentation.

CHAPTER 4. CUSTOMIZING COMPONENT DEPLOYMENT RESOURCES

4.1. OVERVIEW OF COMPONENT RESOURCE CUSTOMIZATION

You can customize deployment resources that are related to the Red Hat OpenShift AI Operator, for example, CPU and memory limits and requests. For resource customizations to persist without being overwritten by the Operator, the **opendatahub.io/managed: true** annotation must not be present in the YAML file for the component deployment. This annotation is absent by default.

The following table shows the deployment names for each component in the **redhat-ods-applications** namespace:

Component	Deployment names
CodeFlare	codeflare-operator-manager
KServe	<ul style="list-style-type: none"> • kserve-controller-manager • odh-model-controller
TrustyAI	trustyai-service-operator-controller-manager
Ray	kuberay-operator
Kueue	kueue-controller-manager
Workbenches	<ul style="list-style-type: none"> • notebook-controller-deployment • odh-notebook-controller-manager
Dashboard	rhods-dashboard
Model serving	<ul style="list-style-type: none"> • modelmesh-controller • odh-model-controller
Data science pipelines	data-science-pipelines-operator-controller-manager
Training Operator	kubeflow-training-operator

4.2. CUSTOMIZING COMPONENT RESOURCES

You can customize component deployment resources by updating the **.spec.template.spec.containers.resources** section of the YAML file for the component deployment.

Prerequisites

- You have cluster administrator privileges for your OpenShift cluster.
- You are part of the administrator group for OpenShift AI in OpenShift.

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. In the **Administrator** perspective, click **Workloads > Deployments**.
3. From the **Project** drop-down list, select **redhat-ods-applications**.
4. In the **Name** column, click the name of the deployment for the component that you want to customize resources for.



NOTE

For more information about the deployment names for each component, see [Overview of component resource customization](#).

5. On the **Deployment details** page that appears, click the **YAML** tab.
6. Find the **.spec.template.spec.containers.resources** section.
7. Update the value of the resource that you want to customize. For example, to update the memory limit to 500Mi, make the following change:

```
containers:
  - resources:
      limits:
        cpu: '2'
        memory: 500Mi
      requests:
        cpu: '1'
        memory: 1Gi
```

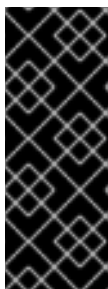
8. Click **Save**.
9. Click **Reload**.

Verification

- Log in to OpenShift AI and verify that your resource changes apply.

4.3. DISABLING COMPONENT RESOURCE CUSTOMIZATION

You can disable customization of component deployment resources, and restore default values, by adding the **opendatahub.io/managed: true** annotation to the YAML file for the component deployment.



IMPORTANT

Manually removing or setting the **opendatahub.io/managed: true** annotation to **false** after manually adding it to the YAML file for a component deployment might cause unexpected cluster issues.

To remove the annotation from a deployment, use the steps described in [Re-enabling component resource customization](#).

Prerequisites

- You have cluster administrator privileges for your OpenShift cluster.
- You are part of the administrator group for OpenShift AI in OpenShift.

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. In the **Administrator** perspective, click **Workloads > Deployments**.
3. From the **Project** drop-down list, select **redhat-ods-applications**.
4. In the **Name** column, click the name of the deployment for the component to which you want to add the annotation.



NOTE

For more information about the deployment names for each component, see [Overview of component resource customization](#).

5. On the **Deployment details** page that appears, click the **YAML** tab.
6. Find the **metadata.annotations:** section.
7. Add the **opendatahub.io/managed: true** annotation.

```
metadata:
  annotations:
    opendatahub.io/managed: true
```

8. Click **Save**.
9. Click **Reload**.

Verification

- The **opendatahub.io/managed: true** annotation appears in the YAML file for the component deployment.

4.4. RE-ENABLING COMPONENT RESOURCE CUSTOMIZATION

You can re-enable customization of component deployment resources after manually disabling it.



IMPORTANT


Manually removing or setting the **opendatahub.io/managed:** annotation to **false** after adding it to the YAML file for a component deployment might cause unexpected cluster issues.

To remove the annotation from a deployment, use the following steps to delete the deployment. The controller pod for the deployment will automatically redeploy with the default settings.

Prerequisites

- You have cluster administrator privileges for your OpenShift cluster.
- You are part of the administrator group for OpenShift AI in OpenShift.

Procedure

1. Log in to the OpenShift console as a cluster administrator.
2. In the **Administrator** perspective, click **Workloads > Deployments**.
3. From the **Project** drop-down list, select **redhat-ods-applications**.
4. In the **Name** column, click the name of the deployment for the component for which you want to remove the annotation.
5. Click the Options menu .
6. Click **Delete Deployment**

Verification

- The controller pod for the deployment automatically redeploys with the default settings.

CHAPTER 5. MANAGING JUPYTER NOTEBOOK SERVERS

5.1. ACCESSING THE JUPYTER ADMINISTRATION INTERFACE

You can use the Jupyter administration interface to control notebook servers in your Red Hat OpenShift AI environment.

Prerequisite

- You are part of the OpenShift administrator group. For more information, see [Adding administrative users in OpenShift](#).

Procedure

- To access the Jupyter administration interface from OpenShift AI, perform the following actions:
 - i. In OpenShift AI, in the **Applications** section of the left menu, click **Enabled**.
 - ii. Locate the Jupyter tile and click **Launch application**.
 - iii. On the page that opens when you launch Jupyter, click the **Administration** tab. The **Administration** page opens.
- To access the Jupyter administration interface from JupyterLab, perform the following actions:
 - i. Click **File** → **Hub Control Panel**
 - ii. On the page that opens in OpenShift AI, click the **Administration** tab. The **Administration** page opens.

Verification

- You can see the Jupyter administration interface.

5.2. STARTING NOTEBOOK SERVERS OWNED BY OTHER USERS

Administrators can start a notebook server for another existing user from the Jupyter administration interface.

Prerequisites

- You are part of the OpenShift administrator group. For more information, see [Adding administrative users in OpenShift](#).
- You have launched the Jupyter application, as described in [Starting a Jupyter notebook server](#).

Procedure

1. On the page that opens when you launch Jupyter, click the **Administration** tab.
2. On the **Administration** tab, perform the following actions:
 - a. In the **Users** section, locate the user whose notebook server you want to start.

- b. Click **Start server** beside the relevant user.
- c. Complete the **Start a notebook server** page.
- d. Optional: Select the **Start server in current tab** checkbox if necessary.
- e. Click **Start server**.
After the server starts, you see one of the following behaviors:
 - If you previously selected the **Start server in current tab** checkbox, the JupyterLab interface opens in the current tab of your web browser.
 - If you did not previously select the **Start server in current tab** checkbox, the **Starting server** dialog box prompts you to open the server in a new browser tab or in the current tab.
The JupyterLab interface opens according to your selection.

Verification

- The JupyterLab interface opens.

5.3. ACCESSING NOTEBOOK SERVERS OWNED BY OTHER USERS

Administrators can access notebook servers that are owned by other users to correct configuration errors or to help them troubleshoot problems with their environment.

Prerequisites

- You are part of the OpenShift administrator group. For more information, see [Adding administrative users in OpenShift](#).
- You have launched the Jupyter application, as described in [Starting a Jupyter notebook server](#).
- The notebook server that you want to access is running.

Procedure

1. On the page that opens when you launch Jupyter, click the **Administration** tab.
2. On the **Administration** page, perform the following actions:
 - a. In the **Users** section, locate the user that the notebook server belongs to.
 - b. Click **View server** beside the relevant user.
 - c. On the **Notebook server control panel** page, click **Access notebook server**.

Verification

- The user's notebook server opens in JupyterLab.

5.4. STOPPING NOTEBOOK SERVERS OWNED BY OTHER USERS

Administrators can stop notebook servers that are owned by other users to reduce resource consumption on the cluster, or as part of removing a user and their resources from the cluster.

Prerequisites

- If you are using specialized OpenShift AI groups, you are part of the administrator group (for example, **rhoai-admins**). If you are not using specialized groups, you are part of the OpenShift administrator group. For more information, see [Adding administrative users in OpenShift](#).
- You have launched the Jupyter application, as described in [Starting a Jupyter notebook server](#).
- The notebook server that you want to stop is running.

Procedure

1. On the page that opens when you launch Jupyter, click the **Administration** tab.
2. Stop one or more servers.
 - If you want to stop one or more specific servers, perform the following actions:
 - i. In the **Users** section, locate the user that the notebook server belongs to.
 - ii. To stop the notebook server, perform one of the following actions:
 - Click the action menu (**⋮**) beside the relevant user and select **Stop server**.
 - Click **View server** beside the relevant user and then click **Stop notebook server**. The **Stop server** dialog box appears.
 - iii. Click **Stop server**.
 - If you want to stop all servers, perform the following actions:
 - i. Click the **Stop all servers** button.
 - ii. Click **OK** to confirm stopping all servers.

Verification

- The **Stop server** link beside each server changes to a **Start server** link when the notebook server has stopped.

5.5. STOPPING IDLE NOTEBOOKS

You can reduce resource usage in your OpenShift AI deployment by stopping notebook servers that have been idle (without logged in users) for a period of time. This is useful when resource demand in the cluster is high. By default, idle notebooks are not stopped after a specific time limit.



NOTE

If you have configured your cluster settings to disconnect all users from a cluster after a specified time limit, then this setting takes precedence over the idle notebook time limit. Users are logged out of the cluster when their session duration reaches the cluster-wide time limit.

Prerequisites

- You have logged in to Red Hat OpenShift AI.

- You are part of the administrator group for OpenShift AI in OpenShift.

Procedure

1. From the OpenShift AI dashboard, click **Settings** → **Cluster settings**.
2. Under **Stop idle notebooks**, select **Stop idle notebooks after**.
3. Enter a time limit, in **hours** and **minutes**, for when idle notebooks are stopped.
4. Click **Save changes**.

Verification

- The **notebook-controller-culler-config** ConfigMap, located in the **redhat-ods-applications** project on the **Workloads** → **ConfigMaps** page, contains the following culling configuration settings:
 - **ENABLE_CULLING**: Specifies if the culling feature is enabled or disabled (this is **false** by default).
 - **IDLENESS_CHECK_PERIOD**: The polling frequency to check for a notebook's last known activity (in minutes).
 - **CULL_IDLE_TIME**: The maximum allotted time to scale an inactive notebook to zero (in minutes).
- Idle notebooks stop at the time limit that you set.

5.6. CONFIGURING A CUSTOM NOTEBOOK IMAGE

In addition to notebook images provided and supported by Red Hat and independent software vendors (ISVs), you can configure custom notebook images that cater to your project's specific requirements.

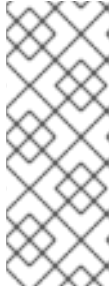
Red Hat supports you in adding custom notebook images to your deployment of OpenShift AI and ensuring that they are available for selection when creating a notebook server. However, Red Hat does not support the contents of your custom notebook image. That is, if your custom notebook image is available for selection during notebook server creation, but does not create a usable notebook server, Red Hat does not provide support to fix your custom notebook image.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- You are assigned the **cluster-admin** role in OpenShift.
- Your custom notebook image exists in an image registry and is accessible.
- You can access the **Settings** → **Notebook images** dashboard navigation menu option.




Procedure


1. From the OpenShift AI dashboard, click **Settings** → **Notebook images**.
The **Notebook images** page appears. Previously imported notebook images are displayed. To enable or disable a previously imported notebook image, on the row containing the relevant notebook image, click the toggle in the **Enable** column.



NOTE

If you have already configured an accelerator identifier for a notebook image, you can specify a recommended accelerator for the notebook image by creating an associated accelerator profile. To do this, click **Create profile** on the row containing the notebook image and complete the relevant fields. If the notebook image does not contain an accelerator identifier, you must manually configure one before creating an associated accelerator profile.

2. Click **Import new image**. Alternatively, if no previously imported images were found, click **Import image**.
The **Import Notebook images** dialog appears.
3. In the **Image location** field, enter the URL of the repository containing the notebook image. For example: **quay.io/my-repo/my-image:tag**, **quay.io/my-repo/my-image@sha256:xxxxxxxxxxxxx**, or **docker.io/my-repo/my-image:tag**.
4. In the **Name** field, enter an appropriate name for the notebook image.
5. Optional: In the **Description** field, enter a description for the notebook image.
6. Optional: From the **Accelerator identifier** list, select an identifier to set its accelerator as recommended with the notebook image. If the notebook image contains only one accelerator identifier, the identifier name displays by default.
7. Optional: Add software to the notebook image. After the import has completed, the software is added to the notebook image's meta-data and displayed on the Jupyter server creation page.
 - a. Click the **Software** tab.
 - b. Click the **Add software** button.
 - c. Click **Edit** ().
 - d. Enter the **Software** name.
 - e. Enter the software **Version**.
 - f. Click **Confirm** () to confirm your entry.
 - g. To add additional software, click **Add software**, complete the relevant fields, and confirm your entry.
8. Optional: Add packages to the notebook images. After the import has completed, the packages are added to the notebook image's meta-data and displayed on the Jupyter server creation page.
 - a. Click the **Packages** tab.
 - b. Click the **Add package** button.
 - c. Click **Edit** ().
 - d. Enter the **Package** name.

- e. Enter the package **Version**.
 - f. Click **Confirm** () to confirm your entry.
 - g. To add an additional package, click **Add package**, complete the relevant fields, and confirm your entry.
9. Click **Import**.

Verification

- The notebook image that you imported is displayed in the table on the **Notebook images** page.
- Your custom notebook image is available for selection on the **Start a notebook server** page in Jupyter.

Additional resources

- [Managing image streams](#)
- [Understanding build configurations](#)

CHAPTER 6. BACKING UP DATA

6.1. BACKING UP STORAGE DATA

It is a best practice to back up the data on your persistent volume claims (PVCs) regularly.

Backing up your data is particularly important before you delete a user and before you uninstall OpenShift AI, as all PVCs are deleted when OpenShift AI is uninstalled.

See the documentation for your cluster platform for more information about backing up your PVCs.

Additional resources

- [Understanding persistent storage](#)

CHAPTER 7. USAGE DATA COLLECTION

Red Hat OpenShift AI administrators can choose whether to allow Red Hat to collect data about OpenShift AI usage in their cluster. Collecting this data allows Red Hat to monitor and improve our software and support. For further details about the data Red Hat collects, see [Usage data collection notice for OpenShift AI](#).

Usage data collection is enabled by default when you install OpenShift AI on your OpenShift cluster except when clusters are installed in a disconnected environment.

See [Disabling usage data collection](#) for instructions on disabling the collection of this data in your cluster. If you have disabled data collection on your cluster, and you want to enable it again, see [Enabling usage data collection](#) for more information.

7.1. USAGE DATA COLLECTION NOTICE FOR OPENSIFT AI

In connection with your use of this Red Hat offering, Red Hat may collect usage data about your use of the software. This data allows Red Hat to monitor the software and to improve Red Hat offerings and support, including identifying, troubleshooting, and responding to issues that impact users.

What information does Red Hat collect?

Tools within the software monitor various metrics and this information is transmitted to Red Hat. Metrics include information such as:

- Information about applications enabled in the product dashboard.
- The deployment sizes used (that is, the CPU and memory resources allocated).
- Information about documentation resources accessed from the product dashboard.
- The name of the notebook images used (that is, Minimal Python, Standard Data Science, and other images.).
- A unique random identifier that generates during the initial user login to associate data to a particular username.
- Usage information about components, features, and extensions.

Third Party Service Providers

Red Hat uses certain third party service providers to collect the telemetry data.

Security

Red Hat employs technical and organizational measures designed to protect the usage data.

Personal Data

Red Hat does not intend to collect personal information. If Red Hat discovers that personal information has been inadvertently received, Red Hat will delete such personal information and treat such personal information in accordance with Red Hat's Privacy Statement. For more information about Red Hat's privacy practices, see Red Hat's [Privacy Statement](#).

Enabling and Disabling Usage Data

You can disable or enable usage data by following the instructions in [Disabling usage data collection](#) or [Enabling usage data collection](#).

7.2. ENABLING USAGE DATA COLLECTION

Red Hat OpenShift AI administrators can choose whether to allow Red Hat to collect data about OpenShift AI usage in their cluster. Usage data collection is enabled by default when you install OpenShift AI on your OpenShift cluster except when clusters are installed in a disconnected environment. If you have disabled data collection previously, you can re-enable it by following these steps.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- You are part of the administrator group for OpenShift AI in OpenShift except when clusters are installed in a disconnected environment.

Procedure

1. From the OpenShift AI dashboard, click **Settings** → **Cluster settings**.
2. Locate the **Usage data collection** section.
3. Select the **Allow collection of usage data** checkbox.
4. Click **Save changes**.

Verification

- A notification is shown when settings are updated: **Settings changes saved**.

Additional resources

- [Usage data collection notice for OpenShift AI](#)

7.3. DISABLING USAGE DATA COLLECTION

Red Hat OpenShift AI administrators can choose whether to allow Red Hat to collect data about OpenShift AI usage in their cluster. Usage data collection is enabled by default when you install OpenShift AI on your OpenShift cluster except when clusters are installed in a disconnected environment.

You can disable data collection by following these steps.

Prerequisites

- You have logged in to Red Hat OpenShift AI.
- You are part of the administrator group for OpenShift AI in OpenShift except when clusters are installed in a disconnected environment.

Procedure

1. From the OpenShift AI dashboard, click **Settings** → **Cluster settings**.
2. Locate the **Usage data collection** section.
3. Clear the **Allow collection of usage data** checkbox.

4. Click **Save changes**.

Verification

- A notification is shown when settings are updated: **Settings changes saved.**

Additional resources

- [Usage data collection notice for OpenShift AI](#)