Red Hat OpenShift AI Self-Managed 2-latest

Upgrading OpenShift AI Self-Managed in a disconnected environment

Upgrade Red Hat OpenShift AI on OpenShift in a disconnected environment

Last Updated: 2024-07-22

Red Hat OpenShift AI Self-Managed 2-latest Upgrading OpenShift AI Self-Managed in a disconnected environment

Upgrade Red Hat OpenShift AI on OpenShift in a disconnected environment

Legal Notice

Abstract

Learn about the Red Hat OpenShift AI Operator upgrade process in a disconnected environment.

Table of Contents

PREFACE 3

CHAPTER 1. OVERVIEW OF UPGRADING OPENSHIFT AI SELF-MANAGED 4

CHAPTER 2. CONFIGURING THE UPGRADE STRATEGY FOR OPENSHIFT AI 6

CHAPTER 3. REQUIREMENTS FOR UPGRADING OPENSHIFT AI 7

CHAPTER 4. ADDING A CA BUNDLE AFTER UPGRADING 9

PREFACE

As a cluster administrator, you can configure either automatic or manual upgrade of the OpenShift AI Operator.

CHAPTER 1. OVERVIEW OF UPGRADING OPENSHIFT AI SELF-MANAGED

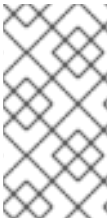
As a cluster administrator, you can configure either automatic or manual upgrades for the Red Hat OpenShift AI Operator in a disconnected environment. A disconnected environment is a network restricted environment where Operator Lifecycle Manager (OLM) cannot access the default OperatorHub and image registries, which require Internet connectivity.

NOTE

For information about upgrading OpenShift AI as self-managed software on your OpenShift cluster in a connected environment, see Upgrading OpenShift AI Self-Managed.

IMPORTANT

Data science pipelines 2.0 contains an installation of Argo Workflows. OpenShift AI does not support direct customer usage of this installation of Argo Workflows. To upgrade to OpenShift AI 2.9 or later with data science pipelines, ensure that no separate installation of Argo Workflows exists on your cluster.



NOTE

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

- If you configure automatic upgrades, when a new version of the Red Hat OpenShift AI Operator is available, and you have updated your mirror registry content, Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without human intervention.
- If you configure manual upgrades, when a new version of the Red Hat OpenShift AI Operator is available and you have updated your mirror registry content, OLM creates an update request. A cluster administrator must manually approve the update request to update the Operator to the new version. See Manually approving a pending Operator upgrade for more information about approving a pending Operator upgrade.
- By default, the Red Hat OpenShift AI Operator follows a sequential update process. This means that if there are several minor versions between the current version and the version that you plan to upgrade to, Operator Lifecycle Manager (OLM) upgrades the Operator to each of the minor versions before it upgrades it to the final, target version. If you configure automatic upgrades, OLM automatically upgrades the Operator to the latest available version, without human intervention. If you configure manual upgrades, a cluster administrator must manually approve each sequential update between the current version and the final, target version. Red Hat supports the current version and three previous minor versions of OpenShift AI Self-Managed. For more information, see the Red Hat OpenShift AI Self-Managed Life Cycle knowledgebase article.
- When you upgrade OpenShift AI, you should complete the Requirements for upgrading OpenShift AI.
- Before you can use an accelerator in OpenShift AI, your instance must have the associated

accelerator profile. If your OpenShift instance has an accelerator, its accelerator profile is preserved after an upgrade. For more information about accelerators, see Working with accelerators.

- Notebook images are integrated into the image stream during the upgrade and subsequently appear in the OpenShift AI dashboard.

NOTE

Notebook images are constructed externally; they are prebuilt images that undergo quarterly changes and they do not change with every OpenShift AI upgrade.

Additional resources

- Operator Lifecycle Manager workflow

CHAPTER 2. CONFIGURING THE UPGRADE STRATEGY FOR OPENSHIFT AI

As a cluster administrator, you can configure either an automatic or manual upgrade strategy for the Red Hat OpenShift AI Operator.

IMPORTANT

By default, the Red Hat OpenShift AI Operator follows a sequential update process. This means that if there are several versions between the current version and the version that you intend to upgrade to, Operator Lifecycle Manager (OLM) upgrades the Operator to each of the intermediate versions before it upgrades it to the final, target version. If you configure automatic upgrades, OLM automatically upgrades the Operator to the *latest* available version, without human intervention. If you configure manual upgrades, a cluster administrator must manually approve each sequential update between the current version and the final, target version.

For information about supported versions, see Red Hat OpenShift AI Life Cycle .

Prerequisites

- You have cluster administrator privileges for your OpenShift cluster.
- The Red Hat OpenShift AI Operator is installed.
- You have mirrored the required container images to a private registry. See Mirroring images to a private registry for a disconnected installation.

Procedure

1. Log in to the OpenShift cluster web console as a cluster administrator.
2. In the **Administrator** perspective, in the left menu, select **Operators → Installed Operators**.
3. Click the **Red Hat OpenShift AI** Operator.
4. Click the **Subscription** tab.
5. Under **Update approval**, click the pencil icon and select one of the following update strategies:
   - **Automatic**: New updates are installed as soon as they become available.
   - **Manual**: A cluster administrator must approve any new update before installation begins.
6. Click **Save**.

Additional resources

- For more information about the subscription channels that are available in version 2 of the Red Hat OpenShift AI Operator, see Installing the Red Hat OpenShift AI Operator .
- For more information about upgrading Operators that have been installed by using OLM, see Updating installed Operators in the OpenShift documentation.

CHAPTER 3. REQUIREMENTS FOR UPGRADING OPENSHIFT AI

This section describes the tasks that you should complete when upgrading OpenShift AI.

Check the components in the **DataScienceCluster** object

When you upgrade Red Hat OpenShift AI, the upgrade process automatically uses the values from the previous **DataScienceCluster** object.

After the upgrade, you should inspect the **DataScienceCluster** object and optionally update the status of any components as described in Updating the installation status of Red Hat OpenShift AI components by using the web console.

Recreate existing pipeline runs

When you upgrade to a newer version, any existing pipeline runs that you created in the previous version continue to refer to the image for the previous version (as expected).

You must delete the pipeline runs (not the pipelines) and create new pipeline runs. The pipeline runs that you create in the newer version correctly refer to the image for the newer version.

For more information on pipeline runs, see Managing pipeline runs.

Upgrading to data science pipelines 2.0

Previously, data science pipelines in OpenShift AI were based on KubeFlow Pipelines v1. It is no longer possible to deploy, view, or edit the details of pipelines that are based on data science pipelines 1.0 from the dashboard in OpenShift AI 2-latest. If you already use data science pipelines, Red Hat recommends that you stay on OpenShift AI 2.8 until full feature parity in data science pipelines 2.0 has been delivered in a stable OpenShift AI release and you are ready to migrate to the new pipeline solution.

Data science pipelines 2.0 contains an installation of Argo Workflows. OpenShift AI does not support direct customer usage of this installation of Argo Workflows. To install or upgrade to OpenShift AI 2.9 or later with data science pipelines 2.0, ensure that there is no existing installation of Argo Workflows on your cluster.

If you want to use existing pipelines and workbenches with data science pipelines 2.0 after upgrading to OpenShift AI 2-latest, you must update your workbenches to use the 2024.1 notebook image version and then manually migrate your pipelines from data science pipelines 1.0 to 2.0. For more information, see Upgrading to data science pipelines 2.0 .

Address KServe requirements

For the KServe component, which is used by the single-model serving platform to serve large models, you must meet the following requirements:

- To fully install and use KServe, you must also install Operators for Red Hat OpenShift Serverless and Red Hat OpenShift Service Mesh and perform additional configuration. For more information, see Serving large models.
- If you want to add an authorization provider for the single-model serving platform, you must install the **Red Hat - Authorino** Operator. For information, see Adding an authorization provider for the single-model serving platform.

- If you have *not* enabled the KServe component (that is, you set the value of the **managementState** field to **Removed** in the **DataScienceCluster** object), you must also disable the dependent Service Mesh component to avoid errors. See Disabling KServe dependencies.

CHAPTER 4. ADDING A CA BUNDLE AFTER UPGRADING

Red Hat OpenShift AI 2-latest provides support for using self-signed certificates. If you have upgraded from OpenShift AI 2.7 or earlier versions, you can add self-signed certificates to the OpenShift AI deployments and Data Science Projects in your cluster.

There are two ways to add a Certificate Authority (CA) bundle to OpenShift AI. You can use one or both of these methods:

- For OpenShift clusters that rely on self-signed certificates, you can add those self-signed certificates to a cluster-wide Certificate Authority (CA) bundle (**ca-bundle.crt**) and use the CA bundle in Red Hat OpenShift AI.
- You can use self-signed certificates in a custom CA bundle (**odh-ca-bundle.crt**) that is separate from the cluster-wide bundle.

For more information, see Working with certificates.

Prerequisites

- You have admin access to the **DSCInitialization** resources in the OpenShift cluster.
- You installed the OpenShift command line interface (**oc**) as described in Get Started with the CLI.
- You upgraded Red Hat OpenShift AI from version 2.7 or earlier. If you are working in a new installation of Red Hat OpenShift AI, see Adding a CA bundle.

Procedure

1. Log in to the OpenShift as a cluster administrator.
2. Click **Operators → Installed Operators** and then click the Red Hat OpenShift AI Operator.
3. Click the **DSC Initialization** tab.
4. Click the **default-dsci** object.
5. Click the **YAML** tab.
6. Add the following to the **spec** section, setting the **managementState** field to **Managed**:

```
spec:
  trustedCABundle:
    managementState: Managed
    customCABundle: ""
```

7. If you want to use self-signed certificates added to a cluster-wide CA bundle, log in to the OpenShift as a cluster administrator and follow the steps as described in Configuring the cluster-wide proxy during installation.
8. If you want to use self-signed certificates in a custom CA bundle that is separate from the cluster-wide bundle, follow these steps:
    a. Add the custom certificate to the **customCABundle** field of the **default-dsci** object, as shown in the following example:

```
spec:
  trustedCABundle:
    managementState: Managed
    customCABundle: |
      -----BEGIN CERTIFICATE-----
      examplebundle123
      -----END CERTIFICATE-----
```

b. Click **Save**.
The Red Hat OpenShift AI Operator creates an **odh-trusted-ca-bundle** ConfigMap containing the certificates in all new and existing non-reserved namespaces.

Verification

- If you are using a cluster-wide CA bundle, run the following command to verify that all non-reserved namespaces contain the **odh-trusted-ca-bundle** ConfigMap:

```
$ oc get configmaps --all-namespaces -l app.kubernetes.io/part-of=opendatahub-operator |
grep odh-trusted-ca-bundle
```

- If you are using a custom CA bundle, run the following command to verify that a non-reserved namespace contains the **odh-trusted-ca-bundle** ConfigMap and that the ConfigMap contains your **customCABundle** value. In the following command, *example-namespace* is the non-reserved namespace and *examplebundle123* is the customCABundle value.

```
$ oc get configmap odh-trusted-ca-bundle -n example-namespace -o yaml | grep
examplebundle123
```