Red Hat OpenShift AI Cloud Service 1

Release notes

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

Last Updated: 2024-04-10

Red Hat OpenShift AI Cloud Service 1 Release notes

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

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

These release notes provide an overview of new features, enhancements, resolved issues, and known issues in this release of Red Hat OpenShift AI. OpenShift AI is currently available in Red Hat OpenShift Dedicated and Red Hat OpenShift Service on Amazon Web Services (ROSA).

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.

2.1. 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 Dedicated cluster.

Some OpenShift AI components have additional options or required configuration for self-signed certificates, as described in 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.

CHAPTER 3. TECHNOLOGY PREVIEW FEATURES

IMPORTANT

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

RStudio Server notebook image

With the **RStudio Server** notebook image, you can access the RStudio IDE, an integrated development environment for R. The R programming language is used for statistical computing and graphics to support data analysis and predictions.
To use the **RStudio Server** notebook image, you must first build it by creating a secret and triggering the **BuildConfig**, and then enable it in the OpenShift AI UI by editing the **rstudio-rhel9** image stream. See Building the RStudio Server notebook images for more information.

IMPORTANT

Disclaimer: Red Hat supports managing workbenches in OpenShift AI. However, Red Hat does not provide support for the RStudio software. RStudio Server is available through rstudio.org and is subject to their licensing terms. You should review their licensing terms before you use this sample workbench.

CUDA - RStudio Server notebook image

With the **CUDA - RStudio Server** notebook image, you can access the RStudio IDE and NVIDIA CUDA Toolkit. The RStudio IDE is an integrated development environment for the R programming language for statistical computing and graphics. With the NVIDIA CUDA toolkit, you can enhance your work by using GPU-accelerated libraries and optimization tools.

To use the **CUDA - RStudio Server** notebook image, you must first build it by creating a secret and triggering the **BuildConfig**, and then enable it in the OpenShift AI UI by editing the **rstudio-rhel9** image stream. See Building the RStudio Server notebook images for more information.

IMPORTANT

Disclaimer: Red Hat supports managing workbenches in OpenShift AI. However, Red Hat does not provide support for the RStudio software. RStudio Server is available through rstudio.org and is subject to their licensing terms. You should review their licensing terms before you use this sample workbench.

The **CUDA - RStudio Server** notebook image contains NVIDIA CUDA technology. CUDA licensing information is available in the CUDA Toolkit documentation. You should review their licensing terms before you use this sample workbench.

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 as a Technology Preview feature.

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 as a Technology Preview feature.

CHAPTER 4. SUPPORT REMOVALS

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

4.1. 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.

4.2. 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. For more information, see Red Hat OpenShift AI Life Cycle.

4.3. 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.

4.4. NVIDIA GPU OPERATOR REPLACES NVIDIA GPU ADD-ON

Previously, to enable graphics processing units (GPUs) to help with compute-heavy workloads, you installed the NVIDIA GPU add-on. OpenShift AI no longer supports this add-on.

Now, to enable GPU support, you must install the NVIDIA GPU Operator. To learn how to install the GPU Operator, see NVIDIA GPU Operator on Red Hat OpenShift Container Platform (external).

4.5. KUBEFLOW NOTEBOOK CONTROLLER REPLACES JUPYTERHUB

In OpenShift AI 1.15 and earlier, JupyterHub was used to create and launch notebook server environments. In OpenShift AI 1.16 and later, JupyterHub is no longer included, and its functionality is replaced by Kubeflow Notebook Controller.

This change provides the following benefits:

- Users can now immediately cancel a request, make changes, and retry the request, instead of waiting 5+ minutes for the initial request to time out. This means that users do not wait as long when requests fail, for example, when a notebook server does not start correctly.
- The architecture no longer prevents a single user from having more than one notebook server session, expanding future feature possibilities.
- The removal of the PostgreSQL database requirement allows for future expanded environment support in OpenShift AI.

However, this update also creates the following behavior changes:

- For IT Operations administrators, the notebook server administration interface does not currently allow login access to data scientist users' notebook servers. This is planned to be added in future releases.
- For data scientists, the JupyterHub interface URL is no longer valid. Update your bookmarks to point to the OpenShift AI Dashboard.

The JupyterLab interface is unchanged and data scientists can continue to use JupyterLab to work with their notebook files as usual.

CHAPTER 5. RESOLVED ISSUES

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

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 as ODH-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-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 as ODH-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.

RHOAIENG-535 - Metrics graph showing HTTP requests for deployed models is incorrect if there are no HTTP requests

Previously, if a deployed model did 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) rendered incorrectly, with a straight line that indicated a steadily increasing number of failed requests. This issue is now resolved.

RHOAIENG-1467 - Serverless net-istio controller pod might hit OOM

Previously, the Knative **net-istio-controller** pod (which is a dependency for KServe) might continuously crash due to an out-of-memory (OOM) error. This issue is now resolved.

RHOAIENG-1899 (previously documented asRHODS-6539) - The Anaconda Professional Edition cannot be validated and enabled

Previously, you could not enable the Anaconda Professional Edition because the dashboard's key validation for it was inoperable. This issue is now resolved.

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

Previously, on a single-model serving platform, the **Models and model servers**section of a data science project did not show the correct number of model replicas. This issue is now resolved.

RHOAIENG-2270 - (Single-model) Users cannot update model deployment settings

Previously, you couldn't edit the deployment settings (for example, the number of replicas) of a model you deployed with a single-model serving platform. This issue is now resolved.

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

Previously, when you created a data connection for a data science project, the **AWS_S3_BUCKET** field was not designated as a mandatory field. However, if you attempted to configure a pipeline server with a data connection where the **AWS_S3_BUCKET** field was not populated, the pipeline server failed to start successfully. This issue is now resolved. The **Configure pipeline server** dialog has been updated to include the **Bucket** field as a mandatory field.

RHODS-12899 - OpenVINO runtime missing annotation for NVIDIA GPUs

Previously, if a user selected the **OpenVINO model server (supports GPUs)**runtime and selected an NVIDIA GPU accelerator in the model server user interface, the system could display a unnecessary warning that the selected accelerator was not compatible with the selected runtime. The warning is no longer displayed.

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

Previously, the single model serving platform did not support self-signed certificates. This issue is now resolved. To use self-signed certificates with KServe, follow the steps described in Working with certificates.

RHOAIENG-164 - Number of model server replicas for Kserve is not applied correctly from the dashboard

Previously, when you set a number of model server replicas different from the default (1), the model (server) was still deployed with 1 replica. This issue is now resolved.

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 Red Hat OpenShift AI 2.4 and 2.5, the **Recommended** tag was erroneously shown for multiple versions of an image. This issue is now resolved.

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 was no longer deployed because it was replaced by user workload monitoring. However, the former monitoring stack was not deleted during an upgrade to OpenShift AI 2.5. Some components remained and used cluster resources. This issue is now resolved.

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

If you installed OpenShift Serverless and OpenShift Service Mesh and then installed Red Hat OpenShift AI with KServe enabled, KServe was not deployed. This issue is now resolved.

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

A user with edit permissions could not see any created models, unless they were the project owner or had admin permissions for the project. This issue is now resolved.

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

Previously, Red Hat OpenShift AI was not yet fully designed for FIPS. You could not deploy Large Language Models (LLMs) with KServe on FIPS-enabled clusters. This issue is now resolved.

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

Previously, when both ModelMesh and KServe were enabled in the **DataScienceCluster** object, and you subsequently removed KServe, you could no longer deploy new models with ModelMesh. You could continue to use models that were previously deployed with ModelMesh. This issue is now resolved.

RHOAIENG-2184 - Cannot create Ray clusters or distributed workloads

Previously, users could not create Ray clusters or distributed workloads in namespaces where they have **admin** or **edit** permissions. This issue is now resolved.

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

Previously, when you added a model server to your project, the **Serving runtime** list did not show the **Recommended serving runtime** label for the NVIDIA GPU. This issue is now resolved.

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

Previously, if you created a workbench that used an accelerator profile that in turn included a toleration, restarting the workbench removed the toleration information, which meant that the restart could not complete. A freshly created GPU-enabled workbench might start the first time, but never successfully restarted afterwards because the generated pod remained forever pending. This issue is now resolved.

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 occurred because the S3 object store endpoint was not successfully passed to the pods for the scheduled pipeline run. This issue is now resolved.

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 could not be selected when creating a notebook server. This issue is now resolved.

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 was 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 failed to clearly indicate the problem. The error message now indicates that invalid characters were entered.

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

In earlier releases, it was not possible to scale down workbench GPUs if all GPUs in the cluster were being used. This issue applied to GPUs being used by one workbench, and GPUs being used by multiple workbenches. You can now scale down the GPUs by selecting **None** from the **Accelerators** list.

RHODS-8939 - Default shared memory for a Jupyter notebook created in a previous release causes a runtime error

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

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.

To fix this issue, you must recreate the notebook or follow the process described in the Knowledgebase article How to change the shared memory for a Jupyter notebook in Red Hat OpenShift AI .

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

The steps for uninstalling the OpenShift AI managed service are described in Uninstalling OpenShift AI.

However, even when you followed this guide, you might have seen that the uninstall process did not finish successfully. Instead, the process stayed on the step of deleting **kfdefs** resources that were 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 have also prevented later installation of a newer version of OpenShift AI. This issue no longer occurs.

RHODS-9764 - Data connection details get reset when editing a workbench

When you edited a workbench that had an existing data connection and then selected the **Create new data connection** option, the edit page might revert to the **Use existing data connection** option before you had finished specifying the new connection details. This issue is now resolved.

RHODS-9583 - Data Science dashboard did not detect an existing OpenShift Pipelines installation

When the OpenShift Pipelines Operator was installed as a global operator on your cluster, the OpenShift AI dashboard did not detect it. The OpenShift Pipelines Operator is now detected successfully.

ODH-DASHBOARD-1639 - Wrong TLS value in dashboard route

Previously, when a route was created for the OpenShift AI dashboard on OpenShift, the **tls.termination** field had an invalid default value of **Reencrypt**. This issue is now resolved. The new value is **reencrypt**.

ODH-DASHBOARD-1638 - Name placeholder in Triggered Runs tab shows Scheduled run name

Previously, when you clicked **Pipelines** > **Runs** and then selected the **Triggered** tab to configure a triggered run, the example value shown in the **Name** field was **Scheduled run name**. This issue is now resolved.

ODH-DASHBOARD-1547 - "We can't find that page" message displayed in dashboard when pipeline operator installed in background

Previously, when you used the **Data Science Pipelines** page of the dashboard to install the OpenShift Pipelines Operator, when the Operator installation was complete, the page refreshed to show a "We can't find that page" message. This issue is now resolved. When the Operator installation is complete, the dashboard redirects you to the **Pipelines** page, where you can create a pipeline server.

ODH-DASHBOARD-1545 - Dashboard keeps scrolling to bottom of project when Models tab is expanded

Previously, on the **Data Science Projects** page of the dashboard, if you clicked the **Deployed models** tab to expand it and then tried to perform other actions on the page, the page automatically scrolled back to the **Deployed models** section. This affected your ability to perform other actions. This issue is now resolved.

NOTEBOOKS-156 - Elyra included an example runtime called Test

Previously, Elyra included an example runtime configuration called **Test**. If you selected this configuration when running a data science pipeline, you could see errors. The **Test** configuration has now been removed.

RHODS-9622 - Duplicating a scheduled pipeline run does not copy the existing period and pipeline input parameter values

Previously, when you duplicated a scheduled pipeline run that had a periodic trigger, the duplication process did not copy the configured execution frequency for the recurring run or the specified pipeline input parameters. This issue is now resolved.

RHODS-8932 - Incorrect cron format was displayed by default when scheduling a recurring pipeline run

When you scheduled a recurring pipeline run by configuring a cron job, the OpenShift AI interface displayed an incorrect format by default. It now displays the correct format.

RHODS-9374 - Pipelines with non-unique names did not appear in the data science project user interface

If you launched a notebook from a Jupyter application that supported Elyra, or if you used a workbench, when you submitted a pipeline to be run, pipelines with non-unique names did not appear in the **Pipelines** section of the relevant data science project page or the **Pipelines** heading of the data science pipelines page. This issue has now been resolved.

RHODS-9329 - Deploying a custom model-serving runtime could result in an error message

Previously, if you used the OpenShift AI dashboard to deploy a custom model-serving runtime, the deployment process could fail with an **Error retrieving Serving Runtime** message. This issue is now resolved.

RHODS-9064 - After upgrade, the Data Science Pipelines tab was not enabled on the OpenShift AI dashboard

When you upgraded from OpenShift AI 1.26 to OpenShift AI 1.28, the **Data Science Pipelines** tab was not enabled in the OpenShift AI dashboard. This issue is resolved in OpenShift AI 1.29.

RHODS-9443 - Exporting an Elyra pipeline exposed S3 storage credentials in plain text

In OpenShift AI 1.28.0, when you exported an Elyra pipeline from JupyterLab in Python DSL format or YAML format, the generated output contained S3 storage credentials in plain text. This issue has been resolved in OpenShift AI 1.28.1. However, after you upgrade to OpenShift AI 1.28.1, if your deployment contains a data science project with a pipeline server and a data connection, you must perform the following additional actions for the fix to take effect:

1. Refresh your browser page.
2. Stop any running workbenches in your deployment and restart them.

Furthermore, to confirm that your Elyra runtime configuration contains the fix, perform the following actions:

1. In the left sidebar of JupyterLab, click **Runtimes** ().
2. Hover the cursor over the runtime configuration that you want to view and click the **Edit** button ().
   The **Data Science Pipelines runtime configuration** page opens.
3. Confirm that **KUBERNETES_SECRET** is defined as the value in the **Cloud Object Storage Authentication Type** field.
4. Close the runtime configuration without changing it.

RHODS-8460 - When editing the details of a shared project, the user interface remained in a loading state without reporting an error

When a user with permission to edit a project attempted to edit its details, the user interface remained in a loading state and did not display an appropriate error message. Users with permission to edit projects cannot edit any fields in the project, such as its description. Those users can edit only components belonging to a project, such as its workbenches, data connections, and storage.

The user interface now displays an appropriate error message and does not try to update the project description.

RHODS-8482 - Data science pipeline graphs did not display node edges for running pipelines

If you ran pipelines that did not contain Tekton-formatted **Parameters** or **when** expressions in their YAML code, the OpenShift AI user interface did not display connecting edges to and from graph nodes. For example, if you used a pipeline containing the **runAfter** property or **Workspaces**, the user interface displayed the graph for the executed pipeline without edge connections. The OpenShift AI user interface now displays connecting edges to and from graph nodes.

RHODS-8923 - Newly created data connections were not detected when you attempted to create a pipeline server

If you created a data connection from within a Data Science project, and then attempted to create a pipeline server, the **Configure a pipeline server** dialog did not detect the data connection that you created. This issue is now resolved.

RHODS-8461 - When sharing a project with another user, the OpenShift AI user interface text was misleading

When you attempted to share a Data Science project with another user, the user interface text misleadingly implied that users could edit all of its details, such as its description. However, users can edit only components belonging to a project, such as its workbenches, data connections, and storage. This issue is now resolved and the user interface text no longer misleadingly implies that users can edit all of its details.

RHODS-8462 - Users with "Edit" permission could not create a Model Server

Users with "Edit" permissions can now create a Model Server without token authorization. Users must have "Admin" permissions to create a Model Server with token authorization.

RHODS-8796 - OpenVINO Model Server runtime did not have the required flag to force GPU usage

OpenShift AI includes the OpenVINO Model Server (OVMS) model-serving runtime by default. When you configured a new model server and chose this runtime, the **Configure model server** dialog enabled you to specify a number of GPUs to use with the model server. However, when you finished configuring the model server and deployed models from it, the model server did not actually use any GPUs. This issue is now resolved and the model server uses the GPUs.

RHODS-8861 - Changing the host project when creating a pipeline ran resulted in an inaccurate list of available pipelines

If you changed the host project while creating a pipeline run, the interface failed to make the pipelines of the new host project available. Instead, the interface showed pipelines that belong to the project you initially selected on the **Data Science Pipelines** > **Runs** page. This issue is now resolved. You no longer select a pipeline from the **Create run** page. The pipeline selection is automatically updated when you click the **Create run** button, based on the current project and its pipeline.

RHODS-8249 - Environment variables uploaded as ConfigMap were stored in Secret instead

Previously, in the OpenShift AI interface, when you added environment variables to a workbench by uploading a **ConfigMap** configuration, the variables were stored in a **Secret** object instead. This issue is now resolved.

RHODS-7975 - Workbenches could have multiple data connections

Previously, if you changed the data connection for a workbench, the existing data connection was not released. As a result, a workbench could stay connected to multiple data sources. This issue is now resolved.

RHODS-7948 - Uploading a secret file containing environment variables resulted in double-encoded values

Previously, when creating a workbench in a data science project, if you uploaded a YAML-based secret file containing environment variables, the environment variable values were not decoded. Then, in the resulting OpenShift secret created by this process, the encoded values were encoded again. This issue is now resolved.

RHODS-6429 - An error was displayed when creating a workbench with the Intel OpenVINO or Anaconda Professional Edition images

Previously, when you created a workbench with the Intel OpenVINO or Anaconda Professional Edition images, an error appeared during the creation process. However, the workbench was still successfully created. This issue is now resolved.

RHODS-6372 - Idle notebook culler did not take active terminals into account

Previously, if a notebook image had a running terminal, but no active, running kernels, the idle notebook culler detected the notebook as inactive and stopped the terminal. This issue is now resolved.

RHODS-5700 - Data connections could not be created or connected to when creating a workbench

When creating a workbench, users were unable to create a new data connection, or connect to existing data connections.

RHODS-6281 - OpenShift AI administrators could not access Settings page if an admin group was deleted from cluster

Previously, if a Red Hat OpenShift AI administrator group was deleted from the cluster, OpenShift AI administrator users could no longer access the **Settings** page on the OpenShift AI dashboard. In particular, the following behavior was seen:

- When an OpenShift AI administrator user tried to access the **Settings → User management** page, a "Page Not Found" error appeared.
- Cluster administrators *did not* lose access to the **Settings** page on the OpenShift AI dashboard. When a cluster administrator accessed the **Settings → User management** page, a warning message appeared, indicating that the deleted OpenShift AI administrator group no longer existed in OpenShift. The deleted administrator group was then removed from **OdhDashboardConfig**, and administrator access was restored.

This issue is now resolved.

RHODS-1968 - Deleted users stayed logged in until dashboard was refreshed

Previously, when a user's permissions for the Red Hat OpenShift AI dashboard were revoked, the user would notice the change only after a refresh of the dashboard page.

This issue is now resolved. When a user's permissions are revoked, the OpenShift AI dashboard locks the user out within 30 seconds, without the need for a refresh.

RHODS-6384 - A workbench data connection was incorrectly updated when creating a duplicated data connection

When creating a data connection that contained the same name as an existing data connection, the data connection creation failed, but the associated workbench still restarted and connected to the wrong data connection. This issue has been resolved. Workbenches now connect to the correct data connection.

RHODS-6370 - Workbenches failed to receive the latest toleration

Previously, to acquire the latest toleration, users had to attempt to edit the relevant workbench, make no changes, and save the workbench again. Users can now apply the latest toleration change by stopping and then restarting their data science project's workbench.

RHODS-6779 - Models failed to be served after upgrading from OpenShift AI 1.20 to OpenShift AI 1.21

When upgrading from OpenShift AI 1.20 to OpenShift AI 1.21, the **modelmesh-serving** pod attempted to pull a non-existent image, causing an image pull error. As a result, models were unable to be served using the model serving feature in OpenShift AI. The **odh-openvino-servingruntime-container-v1.21.0-15** image now deploys successfully.

RHODS-5945 - Anaconda Professional Edition could not be enabled in OpenShift AI

Anaconda Professional Edition could not be enabled for use in OpenShift AI. Instead, an **InvalidImageName** error was displayed in the associated pod's **Events** page. Anaconda Professional Edition can now be successfully enabled.

RHODS-5822 - Admin users were not warned when usage exceeded 90% and 100% for PVCs created by data science projects.

Warnings indicating when a PVC exceeded 90% and 100% of its capacity failed to display to admin users for PVCs created by data science projects. Admin users can now view warnings about when a PVC exceeds 90% and 100% of its capacity from the dashboard.

RHODS-5889 - Error message was not displayed if a data science notebook was stuck in "pending" status

If a notebook pod could not be created, the OpenShift AI interface did not show an error message. An error message is now displayed if a data science notebook cannot be spawned.

RHODS-5886 - Returning to the Hub Control Panel dashboard from the data science workbench failed

If you attempted to return to the dashboard from your workbench Jupyter notebook by clicking on **File → Log Out**, you were redirected to the dashboard and remained on a "Logging out" page. Likewise, if you attempted to return to the dashboard by clicking on **File → Hub Control Panel**, you were incorrectly redirected to the **Start a notebook server** page. Returning to the Hub Control Panel dashboard from the data science workbench now works as expected.

RHODS-6101 - Administrators were unable to stop all notebook servers

OpenShift AI administrators could not stop all notebook servers simultaneously. Administrators can now stop all notebook servers using the **Stop all servers** button and stop a single notebook by selecting **Stop server** from the action menu beside the relevant user.

RHODS-5891 - Workbench event log was not clearly visible

When creating a workbench, users could not easily locate the event log window in the OpenShift AI interface. The **Starting** label under the **Status** column is now underlined when you hover over it, indicating you can click on it to view the notebook status and the event log.

RHODS-6296 - ISV icons did not render when using a browser other than Google Chrome

When using a browser other than Google Chrome, not all ISV icons under **Explore** and **Resources** pages were rendered. ISV icons now display properly on all supported browsers.

RHODS-3182 - Incorrect number of available GPUs was displayed in Jupyter

When a user attempts to create a notebook instance in Jupyter, the maximum number of GPUs available for scheduling was not updated as GPUs are assigned. Jupyter now displays the correct number of GPUs available.

RHODS-5890 - When multiple persistent volumes were mounted to the same directory, workbenches failed to start

When mounting more than one persistent volume (PV) to the same mount folder in the same workbench, creation of the notebook pod failed and no errors were displayed to indicate there was an issue.

RHODS-5768 - Data science projects were not visible to users in Red Hat OpenShift AI

Removing the **[DSP]** suffix at the end of a project's **Display Name** property caused the associated data science project to no longer be visible. It is no longer possible for users to remove this suffix.

RHODS-5701 - Data connection configuration details were overwritten

When a data connection was added to a workbench, the configuration details for that data connection were saved in environment variables. When a second data connection was added, the configuration details are saved using the same environment variables, which meant the configuration for the first data connection was overwritten. At the moment, users can add a maximum of one data connection to each workbench.

RHODS-5252 - The notebook Administration page did not provide administrator access to a user's notebook server

The notebook **Administration** page, accessed from the OpenShift AI dashboard, did not provide the means for an administrator to access a user's notebook server. Administrators were restricted to only starting or stopping a user's notebook server.

RHODS-2438 - PyTorch and TensorFlow images were unavailable when upgrading

When upgrading from OpenShift AI 1.3 to a later version, PyTorch and TensorFlow images were unavailable to users for approximately 30 minutes. As a result, users were unable to start PyTorch and TensorFlow notebooks in Jupyter during the upgrade process. This issue has now been resolved.

RHODS-5354 - Environment variable names were not validated when starting a notebook server

Environment variable names were not validated on the **Start a notebook server** page. If an invalid environment variable was added, users were unable to successfully start a notebook. The environmental variable name is now checked in real-time. If an invalid environment variable name is entered, an error message displays indicating valid environment variable names must consist of alphabetic characters, digits, _, -, or ., and must not start with a digit.

RHODS-4617 - The Number of GPUs drop-down was only visible if there were GPUs available

Previously, the **Number of GPUs** drop-down was only visible on the **Start a notebook server** page if GPU nodes were available. The **Number of GPUs** drop-down now also correctly displays if an autoscaling machine pool is defined in the cluster, even if no GPU nodes are currently available, possibly resulting in the provisioning of a new GPU node on the cluster.

RHODS-5420 - Cluster admin did not get administrator access if it was the only user present in the cluster

Previously, when the cluster admin was the only user present in the cluster, it did not get Red Hat OpenShift administrator access automatically. Administrator access is now correctly applied to the cluster admin user.

RHODS-4321 - Incorrect package version displayed during notebook selection

The **Start a notebook server** page displayed an incorrect version number (11.4 instead of 11.7) for the CUDA notebook image. The version of CUDA installed is no longer specified on this page.

RHODS-5001 - Admin users could add invalid tolerations to notebook pods

An admin user could add invalid tolerations on the **Cluster settings** page without triggering an error. If a invalid toleration was added, users were unable to successfully start notebooks. The toleration key is now checked in real-time. If an invalid toleration name is entered, an error message displays indicating valid toleration names consist of alphanumeric characters, -, _, or ., and must start and end with an alphanumeric character.

RHODS-5100 - Group role bindings were not applied to cluster administrators

Previously, if you had assigned cluster admin privileges to a group rather than a specific user, the dashboard failed to recognize administrative privileges for users in the administrative group. Group role bindings are now correctly applied to cluster administrators as expected.

RHODS-4947 - Old Minimal Python notebook image persisted after upgrade

After upgrading from OpenShift AI 1.14 to 1.15, the older version of the Minimal Python notebook persisted, including all associated package versions. The older version of the Minimal Python notebook no longer persists after upgrade.

RHODS-4935 - Excessive "missing x-forwarded-access-token header" error messages displayed in dashboard log

The **rhods-dashboard** pod's log contained an excessive number of "missing x-forwarded-access-token header" error messages due to a readiness probe hitting the /**status** endpoint. This issue has now been resolved.

RHODS-2653 - Error occurred while fetching the generated images in the sample Pachyderm notebook

An error occurred when a user attempted to fetch an image using the sample Pachyderm notebook in Jupyter. The error stated that the image could not be found. Pachyderm has corrected this issue.

RHODS-4584 - Jupyter failed to start a notebook server using the OpenVINO notebook image

Jupyter's **Start a notebook server** page failed to start a notebook server using the OpenVINO notebook image. Intel has provided an update to the OpenVINO operator to correct this issue.

RHODS-4923 - A non-standard check box displayed after disabling usage data collection

After disabling usage data collection on the **Cluster settings** page, when a user accessed another area of the OpenShift AI dashboard, and then returned to the **Cluster settings** page, the **Allow collection of usage data** check box had a non-standard style applied, and therefore did not look the same as other check boxes when selected or cleared.

RHODS-4938 - Incorrect headings were displayed in the Notebook Images page

The **Notebook Images** page, accessed from the **Settings** page on the OpenShift AI dashboard, displayed incorrect headings in the user interface. The **Notebook image settings** heading displayed as **BYON image settings**, and the **Import Notebook images** heading displayed as **Import BYON images**. The correct headings are now displayed as expected.

RHODS-4818 - Jupyter was unable to display images when the NVIDIA GPU add-on was installed

The **Start a notebook server** page did not display notebook images after installing the NVIDIA GPU add-on. Images are now correctly displayed, and can be started from the **Start a notebook server** page.

RHODS-4797 - PVC usage limit alerts were not sent when usage exceeded 90% and 100%

Alerts indicating when a PVC exceeded 90% and 100% of its capacity failed to be triggered and sent. These alerts are now triggered and sent as expected.

RHODS-4366 - Cluster settings were reset on operator restart

When the OpenShift AI operator pod was restarted, cluster settings were sometimes reset to their default values, removing any custom configuration. The OpenShift AI operator was restarted when a new version of OpenShift AI was released, and when the node that ran the operator failed. This issue occurred because the operator deployed ConfigMaps incorrectly. Operator deployment instructions have been updated so that this no longer occurs.

RHODS-4318 - The OpenVINO notebook image failed to build successfully

The OpenVINO notebook image failed to build successfully and displayed an error message. This issue has now been resolved.

RHODS-3743 - Starburst Galaxy quick start did not provide download link in the instruction steps

The Starburst Galaxy quick start, located on the **Resources** page on the dashboard, required the user to open the **explore-data.ipynb notebook**, but failed to provide a link within the instruction steps. Instead, the link was provided in the quick start's introduction.

RHODS-1974 - Changing alert notification emails required pod restart

Changes to the list of notification email addresses in the Red Hat OpenShift AI Add-On were not applied until after the **rhods-operator** pod and the **prometheus-*** pod were restarted.

RHODS-2738 - Red Hat OpenShift API Management 1.15.2 add-on installation did not successfully complete

For OpenShift AI installations that are integrated with the Red Hat OpenShift API Management 1.15.2 add-on, the Red Hat OpenShift API Management installation process did not successfully obtain the SMTP credentials secret. Subsequently, the installation did not complete.

RHODS-3237 - GPU tutorial did not appear on dashboard

The "GPU computing" tutorial, located at Gtc2018-numba, did not appear on the **Resources** page on the dashboard.

RHODS-3069 - GPU selection persisted when GPU nodes were unavailable

When a user provisioned a notebook server with GPU support, and the utilized GPU nodes were subsequently removed from the cluster, the user could not create a notebook server. This occurred because the most recently used setting for the number of attached GPUs was used by default.

RHODS-3181 - Pachyderm now compatible with OpenShift Dedicated 4.10 clusters

Pachyderm was not initially compatible with OpenShift Dedicated 4.10, and so was not available in OpenShift AI running on an OpenShift Dedicated 4.10 cluster. Pachyderm is now available on and compatible with OpenShift Dedicated 4.10.

RHODS-2160 - Uninstall process failed to complete when both OpenShift AI and OpenShift API Management were installed

When OpenShift AI and OpenShift API Management are installed together on the same cluster, they use the same Virtual Private Cluster (VPC). The uninstall process for these Add-ons attempts to delete the VPC. Previously, when both Add-ons are installed, the uninstall process for one service was blocked because the other service still had resources in the VPC. The cleanup process has been updated so that this conflict does not occur.

RHODS-2747 - Images were incorrectly updated after upgrading OpenShift AI

After the process to upgrade OpenShift AI completed, Jupyter failed to update its notebook images. This was due to an issue with the image caching mechanism. Images are now correctly updating after an upgrade.

RHODS-2425 - Incorrect TensorFlow and TensorBoard versions displayed during notebook selection

The **Start a notebook server** page displayed incorrect version numbers (2.4.0) for TensorFlow and TensorBoard in the TensorFlow notebook image. These versions have been corrected to TensorFlow 2.7.0 and TensorBoard 2.6.0.

RHODS-24339 - **Quick start links did not display for enabled applications**

For some applications, the **Open quick start** link failed to display on the application tile on the **Enabled** page. As a result, users did not have direct access to the quick start tour for the relevant application.

RHODS-2215 - **Incorrect Python versions displayed during notebook selection**

The **Start a notebook server** page displayed incorrect versions of Python for the TensorFlow and PyTorch notebook images. Additionally, the third integer of package version numbers is now no longer displayed.

RHODS-1977 - **Ten minute wait after notebook server start fails**

If the Jupyter leader pod failed while the notebook server was being started, the user could not access their notebook server until the pod restarted, which took approximately ten minutes. This process has been improved so that the user is redirected to their server when a new leader pod is elected. If this process times out, users see a 504 Gateway Timeout error, and can refresh to access their server.

CHAPTER 6. KNOWN ISSUES

This section describes known issues in Red Hat OpenShift AI 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 Dedicated, change the display names of your data science projects to meet Kubernetes resource name standards: use only lowercase alphanumeric characters and hyphens.

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-4572 - Unable to run data science pipelines after install and upgrade in certain circumstances

You are unable to run data science pipelines after installing or upgrading OpenShift AI in the following circumstances:

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

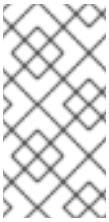
  Workaround

  As a workaround, perform the following steps:

  1. In the OpenShift Dedicated web console, click **Operators → Installed Operators** and then click the **Red Hat OpenShift AI Operator**.
  2. Click the **DSC Initialization** tab.
  3. Click the **default-dsci** object.
  4. Click the **YAML** tab.

5. In the **spec** section, change the value of the **managementState** field for **trustedCABundle** to **Managed**, as shown:

```
spec:
  trustedCABundle:
    managementState: Managed
```



NOTE

If you upgraded from OpenShift AI version 2.6 or 2.7 to version 1, you must manually add the **trustedCABundle** field and the **managementState** field as they are not present in the YAML code. In addition, you do not need to enter a value in the **customCABundle** field.

6. Click **Save**.

7. Restart the dashboard replicaset.

   a. In the OpenShift web console, switch to the **Administrator** perspective.

   b. Click **Workloads → Deployments**.

   c. Set the **Project** to **All Projects** or **redhat-ods-applications** to ensure you can see the appropriate deployment.

   d. Search for the **rhods-dashboard** deployment.

   e. Click the action menu (⋮) and select **Restart Rollout** from the list.

   f. Wait until the **Status** column indicates that all pods in the rollout have fully restarted.

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

Workaround

After configuring self-signed certificates centrally, they apply to any new workbenches and are available at **/etc/pki/tls/certs/** with the **custom** prefix. You can force the tools in your workbench to use these certificates by setting a known environment variable that points to your certificate path.

- If you used **ca-bundle.crt** when you configured certificates centrally, your path is **/etc/pki/tls/certs/custom-ca-bundle.crt**.
- If you used **odh-ca-bundle.crt** when you configured certificates centrally, your path is **/etc/pki/tls/certs/custom-odh-ca-bundle.crt**.

Set a known environment variable:

1. From the OpenShift AI dashboard, go to **Data Science Projects** and select the name of the project containing your workbench.
2. In the **Workbenches** section, click the action menu (⋮) beside the workbench that you want to update, and click **Edit workbench**.
3. Click the **Environment variables** tab.
4. Click **Add variable**.
5. From the **Select environment variable type** dropdown list, select **ConfigMap**.
6. In the **Key** field, enter **SSL_CERT_FILE**.
7. In the **Value** field, enter the path to your certificate file. For example, **/etc/pki/tls/certs/custom-ca-bundle.crt**.
8. Click **Update workbench**.

For more information, see How to execute a pipeline from a Jupyter notebook in a disconnected environment.

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.

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 Dedicated. 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 Dedicated.

See Image Registry Operator in OpenShift Dedicated 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 Dedicated 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.
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.

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-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-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-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-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 the OpenShift AI managed service are described in Uninstalling

OpenShift AI.

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 the current release, 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.