Red Hat OpenShift AI Self-Managed 2.8

Release notes

Features, enhancements, resolved issues, and known issues associated with this release

Last Updated: 2024-04-09

Red Hat OpenShift AI Self-Managed 2.8 Release notes

Features, enhancements, resolved issues, and known issues associated with this release

Legal Notice

Abstract

These release notes provide an overview of new features, enhancements, resolved issues, and known issues in version 2.8.1 of Red Hat OpenShift AI.

Table of Contents

CHAPTER 1. OVERVIEW OF OPENSHIFT AI

Red Hat OpenShift AI is a platform for data scientists and developers of artificial intelligence and machine learning (AI/ML) applications.

OpenShift AI provides an environment to develop, train, serve, test, and monitor AI/ML models and applications on-premises or in the cloud.

For data scientists, OpenShift AI includes Jupyter and a collection of default notebook images optimized with the tools and libraries required for model development, and the TensorFlow and PyTorch frameworks. Deploy and host your models, integrate models into external applications, and export models to host them in any hybrid cloud environment. You can also accelerate your data science experiments through the use of graphics processing units (GPUs) and Habana Gaudi devices.

For administrators, OpenShift AI enables data science workloads in an existing Red Hat OpenShift or ROSA environment. Manage users with your existing OpenShift identity provider, and manage the resources available to notebook servers to ensure data scientists have what they require to create, train, and host models. Use accelerators to reduce costs and allow your data scientists to enhance the performance of their end-to-end data science workflows using graphics processing units (GPUs) and Habana Gaudi devices.

OpenShift AI offers two distributions:

- A **managed cloud service add-on** for Red Hat OpenShift Dedicated (with a Customer Cloud Subscription for AWS or GCP) or for Red Hat OpenShift Service on Amazon Web Services (ROSA).
  For information about OpenShift AI on a Red Hat managed environment, see Product Documentation for Red Hat OpenShift AI.

- **Self-managed software** that you can install on-premise or on the public cloud in a self-managed environment, such as OpenShift Container Platform.
  For information about OpenShift AI as self-managed software on your OpenShift cluster in a connected or a disconnected environment, see Product Documentation for Red Hat OpenShift AI Self-Managed.

For information about OpenShift AI supported software platforms, components, and dependencies, see Supported configurations.

CHAPTER 2. NEW FEATURES AND ENHANCEMENTS

This section describes new features and enhancements in Red Hat OpenShift AI 2.8.

2.1. NEW FEATURES

Support for self-signed certificates

You can now use self-signed certificates in your Red Hat OpenShift AI deployments and Data Science Projects in an OpenShift Container Platform cluster.

Some OpenShift AI components have additional options or required configuration for self-signed certificates, as described in Working with certificates (for disconnected environments, see Working with certificates).

2.2. ENHANCEMENTS

Upgraded OpenVINO Model Server

The OpenVINO Model Server has been upgraded to version 2023.3. For information on the changes and enhancements, see OpenVINO™ Model Server 2023.3 .

Support for gRPC protocol on single-model serving platform

The single-model serving platform now supports the gRPC API protocol in addition to REST. This support means that when you add a custom model serving runtime to the platform, you can specify which protocol the runtime uses.

Extended support with new release channels

Starting with OpenShift AI 2.8, Red Hat provides production updates and support for the Red Hat OpenShift AI Operator in two new channels, in addition to the **fast**, **stable**, and **alpha** channels:

- The **stable-2.8** channel allows you to stay on the latest 2.8.x release with full support for seven months.
- The **eus-2.8** channel allows you to stay on the latest 2.8.x release with full support for seven months, followed by Extended Update Support for eleven months.

For more information about subscription channels, see Installing the Red Hat OpenShift AI Operator .

CHAPTER 3. TECHNOLOGY PREVIEW FEATURES

IMPORTANT

This section describes Technology Preview features in Red Hat OpenShift AI 2.8. 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 .

Distributed workloads

Distributed workloads enable data scientists to use multiple cluster nodes in parallel for faster, more efficient data processing and model training. The CodeFlare framework simplifies task orchestration and monitoring, and offers seamless integration for automated resource scaling and optimal node utilization with advanced GPU support.

Designed for data scientists, the CodeFlare framework enables direct workload configuration from Jupyter Notebooks or Python code, ensuring a low barrier of adoption, and streamlined, uninterrupted workflows. Distributed workloads significantly reduce task completion time, and enable the use of larger datasets and more complex models. The distributed workloads feature is currently available in Red Hat OpenShift AI 2.8 as a Technology Preview feature. This feature was first introduced in OpenShift AI 2.4.

code-server notebook image

Red Hat OpenShift AI now includes the code-server notebook image. See code-server in GitHub for more information.

With the code-server notebook image, you can customize your notebook environment to meet your needs using a variety of extensions to add new languages, themes, debuggers, and connect to additional services. Enhance the efficiency of your data science work with syntax highlighting, auto-indentation, and bracket matching.

NOTE

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

The code-server notebook image is currently available in Red Hat OpenShift AI 2.8 as a Technology Preview feature. This feature was first introduced in OpenShift AI 2.6.

CHAPTER 4. SUPPORT REMOVALS

This section describes major changes in support for user-facing features in Red Hat OpenShift AI.

4.1. EMBEDDED SUBSCRIPTION CHANNEL DEPRECATED

Starting with OpenShift AI 2.8, the **embedded** subscription channel has been removed. You can no longer select the **embedded** channel for a new installation of the Operator. For more information about subscription channels, see Installing the Red Hat OpenShift AI Operator .

4.2. REMOVAL OF BIAS DETECTION (TRUSTYAI)

Starting with OpenShift AI 2.7, the bias detection (TrustyAI) functionality has been removed. If you previously had this functionality enabled, upgrading to OpenShift AI 2.7 or later will remove the feature. The default TrustyAI notebook image remains supported.

4.3. UPCOMING DEPRECATION OF DATA SCIENCE PIPELINES V1

Currently, data science pipelines in OpenShift AI are based on Kubeflow Pipelines v1. See Working with data science pipelines for more information.

Data science pipelines in upcoming releases will be based on Kubeflow Pipelines v2, using a different engine. OpenShift AI 2.8 is a stable release that will be supported for 7 months. We recommend that current data science pipeline users stay on OpenShift AI 2.8 until you are ready to migrate to the new pipelines solution.

For a detailed view of the 2.8 release lifecycle, including its full support phase window, see Red Hat OpenShift AI Self-Managed Life Cycle.

4.4. VERSION 1.2 NOTEBOOK CONTAINER IMAGES FOR WORKBENCHES ARE NO LONGER SUPPORTED

When you create a workbench, you specify a notebook container image to use with the workbench. Starting with OpenShift AI 2.5, when you create a new workbench, version 1.2 notebook container images are not available to select. Workbenches that are already running with a version 1.2 notebook image continue to work normally. However, Red Hat recommends that you update your workbench to use the latest notebook container image.

4.5. BETA SUBSCRIPTION CHANNEL DEPRECATED

Starting with OpenShift AI 2.5, the **beta** subscription channel has been removed. You can no longer select the **beta** channel for a new installation of the Operator. For more information about subscription channels, see Installing the Red Hat OpenShift AI Operator .

CHAPTER 5. RESOLVED ISSUES

The following notable issues are resolved in Red Hat OpenShift AI 2.8.1 and 2.8.

5.1. ISSUES RESOLVED IN RED HAT OPENSHIFT AI 2.8.1

RHOAIENG-4937 (previously documented asRHOAIENG-4572) - Unable to run data science pipelines after install and upgrade in certain circumstances

Previously, you were unable to run data science pipelines after installing or upgrading OpenShift AI in the following circumstances:

- You installed OpenShift AI and you had a valid CA certificate. Within the **default-dsci** object, you changed the **managementState** field for the **trustedCABundle** field to **Removed** post-installation.
- You upgraded OpenShift AI from version 2.6 to version 2.8 and you had a valid CA certificate.
- You upgraded OpenShift AI from version 2.7 to version 2.8 and you had a valid CA certificate.

This issue is now resolved.

RHOAIENG-4327 - Workbenches do not use the self-signed certificates from centrally configured bundle automatically

There are two bundle options to include self-signed certificates in OpenShift AI, **ca-bundle.crt** and **odh-ca-bundle.crt**. Self-signed certificates should apply to workbenches that you create after configuring self-signed certificates centrally. Previously, workbenches did not use the self-signed certificates from the centrally configured bundle automatically and you had to define environment variables that pointed to your certificate path. This issue is now resolved.

RHOAIENG-673 (previously documented as RHODS-12946) - Cannot install from PyPI mirror in disconnected environment or when using private certificates

In disconnected environments, Red Hat OpenShift AI cannot connect to the public-facing PyPI repositories, so you must specify a repository inside your network. Previously, if you were using private TLS certificates and a data science pipeline was configured to install Python packages, the pipeline run would fail. This issue is now resolved.

RHOAIENG-637 (previously documented as RHODS-12904) - Pipeline submitted from Elyra might fail when using private certificate

If you use a private TLS certificate and you submit a pipeline from Elyra, previously the pipeline could fail with a **certificate verify failed** error message. This issue is now resolved.

5.2. ISSUES RESOLVED IN RED HAT OPENSHIFT AI 2.8

RHOAIENG-3355 - OVMS on KServe does not use accelerators correctly

Previously, when you deployed a model using the single-model serving platform and selected the **OpenVINO Model Server** serving runtime, if you requested an accelerator to be attached to your model server, the accelerator hardware was detected but was not used by the model when responding to queries. This issue is now resolved.

RHOAIENG-2869 - Cannot edit existing model framework and model path in a multi-model project

Previously, when you tried to edit a model in a multi-model project using the **Deploy model** dialog, the **Model framework** and **Path** values did not update. This issue is now resolved.

RHOAIENG-2724 - Model deployment fails because fields automatically reset in dialog

Previously, when you deployed a model or edited a deployed model, the **Model servers** and **Model framework** fields in the "Deploy model" dialog might have reset to the default state. The **Deploy** button might have remained enabled even though these mandatory fields no longer contained valid values. This issue is now resolved.

RHOAIENG-2099 - Data science pipeline server fails to deploy in fresh cluster

Previously, when you created a data science pipeline server on a fresh cluster, the user interface remained in a loading state and the pipeline server did not start. This issue is now resolved.

RHOAIENG-1199 (previously documented asODH-DASHBOARD-1928) - Custom serving runtime creation error message is unhelpful

Previously, when you tried to create or edit a custom model-serving runtime and an error occurred, the error message did not indicate the cause of the error. The error messages have been improved.

RHOAIENG-675 (previously documented as RHODS-12906) - Cannot use ModelMesh with object storage that uses private certificates

Previously, when you stored models in an object storage provider that used a private TLS certificate, the model serving pods failed to pull files from the object storage, and the **signed by unknown authority** error message was shown. This issue is now resolved.

RHOAIENG-556 - ServingRuntime for KServe model is created regardless of error

Previously, when you tried to deploy a KServe model and an error occurred, the **InferenceService** custom resource (CR) was still created and the model was shown in the **Data Science Project** page, but the status would always remain unknown. The KServe deploy process has been updated so that the ServingRuntime is not created if an error occurs.

RHOAIENG-548 (previously documented asODH-DASHBOARD-1776) - Error messages when user does not have project administrator permission

Previously, if you did not have administrator permission for a project, you could not access some features, and the error messages did not explain why. For example, when you created a model server in an environment where you only had access to a single namespace, an **Error creating model server** error message appeared. However, the model server is still successfully created. This issue is now resolved.

RHOAIENG-66 - Ray dashboard route deployed by CodeFlare SDK exposes self-signed certs instead of cluster cert

Previously, when you deployed a Ray cluster by using the CodeFlare SDK with the **openshift_oauth=True** option, the resulting route for the Ray cluster was secured by using the **passthrough** method and as a result, the self-signed certificate used by the OAuth proxy was exposed. This issue is now resolved.

RHOAIENG-12 - Cannot access Ray dashboard from some browsers

In some browsers, users of the distributed workloads feature might not have been able to access the Ray dashboard because the browser automatically changed the prefix of the dashboard URL from **http** to **https**. This issue is now resolved.

RHODS-6216 - The ModelMesh oauth-proxy container is intermittently unstable

Previously, ModelMesh pods did not deploy correctly due to a failure of the ModelMesh **oauth-proxy** container. This issue occurred intermittently and only if authentication was enabled in the ModelMesh runtime environment. This issue is now resolved.

CHAPTER 6. KNOWN ISSUES

This section describes known issues in Red Hat OpenShift AI 2.8.1 and any known methods of working around these issues.

RHOAIENG-5067 - Model server metrics page does not load for a model server based on the ModelMesh component

Data science project names that contain capital letters or spaces can cause issues on the model server metrics page for model servers based on the ModelMesh component. The metrics page might not receive data correctly, resulting in a **400 Bad Request** error and preventing the page from loading.

Workaround

In OpenShift Container Platform, change the display names of your data science projects to meet Kubernetes resource name standards: use only lowercase alphanumeric characters and hyphens.

RHOAIENG-5025 - Self-signed certificates do not apply to the first created workbench

After self-signed certificates are configured centrally, the certificates do not apply to the first workbench created in a data science project.

Workaround

For each data science project that contains a workbench, delete the first workbench that was created after configuring self-signed certificates, and then create a new workbench. The self-signed certificates work as expected with the new workbench.

RHOAIENG-4966 - Self-signed certificates in a custom CA bundle might be missing from the**odh-trusted-ca-bundle** configuration map

Sometimes after self-signed certificates are configured in a custom CA bundle, the custom certificate is missing from the **odh-trusted-ca-bundle** ConfigMap, or the non-reserved namespaces do not contain the **odh-trusted-ca-bundle** ConfigMap when the ConfigMap is set to **managed**. These issues rarely occur.

Workaround

Restart the Red Hat OpenShift AI Operator pod.

RHOAIENG-4524 - BuildConfig definitions for RStudio images contain occurrences of incorrect branch

The BuildConfig definitions for the **RStudio** and **CUDA - RStudio** workbench images point to the wrong branch in OpenShift AI. The BuildConfig definitions incorrectly point to the **main** branch instead of the **rhoai-2.8** branch.

Workaround

To use the **RStudio** and **CUDA - RStudio** workbench images in OpenShift AI, follow the steps in the Branch workaround for RStudio image BuildConfig definition knowledgebase article.

RHOAIENG-4497 - Models on the multi-model serving platform with self-signed certificates stop working after upgrading to 2.8

In previous versions, if you wanted to use a self-signed certificate when serving models on the multi-model serving platform, you had to manually configure the **storage-config** secret used by your data connection to specify a certificate authority (CA) bundle.

If you upgrade a previous version of OpenShift AI that used that workaround to the latest version, the multi-model serving platform can no longer serve models.

Workaround

To use a self-signed certificate with both the multi- and single-model serving platforms, follow the steps in Adding a CA bundle.

RHOAIENG-4430 - CA Bundle does not work for KServe without a data connection

If you have installed a certificate authority (CA) bundle on your OpenShift cluster to use self-signed certificates and then use the OpenShift AI dashboard to create a data connection to serve a model, OpenShift AI automatically stores the certificate in a secret called **storage-config**. However, if you bypass the OpenShift AI dashboard and configure the underlying **InferenceService** resource to specify a different secret name or a service account, OpenShift AI fails to validate SSL connections to the model and the model status includes **[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate**.

Workaround

Use the OpenShift AI dashboard to create the data connection for your model. Do not manually modify the **InferenceService** resource to specify a different secret name or a service account.

RHOAIENG-4252 - Data science pipeline server deletion process fails to remove **ScheduledWorkFlow** resource

The pipeline server deletion process does not remove the **ScheduledWorkFlow** resource. As a result, new DataSciencePipelinesApplications (DSPAs) do not recognize the redundant **ScheduledWorkFlow** resource.

Workaround

1. Delete the pipeline server. For more information, see Deleting a pipeline server.
2. In the OpenShift command-line interface (CLI), log in to your cluster as a cluster administrator and perform the following command to delete the redundant **ScheduledWorkFlow** resource.

```
$ oc -n <data science project name> delete scheduledworkflows --all
```

RHOAIENG-4240 - Jobs fail to submit to Ray cluster in unsecured environment

When running distributed data science workloads from notebooks in an unsecured OpenShift cluster, a **ConnectionError: Failed to connect to Ray** error message might be shown.

Workaround

In the **ClusterConfiguration** section of the notebook, set the **openshift_oauth** option to **True**.

RHOAIENG-3981 - In unsecured environment, the functionality to wait for Ray cluster to be ready gets stuck

When running distributed data science workloads from notebooks in an unsecured OpenShift cluster, the functionality to wait for the Ray cluster to be ready before proceeding (**cluster.wait_ready()**) gets stuck even when the Ray cluster is ready.

Workaround

Perform one of the following actions:

- In the **ClusterConfiguration** section of the notebook, set the **openshift_oauth** option to **True**.
- Instead of using the **cluster.wait_ready()**, functionality, you can manually check the Ray cluster availability by opening the Ray cluster Route URL. When the Ray dashboard is available on the URL, then the cluster is ready.

RHOAIENG-3963 - Unnecessary managed resource warning

When you edit and save the **OdhDashboardConfig** custom resource for the **redhat-ods-applications** project, the system incorrectly displays the following **Managed resource** warning message.

> This resource is managed by DSC default-doc and any modifications may be overwritten. Edit the managing resource to preserve changes.

You can safely ignore this message.

Workaround

Click **Save** to close the warning message and apply your edits.

RHOAIENG-1825 - After setting up self-signed certificates, executing pipelines might fail with workbenches that contain Elyra

After configuring self-signed certificates centrally, executing pipelines with workbenches that contain Elyra might fail.

Workaround

See the following knowledgebase articles for workaround steps:

- Workbench workaround for executing a pipeline using Elyra
- Workbench workaround for an object storage connection with a self-signed certificate
- How to execute a pipeline from a Jupyter notebook in a disconnected environment

When you deploy a model using the single-model serving platform and select the **OpenVINO Model Server** serving runtime, if you request an accelerator to be attached to your model server, the accelerator hardware is detected but is not used by the model when responding to queries. The queries are computed by using the CPUs only.

Workaround

To configure OVMS to use accelerators in preference to CPUs, update your OVMS runtime template to add **--target_device AUTO** to the CLI options.

RHOAIENG-3134 - OVMS supports different model frameworks in single- and multi-model serving platforms

When you deploy a model using the single-model serving platform and select the **OpenVINO Model Server** runtime, you see additional frameworks in the **Model framework (name - version)** list.

Workaround

None.

When you use the OpenVINO Model Server (OVMS) runtime to deploy a model on the single model serving platform (which uses KServe), there is a mismatch between the directory layout expected by OVMS and that of the model-pulling logic used by KServe. Specifically, OVMS requires the model files to be in the /**<mnt>**/**models**/**1**/ directory, while KServe places them in the /**<mnt>**/**models**/ directory.

Workaround

Perform the following actions:

1. In your S3-compatible storage bucket, place your model files in a directory called **1**/, for example, /**<s3_storage_bucket>**/**models**/**1**/**<model_files>**.

2. To use the OVMS runtime to deploy a model on the single model serving platform, choose one of the following options to specify the path to your model files:

   - If you are using the OpenShift AI dashboard to deploy your model, in the **Path** field for your data connection, use the /**<s3_storage_bucket>**/**models**/ format to specify the path to your model files. Do not specify the **1**/ directory as part of the path.

   - If you are creating your own **InferenceService** custom resource to deploy your model, configure the value of the **storageURI** field as /**<s3_storage_bucket>**/**models**/. Do not specify the **1**/ directory as part of the path.

KServe pulls model files from the subdirectory in the path that you specified. In this case, KServe correctly pulls model files from the /**<s3_storage_bucket>**/**models**/**1**/ directory in your S3-compatible storage.

RHOAIENG-3018 - OVMS on KServe does not expose the correct endpoint in the dashboard

When you use the OpenVINO Model Server (OVMS) runtime to deploy a model on the single-model serving platform, the URL shown in the **Inference endpoint** field for the deployed model is not complete. To send queries to the model, you must add the **/v2/models/_<model-name>_/infer** string to the end of the URL. Replace **_<model-name>_** with the name of your deployed model.

Workaround

None.

RHOAIENG-2542 - Inference service pod does not always get an Istio sidecar

When you deploy a model using the single model serving platform (which uses KServe), the **istio-proxy** container might be missing in the resulting pod, even if the inference service has the **sidecar.istio.io/inject=true** annotation.

In OpenShift AI 2.7, the missing **istio-proxy** container might not present a problem. However, if the pod experiences connectivity issues, they might be caused by the missing container.

Workaround

Delete the faulty pod. OpenShift AI automatically creates a new pod, which should have the missing container.

RHOAIENG-3378 - Internal Image Registry is an undeclared hard dependency for Jupyter notebooks spawn process

Before you can start OpenShift AI notebooks and workbenches, you must first enable the internal, integrated container image registry in OpenShift Container Platform. Attempts to start notebooks or workbenches without first enabling the image registry will fail with an "InvalidImageName" error.

You can confirm whether the image registry is enabled for a cluster by using the following command:

```
$ oc get pods -n openshift-image-registry
```

Workaround

Enable the internal, integrated container image registry in OpenShift Container Platform.

See Image Registry Operator in OpenShift Container Platform for more information about how to set up and configure the image registry.

When you try to edit a model in a multi-model project using the **Deploy model** dialog, the **Model framework** and **Path** values do not update.

Workaround

None available.

When you create a second model server in a project where one server is using token authentication, and the other server does not use authentication, the deployment of the second model might fail to start.

Workaround

None available.

When you deploy a model or edit a deployed model, the **Model servers** and **Model framework** fields in the "Deploy model" dialog might reset to the default state. The **Deploy** button might remain enabled even though these mandatory fields no longer contain valid values.

If you click **Deploy** when the **Model servers** and **Model framework** fields are not set, the model deployment pods are not created.

Workaround

None available.

RHOAIENG-2620 - Unable to create duplicate bias metrics from existing bias metrics

You can’t duplicate existing bias metrics.

Workaround

1. In the left menu of the OpenShift AI dashboard, click **Model Serving**.
2. On the **Deployed models** page, click the name of the model with the bias metric that you want to duplicate.
3. In the metrics page for the model, click the **Model bias** tab.
4. Click the action menu (⋮) next to the metric that you want to copy and then click **Duplicate**.
5. The **Configure bias metrics** dialog will open with prepopulated values for the bias configuration. For each of the **Privileged value**, **Unprivileged value** and **Output value** fields, cut the value and then paste it back in.
   Note: Do not copy and paste these values.
6. Click **Configure**.

The **Average response time** server metric graph shows multiple lines if the ModelMesh pod is restarted.

Workaround

None available.

RHOAIENG-2585 - UI does not display an error/warning when UWM is not enabled in the cluster

Red Hat OpenShift AI does not correctly warn users if User Workload Monitoring (UWM) is **disabled** in the cluster. UWM is necessary for the correct functionality of model metrics.

Workaround

Manually ensure that UWM is enabled in your cluster, as described in Enabling monitoring for user-defined projects.

RHOAIENG-2555 - Model framework selector does not reset when changing Serving Runtime in form

When you use the **Deploy model** dialog to deploy a model on the single model serving platform, if you select a runtime and a supported framework, but then switch to a different runtime, the existing framework selection is not reset. This means that it is possible to deploy the model with a framework that is not supported for the selected runtime.

Workaround

While deploying a model, if you change your selected runtime, click the **Select a framework** list again and select a supported framework.

The Prometheus target for the TrustyAI controller manager is down due to a mismatch with the endpoint's port. Alerts for TrustyAI will fire if the controller deployment pod is down.

Workaround

None available.

If you upgrade the Red Hat OpenShift AI operator from version 2.4 to 2.5, and then update the operator to version 2.6, 2.7, or 2.8, all components related to hardware resource-consuming model monitoring are removed from the cluster. Some residual model-monitoring resources, which do not consume hardware resources, will still be present.

Workaround

To delete these resources, execute the following **oc delete** commands with cluster-admin privileges:

```
$ oc delete service rhods-model-monitoring -n redhat-ods-monitoring
$ oc delete service prometheus-operated -n redhat-ods-monitoring
$ oc delete sa prometheus-custom -n redhat-ods-monitoring
$ oc delete sa rhods-prometheus-operator -n redhat-ods-monitoring
$ oc delete prometheus rhods-model-monitoring -n redhat-ods-monitoring
$ oc delete route rhods-model-monitoring -n redhat-ods-monitoring
```

RHOAIENG-2468 - Services in the same project as KServe might become inaccessible in OpenShift

If you deploy a non-OpenShift AI service in a data science project that contains models deployed on the single model serving platform (which uses KServe), the accessibility of the service might be affected by the network configuration of your OpenShift cluster. This is particularly likely if you are using the OVN-Kubernetes network plugin in combination with host network namespaces.

Workaround

Perform one of the following actions:

- Deploy the service in another data science project that does not contain models deployed on the single model serving platform. Or, deploy the service in another OpenShift project.

- In the data science project where the service is, add a network policy to accept ingress traffic to your application pods, as shown in the following example:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-ingress-to-myapp
spec:
  podSelector:
    matchLabels:
      app: myapp
  ingress:
    - {}
```

RHOAIENG-2312 - Importing numpy fails in code-server workbench

Importing **numpy** in your code-server workbench fails.

Workaround

1. In your code-server workbench, from the **Activity bar**, select the menu icon(☰) > **View** > **Command Palette** to open the Command Palette.
   In Firefox, you can use the F1 keyboard shortcut to open the command palette.

2. Enter **python: s**.

3. From the drop-down list, select the **Python: Select interpreter** action.

4. In the **Select Interpreter** dialog, select **Enter interpreter path…**.

5. Enter **/opt/app-root/bin/python3** as the interpreter path and press **Enter**.

6. From the drop-down list, select the new Python interpreter.

7. Confirm that the new interpreter (**app-root**) appears on the **Status bar**. The selected interpreter persists if the workbench is stopped and started again, so the workaround should need to be performed only once for each workbench.

You can't edit the deployment settings (for example, the number of replicas) of a model you deployed with a single-model platform.

Workaround

None available.

RHOAIENG-2269 - (Single-model) Dashboard fails to display the correct number of model replicas

On a single-model platform, the **Models and model servers** section of a data science project does not show the correct number of model replicas.

Workaround

Check the number of replicas using the following CLI command:

```
$ oc -n <project_resource_name> get pods --selector
serving.kserve.io/inferenceservice=<model_resource_name>
```

You can find your **<project_resource_name>** and **<model_resource_name>** values in the OpenShift AI dashboard.

You can also check the number of model replicas from the OpenShift Container Platform web console, under **Workloads** > **Pods**.

On the **Endpoint performance** tab of the model metrics screen, if you set the **Refresh interval** to 15 seconds and the **Time range** to 1 hour, the graph results change continuously.

Workaround

None available.

RHOAIENG-2183 - Endpoint performance graphs might show incorrect labels

In the **Endpoint performance** tab of the model metrics screen, the graph tooltip might show incorrect labels.

Workaround

None available.

RHOAIENG-1919 - Model Serving page fails to fetch or report the model route URL soon after its deployment

When deploying a model from the OpenShift AI dashboard, the system displays the following warning message while the **Status** column of your model indicates success with an **OK**/green checkmark.

```
Failed to get endpoint for this deployed model. routes.rout.openshift.io"<model_name>" not found
```

Workaround

Refresh your browser page.

The Knative **net-istio-controller** pod (which is a dependency for KServe) might continuously crash due to an out-of-memory (OOM) error.

Workaround

In the custom resource (CR) for your KnativeServing instance, add an **ENABLE_SECRET_INFORMER_FILTERING_BY_CERT_UID=true** annotation to inject an environment variable to the **net-istio-controller** pod. Injecting this environment variable reduces the number of secrets that the **net-istio-controller** watches and loads into memory.

For more information about this configuration, see Creating a Knative Serving instance .

The Red Hat OpenShift AI Add-on uninstall does not delete OpenShift AI components after being triggered via OCM APIs.

Workaround

Manually delete the remaining OpenShift AI resources as follows:

1. Delete the **DataScienceCluster** CR.

1. Delete the **DataScienceCluster** CR.
2. Wait until all pods are deleted from the **redhat-ods-applications** namespace.
3. If Serverless was set to **Managed** in the **DataScienceCluster** CR, wait until all pods are deleted from the **knative-serving** namespace.
4. Delete the **DSCInitialization** CR.
5. If Service Mesh was set to **Managed** in the **DSCInitialization** CR, wait until all pods are deleted from the **istio-system** namespace.
6. Uninstall the Red Hat OpenShift AI Operator.
7. Wait until all pods are deleted from the **redhat-ods-operator** namespace and the **redhat-ods-monitoring** namespace.

RHOAIENG-880 - Default pipelines service account is unable to create Ray clusters

You cannot create Ray clusters using the default pipelines Service Account.

Workaround

Authenticate using the CodeFlare SDK, by adding the following lines to the pipeline code:

```
from codeflare_sdk.cluster.auth import TokenAuthentication
  auth = TokenAuthentication(
    token=openshift_token, server=openshift_server, skip_tls=True
  )
  auth_return = auth.login()
```

If a deployed model does not receive at least one HTTP request for each of the two data types (success and failed), the graphs that show HTTP request performance metrics (for all models on the model server or for the specific model) render incorrectly, with a straight line that indicates a steadily increasing number of failed requests.

Workaround

After the deployed data model receives at least one HTTP request that is successful and one that is failed, the graphs show the HTTP request performance metrics correctly. The graphs work correctly as long as one HTTP request of each data type (success and failed) occur at any point in the history of the deployed model, regardless of the time range that you specify for the graphs.

A No Components Found page might appear when you access the Red Hat OpenShift AI dashboard.

Workaround

Refresh the browser page.

RHOAIENG-234 - Unable to view .ipynb files in VSCode in Insecured cluster

When you use the code-server notebook image on Google Chrome in an insecure cluster, you cannot view .ipynb files.

Workaround

Use a different browser.

When you set a number of model server replicas different from the default (1), the model (server) is still deployed with 1 replica. —-

RHOAIENG-2184 - Cannot create Ray clusters or distributed workloads

Users cannot create Ray clusters or distributed workloads in namespaces where they have **admin** or **edit** permissions.

Workaround

To grant the appropriate permissions, create a ClusterRole for the resources created by the KubeRay Operator and CodeFlare Operator, and specify the **admin** and **edit** aggregation labels, as described in the Red Hat Knowledgebase solution How to grant permission to create Ray clusters and distributed workloads in RHOAI.

RHOAIENG-2099 - Data science pipeline server fails to deploy in fresh cluster

When you create a data science pipeline server on a fresh cluster, the user interface remains in a loading state and the pipeline server does not start. A “Pipeline server failed” error message might be displayed.

Workaround

Delete the pipeline server and create a new one.

If the problem persists, disable the database health check in the DSPA custom resource:

1. Use the following command to edit the custom resource:

   ```
   $ oc edit dspa pipelines-definition -n my-project
   ```

2. Set the **spec.database.disableHealthCheck** value to **true**.

3. Save the change.

RHOAIENG-908 - Cannot use ModelMesh if KServe was previously enabled and then removed

When both ModelMesh and KServe are enabled in the **DataScienceCluster** object, and you subsequently remove KServe, you can no longer deploy new models with ModelMesh. You can continue to use models that were previously deployed with ModelMesh.

Example error message:

```
Error creating model serverInternal error occurred: failed calling webhook "inferenceservice.kserve-
webhook-server.defaulter": failed to call webhook: Post "https://kserve-webhook-server-
service.redhat-ods-applications.svc:443/mutate-serving-kserve-io-v1beta1-inferenceservice?
timeout=10s": service "kserve-webhook-server-service" not found
```

Workaround

You can resolve this issue in either of the following ways:

- Re-enable KServe.

- Delete the KServe MutatingWebHook configuration by completing the following steps as a user with **cluster-admin** permissions:

  1. Log in to your cluster by using the **oc** client.

  2. Enter the following command:

```
oc delete mutatingwebhookconfigurations inferenceservice.serving.kserve.io
```

RHOAIENG-807 - Accelerator profile toleration removed when restarting a workbench

If you create a workbench that uses an accelerator profile that in turn includes a toleration, restarting the workbench removes the toleration information, which means that the restart cannot complete. A freshly created GPU-enabled workbench might start the first time, but never successfully restarts afterwards because the generated pod remains forever pending.

RHOAIENG-804 - Cannot deploy Large Language Models with KServe on FIPS-enabled clusters

Red Hat OpenShift AI is not yet fully designed for FIPS. You cannot deploy Large Language Models (LLMs) with KServe on FIPS-enabled clusters.

RHOAIENG-517 - User with edit permissions cannot see created models

A user with edit permissions cannot see any created models, unless they are the project owner or have admin permissions for the project.

Workaround

If the project owner or a user with admin permissions subsequently creates a model, the user with edit permissions can then see all models.

RHOAIENG-499 - Uninstalling Red Hat OpenShift AI Self Managed by using the CLI does not uninstall

If you uninstall Red Hat OpenShift AI by using the command-line interface, then the **DataScienceCluster** CR, the **DSCInitialization** CR, and the Red Hat OpenShift AI Operator are not removed.

Workaround

Manually delete the remaining OpenShift AI resources as follows:

1. Delete the **DataScienceCluster** CR.
2. Wait until all pods are deleted from the **redhat-ods-applications** namespace.
3. If Serverless was set to **Managed** in the **DataScienceCluster** CR, wait until all pods are deleted from the **knative-serving** namespace.
4. Delete the **DSCInitialization** CR.
5. If Service Mesh was set to **Managed** in the **DSCInitialization** CR, wait until all pods are deleted from the **istio-system** namespace.
6. Uninstall the Red Hat OpenShift AI Operator.
7. Wait until all pods are deleted from the **redhat-ods-operator** namespace and the **redhat-ods-monitoring** namespace.

RHOAIENG-343 - Manual configuration of OpenShift Service Mesh and OpenShift Serverless does not work for KServe

If you install OpenShift Serverless and OpenShift Service Mesh and then install Red Hat OpenShift AI with KServe enabled, KServe is not deployed.

Workaround

1. Edit the **DSCInitialization** resource: Set the **managementState** field of the **serviceMesh** component to **Unmanaged**.
2. Edit the **DataScienceCluster** resource: Within the **kserve** component, set the **managementState** field of the **serving** component to **Unmanaged**. For more information, see Installing KServe.

RHOAIENG-339 - KServe component images are not updated after upgrade to 2.5

Previously, the KServe component was a Limited Availability feature. If you enabled the **kserve** component and created models in an earlier version, then after you upgrade to Red Hat OpenShift AI 2.5, you must update some OpenShift AI resources as follows:

1. Log in as an admin user to the OpenShift Container Platform cluster where OpenShift AI 2.5 is installed:

```
$ oc login
```

2. Update the **DSCInitialization** resource as follows:

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

3. Update the **DataScienceCluster** resource as follows:

```
$ 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 as follows:

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

5. Optionally, restart the Operator pod.

RHOAIENG-293 - Deprecated ModelMesh monitoring stack not deleted after upgrading from 2.4 to 2.5

In Red Hat OpenShift AI 2.5, the former ModelMesh monitoring stack is no longer deployed because it is replaced by user workload monitoring. However, the former monitoring stack is not deleted during an upgrade to OpenShift AI 2.5. Some components remain and use cluster resources.

RHOAIENG-288 - Recommended image version label for workbench is shown for two versions

Most of the workbench images that are available in OpenShift AI are provided in multiple versions. The only recommended version is the latest version. In the current release, the **Recommended** tag is erroneously shown for multiple versions of an image.

RHOAIENG-162 - Project remains selected after navigating to another page

When you select a project on the **Data Science Projects** page, the project remains selected, even after you navigate to another page. For example, if you subsequently open the **Model Serving** page, the page lists only the models for the previously selected project, instead of the models for all projects.

Workaround

From the **Project** list, select **All projects**.

RHOAIENG-84 - Cannot use self-signed certificates with KServe

The single model serving platform does not support self-signed certificates.

Workaround

To deploy a model from S3 storage, disable SSL authentication as described in the Red Hat Knowledgebase solution How to skip the validation of SSL for KServe .

RHOAIENG-66 - Ray dashboard route deployed by CodeFlare SDK exposes self-signed certs instead of cluster cert

When you deploy a Ray cluster by using the CodeFlare SDK with the **openshift_oauth=True** option, the resulting route for the Ray cluster is secured by using the **passthrough** method. As a result, the self-signed certificate used by the OAuth proxy is exposed.

Workaround

Use one of the following workarounds:

- Set the **openshift_oauth** option to **False**.
- Add the self-signed certificate used by the OAuth proxy to the client's truststore.
- Create a route manually, using a route configuration and certificate that is based on the needs of the client.

RHOAIENG-1199 (previously documented asODH-DASHBOARD-1928 - Custom serving runtime creation error message is unhelpful

When you try to create or edit a custom model-serving runtime and an error occurs, the error message does not indicate the cause of the error.

Example error message: **Request failed with status code 422**

Workaround

Check the YAML code for the serving runtime to identify the reason for the error.

ODH-DASHBOARD-1991 - ovms-gpu-ootb is missing recommended accelerator annotation

When you add a model server to your project, the **Serving runtime** list does not show the **Recommended serving runtime** label for the NVIDIA GPU.

Workaround

Make a copy of the model-server template and manually add the label.

RHODS-12717 - Pipeline server creation might fail on OpenShift Container Platform with Open Virtual Network on OpenStack

When you try to create a pipeline server on OpenShift Container Platform with Open Virtual Network on OpenStack, the creation might fail with a **Pipeline server failed** error. See OCPBUGS-22251.

RHODS-12899 - OpenVINO runtime missing annotation for NVIDIA GPUs

Red Hat OpenShift AI currently includes an out-of-the-box serving runtime that supports NVIDIA GPUs: **OpenVINO model server (support GPUs)** You can use the accelerator profile feature introduced in OpenShift AI 2.4 to select a specific accelerator in model serving, based on configured accelerator profiles. If the cluster had NVIDIA GPUs enabled in an earlier OpenShift AI release, the system automatically creates a default NVIDIA accelerator profile during upgrade to OpenShift AI 2.4. However, the **OpenVINO model server (supports GPUs)**runtime has not been annotated to indicate that it supports NVIDIA GPUs. Therefore, if a user selects the **OpenVINO model server (supports GPUs)** runtime and selects an NVIDIA GPU accelerator in the model server user interface, the system displays a warning that the selected accelerator is not compatible with the selected runtime. In this situation, you can ignore the warning.

RHOAIENG-637 (previously documented as RHODS-12904) - Pipeline submitted from Elyra might fail when using private certificate

If you use a private TLS certificate, and you submit a pipeline from Elyra, the pipeline might fail with a **certificate verify failed** error message. This issue might be caused by either or both of the following situations:

- The object storage used for the pipeline server is using private TLS certificates.
- The data science pipeline server API endpoint is using private TLS certificates.

Workaround

Provide the workbench with the correct Certificate Authority (CA) bundle, and set various environment variables so that the correct CA bundle is recognized. Contact Red Hat Support for detailed steps to resolve this issue.

RHODS-12906 - Cannot use ModelMesh with object storage that uses private certificates

Sometimes, when you store models in an object storage provider that uses a private TLS certificate, the model serving pods fail to pull files from the object storage, and the **signed by unknown authority** error message is shown.

Workaround

Manually update the secret created by the data connection so that the secret includes the correct CA bundle. Contact Red Hat Support for detailed steps to resolve this issue.

RHODS-12937 - Previously deployed model server might no longer work after upgrade in disconnected environment

In disconnected environments, after upgrade to Red Hat OpenShift AI 2.8, previously deployed model servers might no longer work. The model status might be incorrectly reported as **OK** on the dashboard.

Workaround

Update the **inferenceservices** resource to replace the **storage** section with the **storageUri** section. In the following instructions, replace *<placeholders>* with the values for your environment.

1. Remove the **storage** parameter section from the existing **inferenceservices** resource:

```
"storage":
    "key": "<your_key>",
    "path": "<your_path>"
```

Example:

```
"storage":
    "key": "aws-connection-minio-connection",
    "path": "mnist-8.onnx"
```

2. Add the **storageUri** section to the **inferenceservices** resource, with the specified format **s3://bucket-name/path/to/object**, as shown in the following example:
   Example:

   ```
   storageUri: 's3://bucket/mnist-8.onnx'
   ```

3. Capture the secret key name as follows:

   ```
   secret_key=$(oc get secret -n <project_name> | grep -i aws-connection | awk '{print $1}')
   ```

4. Update the annotation as follows:

   ```
   oc annotate $(oc get inferenceservices -n <project_name> -o name) -n <project_name>
   serving.kserve.io/secretKey="$secret_key"
   ```

RHOAIENG-12 - Cannot access Ray dashboard from some browsers

In some browsers, users of the distributed workloads feature might not be able to access the Ray dashboard, because the browser automatically changes the prefix of the dashboard URL from **http** to **https**. The distributed workloads feature is currently available in Red Hat OpenShift AI as a Technology Preview feature. See Technology Preview features.

Workaround

Change the URL prefix from **https** to **http**.

DATA-SCIENCE-PIPELINES-OPERATOR-362 - Pipeline server fails that uses object storage signed by an unknown authority

Data science pipeline servers fail if you use object storage signed by an unknown authority. As a result, you cannot currently use object storage with a self-signed certificate. This issue has been observed in a disconnected environment.

Workaround

Configure your system to use object storage with a self-signed certificate, as described in the Red Hat Knowledgebase solution Data Science Pipelines workaround for an object storage connection with a self-signed certificate.

RHOAIENG-548 (previously documented asODH-DASHBOARD-1776) - Error messages when user does not have project administrator permission

If you do not have administrator permission for a project, you cannot access some features, and the error messages do not explain why. For example, when you create a model server in an environment where you only have access to a single namespace, an **Error creating model server** error message appears. However, the model server is still successfully created.

DATA-SCIENCE-PIPELINES-OPERATOR-294 - Scheduled pipeline run that uses data-passing might fail to pass data between steps, or fail the step entirely

A scheduled pipeline run that uses an S3 object store to store the pipeline artifacts might fail with an error such as the following:

> Bad value for --endpoint-url "cp": scheme is missing. Must be of the form http://<hostname>/ or https://<hostname>/

This issue occurs because the S3 object store endpoint is not successfully passed to the pods for the scheduled pipeline run.

Workaround

Depending on the size of the pipeline artifacts being passed, you can either partially or completely work around this issue by applying a custom artifact-passing script and then restarting the pipeline server. Specifically, this workaround results in the following behavior:

- For pipeline artifacts smaller than 3 kilobytes, the pipeline run now successfully passes the artifacts into your S3 object store.
- For pipeline artifacts larger than 3 kilobytes, the pipeline run still *does not* pass the artifacts into your S3 object store. However, the workaround ensures that the run continues to completion. Any smaller artifacts in the remainder of the pipeline run are successfully stored.

To apply this workaround, perform the following actions:

1. In a text editor, paste the following YAML-based artifact-passing script. The script defines a **ConfigMap** object.

```
apiVersion: v1
data:
  artifact_script: |-
  #!/usr/bin/env sh
    push_artifact() {
       workspace_dir=$(echo $(context.taskRun.name) | sed -e "s/$(context.pipeline.name)-
//g")

workspace_dest=/workspace/${workspace_dir}/artifacts/$(context.pipelineRun.name)/$(context.
taskRun.name)
        artifact_name=$(basename $2)
        if [ -f "$workspace_dest/$artifact_name" ]; then
            echo sending to: ${workspace_dest}/${artifact_name}
            tar -cvzf $1.tgz -C ${workspace_dest} ${artifact_name}
            aws s3 --endpoint <Endpoint> cp $1.tgz
s3://<Bucket>/artifacts/$PIPELINERUN/$PIPELINETASK/$1.tgz
        elif [ -f "$2" ]; then
            tar -cvzf $1.tgz -C $(dirname $2) ${artifact_name}
            aws s3 --endpoint <Endpoint> cp $1.tgz
s3://<Bucket>/artifacts/$PIPELINERUN/$PIPELINETASK/$1.tgz
        else
            echo "$2 file does not exist. Skip artifact tracking for $1"
        fi
    }
    push_log() {
        cat /var/log/containers/$PODNAME*$NAMESPACE*step-main*.log > step-main.log
        push_artifact main-log step-main.log
    }
    strip_eof() {
        if [ -f "$2" ]; then
            awk 'NF' $2 | head -c -1 > $1_temp_save && cp $1_temp_save $2
        fi
    }
kind: ConfigMap
metadata:
  name: custom-script
 ----

. In the script, replace any occurrences  of _<Endpoint>_ with your S3 endpoint (for example,
https://s3.amazonaws.com), and occurrences of _<Bucket>_ with your S3 bucket name.
. Save the YAML file for the `ConfigMap` object.

. Apply the YAML file.
+
[source,subs="+quotes"]
```

```
$ oc apply -f <configmap_file_name>.yaml
```

```
. Restart the pipeline server.
+
[source,subs="+quotes"]
```

```
$ oc project <data_science_project_name> $ oc delete pod $(oc get pods -l app=ds-pipeline-pipelines-definition --no-headers | awk {print $1})
```

> RHODS-9764 - Data connection details get reset when editing a workbench
>
> When you edit a workbench that has an existing data connection and then select the Create new data connection option, the edit page might revert to the Use existing data connection option before you have finished specifying the new connection details.
>
> Workaround
>
> To work around this issue, perform the following actions:
>
> 1. Select the Create new data connection option again.
> 2. Specify the new connection details and click Update workbench before the page reverts to the Use existing data connection option.

RHODS-9030 - Uninstall process for OpenShift AI might become stuck when removing kfdefs resources

The steps for uninstalling OpenShift AI self-managed are described in Uninstalling OpenShift AI self-managed.

However, even when you follow this guide, you might see that the uninstall process does not finish successfully. Instead, the process stays on the step of deleting kfdefs resources that are used by the Kubeflow Operator. As shown in the following example, kfdefs resources might exist in the redhat-ods-applications, redhat-ods-monitoring, and rhods-notebooks namespaces:

```
$ oc get kfdefs.kfdef.apps.kubeflow.org -A

NAMESPACE                  NAME                                   AGE
redhat-ods-applications    rhods-anaconda                         3h6m
redhat-ods-applications    rhods-dashboard                        3h6m
redhat-ods-applications    rhods-data-science-pipelines-operator  3h6m
redhat-ods-applications    rhods-model-mesh                       3h6m
redhat-ods-applications    rhods-nbc                              3h6m
redhat-ods-applications    rhods-osd-config                       3h6m
redhat-ods-monitoring      modelmesh-monitoring                   3h6m
redhat-ods-monitoring      monitoring                             3h6m
rhods-notebooks            rhods-notebooks                        3h6m
rhods-notebooks            rhods-osd-config                       3h5m
```

Failed removal of the kfdefs resources might also prevent later installation of a newer version of OpenShift AI.

Workaround

To manually delete the kfdefs resources so that you can complete the uninstall process, see the "Force individual object removal when it has finalizers" section of the Red Hat Knowledgebase solution Unable to Delete a Project or Namespace in OCP.

RHODS-8939 - For a Jupyter notebook created in a previous release, default shared memory might cause a runtime error

For a Jupyter notebook created in a release earlier than 1.31, the default shared memory for a Jupyter notebook is set to 64 MB and you cannot change this default value in the notebook configuration.

For example, PyTorch relies on shared memory and the default size of 64 MB is not enough for large use cases, such as training a model or performing heavy data manipulations. Jupyter reports a "no space left on device" message and /dev/smh is full.

Starting with release 1.31, this issue is fixed and any new notebook's shared memory is set to the size of the node.

Workaround

For a Jupyter notebook created in a release earlier than 1.31, either recreate the Jupyter notebook or follow these steps:

1. In your data science project, create a workbench as described in Creating a project workbench.

2. In the data science project page, in the Workbenches section, click the Status toggle for the workbench to change it from Running to Stopped.

3. Open your OpenShift Console and then select Administrator.

4. Select Home → API Explorer.

5. In the Filter by kind field, type notebook.

6. Select the kubeflow v1 notebook.

7. Select the Instances tab and then select the instance for the workbench that you created in Step 1.

8. Click the YAML tab and then select Actions → Edit Notebook.

9. Edit the YAML file to add the following information to the configuration:

   -

For the container that has the name of your Workbench notebook, add the following lines to the volumeMounts section:

```
- mountPath: /dev/shm
  name: shm
```

For example, if your workbench name is myworkbench, update the YAML file as follows:

```
spec:
    containers:
      - env
        ...
        name: myworkbench
        ...
         volumeMounts:
         - mountPath: /dev/shm
           name: shm
```

- In the volumes section, add the lines shown in the following example:

```
     volumes:
       name: shm
       emptyDir:
         medium: Memory
```

Note: Optionally, you can specify a limit to the amount of memory to use for the emptyDir.

10. Click Save.

11. In the data science dashboard, in the Workbenches section of the data science project, click the Status toggle for the workbench. The status changes from Stopped to Starting and then Running.

12. Restart the notebook.

WARNING

If you later edit the notebook's configuration through the Data Science dashboard UI, your workaround edit to the notebook configuration will be erased.

RHODS-8865 - A pipeline server fails to start unless you specify an Amazon Web Services (AWS) Simple Storage Service (S3) bucket resource

When you create a data connection for a data science project, the AWS_S3_BUCKET field is not designated as a mandatory field. However, if you do not specify a value for this field, and you attempt to configure a pipeline server, the pipeline server fails to start successfully.

RHODS-6907 - Attempting to increase the size of a Persistent Volume (PV) fails when it is not connected to a workbench

Attempting to increase the size of a Persistent Volume (PV) that is not connected to a workbench fails. When changing a data science project's storage, users can still edit the size of the PV in the user interface, but this action does not have any effect.

RHODS-6950 - Unable to scale down a workbench's GPUs when all GPUs in the cluster are being used

It is not possible to scale down a workbench's GPUs if all GPUs in the cluster are being used. This issue applies to GPUs being used by one workbench, and GPUs being used by multiple workbenches.

Workaround

To workaround around this issue, perform the following steps:

1. Stop all active workbenches that are using GPUs.
2. Wait until the relevant GPUs are available again.

3. Edit the workbench and scale down the GPU instances.

RHODS-6539 - Anaconda Professional Edition cannot be validated and enabled in OpenShift AI

Anaconda Professional Edition cannot be enabled as the dashboard's key validation for Anaconda Professional Edition is inoperable.

RHODS-6346 - Unclear error message displays when using invalid characters to create a data science project

When creating a data science project's data connection, workbench, or storage connection using invalid special characters, the following error message is displayed:

> the object provided is unrecognized (must be of type Secret): couldn't get version/kind; json parse error: unexpected end of JSON input ({"apiVersion":"v1","kind":"Sec ...)

The error message fails to clearly indicate the problem.

RHODS-6913 - When editing the configuration settings of a workbench, a misleading error message appears

When you edit the configuration settings of a workbench, a warning message appears stating the workbench will restart if you make any changes to its configuration settings. This warning is misleading, as if you change the values of its environment variables, the workbench does not automatically restart.

RHODS-6373 - Workbenches fail to start when cumulative character limit is exceeded

When the cumulative character limit of a data science project's title and workbench title exceeds 62 characters, workbenches fail to start.

RHODS-6216 - The ModelMesh oauth-proxy container is intermittently unstable

ModelMesh pods do not deploy correctly due to a failure of the ModelMesh oauth-proxy container. This issue occurs intermittently and only if authentication is enabled in the ModelMesh runtime environment. It is more likely to occur when additional ModelMesh instances are deployed in different namespaces.

RHODS-4769 - GPUs on nodes with unsupported taints cannot be allocated to notebook servers

GPUs on nodes marked with any taint other than the supported *nvidia.com/gpu* taint cannot be selected when creating a notebook server. To avoid this issue, use only the *nvidia.com/gpu* taint on GPU nodes used with OpenShift AI.