Red Hat OpenShift AI Cloud Service 1

Upgrading OpenShift AI Cloud Service

Upgrade OpenShift AI on an OpenShift Dedicated or Red Hat OpenShift Service on AWS (ROSA) cluster

Last Updated: 2024-03-21

Red Hat OpenShift AI Cloud Service 1 Upgrading OpenShift AI Cloud Service

Upgrade OpenShift AI on an OpenShift Dedicated or Red Hat OpenShift Service on AWS (ROSA) cluster

Legal Notice

Abstract

Upgrade OpenShift AI on an OpenShift Dedicated or Red Hat OpenShift Service on AWS (ROSA) cluster.

Table of Contents

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

Red Hat OpenShift AI is automatically updated as new release or versions become available. Currently, no administrator action is necessary to trigger the process.

When an OpenShift AI upgrade occurs, you should complete the Requirements for upgrading OpenShift AI.

Notes:

- If the upgrade was from version 1 of OpenShift AI (previously OpenShift Data Science), follow the guidelines in Cleaning up unused resources from version 1 of Red Hat OpenShift AI (OpenShift Data Science).

- Before you can use an accelerator in OpenShift AI, your instance must have the associated accelerator profile. If your OpenShift cluster instance has an accelerator, its accelerator profile is preserved after the 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. Notebook images are constructed externally; they are prebuilt images that undergo quarterly changes and they do not change with every OpenShift AI upgrade.

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.

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 upgrading Operators that have been installed by using OLM, see Updating installed Operators in OpenShift Dedicated or Updating installed Operators in Red Hat OpenShift Service on AWS (ROSA)

CHAPTER 3. REQUIREMENTS FOR UPGRADING OPENSHIFT AI

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

Check the components in theDataScienceCluster 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.

Address KServe requirements

For KServe (single-model serving platform), you must meet these requirements:

- Install dependent Operators, including the Red Hat OpenShift Serverless and Red Hat OpenShift Service Mesh Operators. For more information, see Serving large models.
- After the upgrade, you must inspect the default **DataScienceCluster** object and verify that the value of the **managementState** field for the **kserve** component is **Managed**.
- In Red Hat OpenShift AI version 2.4, the KServe component is a Limited Availability feature. If you enabled the **kserve** component and created models in version 2.4, then after you upgrade to version 2.5, you must update some OpenShift AI resources as follows:

  1. Log in as an administrator to your OpenShift cluster where OpenShift AI 2.5 is installed:

     ```
     $ oc login
     ```

  2. Update the **DSC Initialization** resource:

     ```
     $ oc patch $(oc get dsci -A -oname) --type='json' -p='[{"op": "replace", "path":
     "/spec/serviceMesh/managementState", "value":"Unmanaged"}]'
     ```

  3. Update the **Data Science Cluster** resource:

     ```
     $ oc patch $(oc get dsc -A -oname) --type='json' -p='[{"op": "replace", "path":
     "/spec/components/kserve/serving/managementState", "value":"Unmanaged"}]'
     ```

  4. Update the **InferenceServices** CRD:

```
$ oc patch crd inferenceservices.serving.kserve.io --type=json -p='[{"op": "remove",
"path": "/spec/conversion"}]'
```

5. Optionally, restart the Operator pod.
   For more information about these configurations, see Installing KServe.

- If you deployed a model by using KServe in OpenShift AI version 2.4, when you upgrade to version 2.5 the model does not automatically appear in the OpenShift AI dashboard. To update the dashboard view, redeploy the model by using the OpenShift AI dashboard.

CHAPTER 4. CLEANING UP UNUSED RESOURCES FROM VERSION 1 OF RED HAT OPENSHIFT AI (OPENSHIFT DATA SCIENCE)

Version 1 of OpenShift AI (previously OpenShift Data Science) created a set of Kubeflow Deployment Definition (that is, **KfDef**) custom resource instances on your OpenShift cluster for various components of OpenShift AI. When you upgrade to version 2, these resources are no longer used and require manual removal from your cluster. The following procedures shows how to remove unused **KfDef** instances from your cluster by using both the OpenShift command-line interface (CLI) and the web console.

4.1. REMOVING UNUSED RESOURCES BY USING THE CLI

The following procedure shows how to remove unused **KfDef** instances from the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects in your OpenShift cluster by using the OpenShift command-line interface (CLI). These resources become unused after you upgrade from version 1 to version 2 of OpenShift AI.

Prerequisites

- You upgraded from version 1 to version 2 of OpenShift AI.
- You have cluster administrator privileges for your OpenShift cluster.

Procedure

1. Open a new terminal window.
2. In the OpenShift command-line interface (CLI), log in to your on your OpenShift cluster as a cluster administrator, as shown in the following example:

   ```
   $ oc login <openshift_cluster_url> -u system:admin
   ```

3. Delete any **KfDef** instances that exist in the **redhat-ods-applications** project.

   ```
   $ oc delete kfdef --all -n redhat-ods-applications --ignore-not-found || true
   ```

   For any **KfDef** instance that is deleted, the output is similar to the following example:

   ```
   kfdef.kfdef.apps.kubeflow.org "rhods-dashboard" deleted
   ```

   #### TIP

   If deletion of a **KfDef** instance fails to finish, you can force deletion of the object using the information in the "Force individual object removal when it has finalizers" section of the following Red Hat solution article: https://access.redhat.com/solutions/4165791.

4. Delete any **KfDef** instances in the **redhat-ods-monitoring** and **rhods-notebooks** projects by entering the following commands:

   ```
   $ oc delete kfdef --all -n redhat-ods-monitoring --ignore-not-found || true
   $ oc delete kfdef --all -n rhods-notebooks --ignore-not-found || true
   ```

Verification

- Check whether all **KfDef** instances have been removed from the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects.

  ```
  $ oc get kfdef --all-namespaces
  ```

  Verify that you see no **KfDef** instances listed in the **redhat-ods-applications**, **redhat-ods-monitoring**, or **rhods-notebooks** projects.

4.2. REMOVING UNUSED RESOURCES BY USING THE WEB CONSOLE

The following procedure shows how to remove unused **KfDef** instances from the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects in your OpenShift cluster by using the OpenShift web console. These resources become unused after you upgrade from version 1 to version 2 of OpenShift AI.

Prerequisites

- You upgraded from version 1 to version 2 of OpenShift AI.
- You have cluster administrator privileges for your OpenShift Dedicated cluster.

Procedure

1. Log in to the OpenShift web console as a cluster administrator.
2. In the web console, click **Administration** → **CustomResourceDefinitions**.
3. On the **CustomResourceDefinitions** page, click the **KfDef** custom resource definition (CRD).
4. Click the **Instances** tab.
   The page shows all **KfDef** instances on the cluster.
5. Take note of any **KfDef** instances that exist in the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects. These are the projects that you will clean up in the remainder of this procedure.
6. To delete a **KfDef** instance from the **redhat-ods-applications**, **redhat-ods-monitoring**, or **rhods-notebooks** project, click the action menu (⋮) beside the instance and select **Delete KfDef** from the list.
7. To confirm deletion of the instance, click **Delete**.

   #### TIP

   If deletion of a **KfDef** instance fails to finish, you can force deletion of the object using the information in the "Force individual object removal when it has finalizers" section of the following Red Hat solution article: https://access.redhat.com/solutions/4165791.

8. Repeat the preceding steps to delete all remaining **KfDef** instances that you see in the **redhat-ods-applications**, **redhat-ods-monitoring**, and **rhods-notebooks** projects.

CHAPTER 5. UPDATING THE INSTALLATION STATUS OF RED HAT OPENSHIFT AI COMPONENTS BY USING THE WEB CONSOLE

The following procedure shows how to use the OpenShift web console to update the installation status of components of Red Hat OpenShift AI on your OpenShift cluster.

> **IMPORTANT**
>
> When your OpenShift AI version upgrades from a previous minor version, the upgrade process uses the settings from the previous **DataScienceCluster** object.
>
> The following procedure describes how to edit the **DataScienceCluster** object:
>
> - Change the installation status of the existing Red Hat OpenShift AI components
> - Add additional components to the **DataScienceCluster** object that were not available in the previous version of OpenShift AI.

Prerequisites

- To support the KServe component, you installed dependent Operators, including the Red Hat OpenShift Serverless and Red Hat OpenShift Service Mesh Operators. For more information, see Serving large models.
- Red Hat OpenShift AI is installed as an Add-on to your Red Hat OpenShift cluster.
- You have cluster administrator privileges for your OpenShift cluster.

Procedure

1. Log in to the OpenShift web console as a cluster administrator.
2. In the web console, click **Operators → Installed Operators** and then click the Red Hat OpenShift AI Operator.
3. Click the **Data Science Cluster** tab.
4. On the **DataScienceClusters** page, click the **default** object.
5. Click the **YAML** tab.
   An embedded YAML editor opens showing the custom resource (CR) file for the **DataScienceCluster** object.
6. In the **spec.components** section of the CR, for each OpenShift AI component shown, set the value of the **managementState** field to either **Managed** or **Removed**. These values are defined as follows:

   > **NOTE**
   >
   > If a component shows with the **component-name: {}** format in the **spec.components** section of the CR, the component is not installed.

   Managed

The Operator actively manages the component, installs it, and tries to keep it active. The Operator will upgrade the component only if it is safe to do so.

Removed

The Operator actively manages the component but does not install it. If the component is already installed, the Operator will try to remove it.

IMPORTANT

- To learn how to install the KServe component, which is used by the single model serving platform to serve large models, see Serving large models.
- If they are not already present in the CR file, you can install the CodeFlare and KubeRay features by adding components called **codeflare** and **ray** to the **spec.components** section of the CR and setting the **managementState** field for the components to **Managed**.
- The CodeFlare and KubeRay components are Technology Preview features only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.
- To learn how to configure the distributed workloads feature that uses the CodeFlare and KubeRay components, see Configuring distributed workloads.

7. Click **Save**.
   For any components that you updated, OpenShift AI initiates a rollout that affects all pods to use the updated image.

Verification

- Confirm that there is a running pod for each component:
  1. In the OpenShift Dedicated web console, click **Workloads** → **Pods**.
  2. In the **Project** list at the top of the page, select **redhat-ods-applications**.
  3. In the applications namespace, confirm that there are running pods for each of the OpenShift AI components that you installed.
- Confirm the status of all installed components:
  1. In the OpenShift Dedicated web console, click **Operators** → **Installed Operators**.
  2. Click the Red Hat OpenShift AI Operator.
  3. Click the **Data Science Cluster** tab and select the **DataScienceCluster** object called **default-dsc**.
  4. Select the **YAML** tab.

5. In the **installedComponents** section, confirm that the components you installed have a status value of **true**.

NOTE

If a component shows with the **component-name: {}** format in the **spec.components** section of the CR, the component is not installed.

CHAPTER 6. ADDING A CA BUNDLE AFTER UPGRADING

Red Hat OpenShift AI 1 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 Dedicated 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 Dedicated cluster.
- You installed the OpenShift command line interface (**oc**) as described in Get Started with the CLI.
- You upgraded Red Hat OpenShift AI. 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 Dedicated 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 Dedicated 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
  ```