Red Hat OpenShift AI Self-Managed 2.6

Release notes

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

Red Hat OpenShift AI Self-Managed 2.6 Release notes

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

Legal Notice

Abstract

These release notes provide an overview of new features, enhancements, resolved issues, and known issues in version 2.6 of Red Hat OpenShift AI.

Table of Contents

CHAPTER 1. OVERVIEW OF OPENSHIFT AI

Using Red Hat OpenShift AI, users can integrate data, artificial intelligence and machine learning software to execute end-to-end machine learning workflows. OpenShift AI is supported in two configurations:

1. Installed as an Add-on to a Red Hat managed environment such as Red Hat OpenShift Dedicated and Red Hat OpenShift Service on Amazon Web Services (ROSA).

2. Installed as a self-managed Operator on a self-managed environment, such as Red Hat OpenShift Container Platform.

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 2.6.

2.1. NEW FEATURES

HabanaAI notebook image

The HabanaAI notebook image optimizes high-performance deep learning (DL) with Habana Gaudi devices. Habana Gaudi devices accelerate DL training workloads and maximize training throughput and efficiency. Notebook images available on Red Hat OpenShift AI are pre-built and ready for you to use immediately after you install or upgrade OpenShift AI. You can use this feature with HabanaAI Operator 1.10 only. Red Hat OpenShift AI 2.6 now includes the accelerator profiles administrator UI. With this UI, you can easily manage your accelerator profiles and configure custom notebooks to display recommended accelerators.

Accelerator profiles

An *accelerator* is a specialized hardware component that increases the efficiency, speed, and scalability of compute-intensive tasks. For Red Hat OpenShift AI, the supported accelerators are NVIDIA graphics processing units (GPUs) and Habana Gaudi devices (Gaudi 1 and Gaudi 2). An *accelerator profile* defines the specification of an accelerator. Before you can use an accelerator in OpenShift AI, your OpenShift instance must contain the associated accelerator profile. Administrators can configure Red Hat OpenShift AI to enable users to select a specific type of accelerator that is most appropriate for a workload, in addition to specifying the preferred number of accelerators. For information on how to manually create an accelerator profile, see Working with accelerator profiles. The OpenShift AI upgrade process automatically creates an accelerator profile for existing NVIDIA GPUs. This profile can be modified after the upgrade.

TrustyAI

TrustyAI is an open-source AI Explainability (XAI) Toolkit that offers comprehensive explanations of predictive models in both enterprise and data science applications. The core features of TrustyAI include a range of explainers such as local, global, and time-series explainers, alongside unique tools for assessing fairness and drift metrics. Capabilities such as drift detection and bias detection can help organizations to deliver higher quality, unbiased AI-based applications. For more information, see Monitoring Data Science Models . Note that TrustyAI features are supported only with ModelMesh serving.

Standalone TGIS runtime

Red Hat OpenShift AI now includes standalone TGIS runtime support, improving deployment of LLMs by removing the need for Caikit conversion.

2.2. ENHANCEMENTS

This section describes enhancements to existing features in Red Hat OpenShift AI.

Red Hat OpenShift Data Science Operator renamed to Red Hat OpenShift AI Operator

In earlier releases, the product name was Red Hat OpenShift Data Science. The product name is now Red Hat OpenShift AI. Only the name has changed; capabilities and support remain the same.

CHAPTER 3. TECHNOLOGY PREVIEW FEATURES
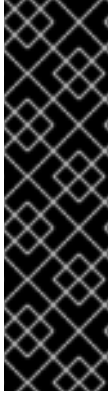
> **IMPORTANT**
>
> This section describes Technology Preview features in Red Hat OpenShift AI 2.6. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
>
> For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

Distributed workloads

Distributed workloads enable data scientists to use multiple cluster nodes in parallel for faster, more efficient data processing and model training. The CodeFlare framework simplifies task orchestration and monitoring, and offers seamless integration for automated resource scaling and optimal node utilization with advanced GPU support. Designed for data scientists, the CodeFlare framework enables direct workload configuration from Jupyter Notebooks or Python code, ensuring a low barrier of adoption, and streamlined, uninterrupted workflows. Distributed workloads significantly reduce task completion time, and enable the use of larger datasets and more complex models. The distributed workloads feature is currently available in Red Hat OpenShift AI 2.6 as a Technology Preview feature. This feature was first introduced in OpenShift AI 2.4.

code-server notebook image

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

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

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

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

CHAPTER 4. SUPPORT REMOVALS

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

4.1. TRUSTYAI'S KUBERNETES APIVERSION RENAMED TO TRUSTYAI.OPENDATAHUB.IO

The TrustyAI Kubernetes apiVersion is no longer **trustyai.opendatahub.io.trustyai.opendatahub.io**, and has been renamed to **trustyai.opendatahub.io**.

If you deployed TrustyAI manually and you upgrade from OpenShift AI 2.5 to 2.6, you must delete and recreate your TrusyAIService custom resource (CR) using **trustyai.opendatahub.io**, and then re-add it to your data science project.

See Enabling the TrustyAI Service for a data science project for more information on creating and adding your TrustyAIService CR.

4.2. UPCOMING DEPRECATION OF DATA SCIENCE PIPELINES V1

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

Data science pipelines in upcoming releases will be based on Kubeflow Pipelines v2, using a different engine. OpenShift AI 2.6 is a stable release that will be supported for 3 months. We recommend that current data science pipeline users stay on OpenShift AI 2.6 until you are ready to migrate to the new pipelines solution.

For a detailed view of the 2.6 release lifecycle, including its full support phase window, see Red Hat OpenShift AI Self-Managed 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. BETA SUBSCRIPTION CHANNEL NO LONGER USED

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

CHAPTER 5. RESOLVED ISSUES

This section describes notable issues that have been resolved in Red Hat OpenShift AI 2.6.

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 the Red Hat Knowledgebase solution article How to use self-signed certificates with KServe .

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-1003 - Services existing in the same project as KServe become inaccessible

When non-service mesh components like Minio services and KServe were deployed together in a namespace enrolled in the service mesh, the components were not accessible via routes. This issue is now resolved.

RHOAIENG-2184 - Non-cluster-admin users are unable to interact with normal Ray resources

Previously, you couldn't interact with Ray resources in namespaces where you had editor or administrator access. This issue is now resolved.

CHAPTER 6. KNOWN ISSUES

This section describes known issues in Red Hat OpenShift AI 2.6 and any known methods of working around these issues.

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

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

Workaround

> Use a different browser.

RHOAIENG-404 - "No Components Found" page randomly appears instead of Enabled page in RHODS dashboard

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

Workaround

> Refresh the browser page.

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

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.

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()
```

RHOAIENG-1452 - The Red Hat OpenShift AI Add-on gets stuck

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-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.

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

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

Workaround

Check the number of replicas using the following CLI command:

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

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

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

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

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-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.

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-2479 – ModelMesh monitoring stack are not deleted during upgrade from 2.4 or 2.5 to 2.6

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, 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-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-2602 – "Average response time" server metric graph shows multiple lines due to ModelMesh pod restart

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

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 **Model serving** 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**.

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-88 - Cannot log in to Red Hat OpenShift AI dashboard

Sometimes, when you try to log in to Red Hat OpenShift AI, the **500 internal error** error message is shown.

Workaround

Disable and re-enable the **dashboard** component in the **DataScienceCluster** object.

RHOAIENG-130 - Synchronization issue when the model is just launched

When the status of the KServe container is **Ready**, a request is accepted even though the TGIS container is not ready.

Workaround

Wait a few seconds to ensure that all initialization has completed and the TGIS container is actually ready, and then review the request output.

RHOAIENG-131 - gRPC endpoint not responding properly after the InferenceService reports as Loaded

When numerous **InferenceService** instances are generated and directed requests, Service Mesh Control Plane (SMCP) becomes unresponsive. The status of the **InferenceService** instance is **Loaded**, but the call to the gRPC endpoint returns with errors.

Workaround

Edit the **ServiceMeshControlPlane** custom resource (CR) to increase the memory limit of the Istio egress and ingress pods.

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-282 - Workload should not be dispatched if required resources are not available

Sometimes a workload is dispatched even though a single machine instance does not have sufficient resources to provision the RayCluster successfully. The **AppWrapper** CRD remains in a **Running** state and related pods are stuck in a **Pending** state indefinitely.

Workaround

Add extra resources to the cluster.

[RHOAIENG-307](#) - **Removing the DataScienceCluster deletes all OpenShift Serverless CRs**

If you delete the **DataScienceCluster** custom resource (CR), all OpenShift Serverless CRs (including knative-serving, deployments, gateways, and pods) are also deleted. A warning message is not shown.

[RHOAIENG-339](#) - **KServe component images are not updated after upgrade to 2.5**

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

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

   ```
   $ oc login
   ```

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

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

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

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

4. Update the **InferenceServices** CRD as follows:

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

5. Optionally, restart the Operator pod.

[RHOAIENG-497](#) - **Removing DSCI Results In OpenShift Service Mesh CR Being Deleted Without User Notification**

If you delete the **DSCInitialization** resource, the OpenShift Service Mesh CR is also deleted. A warning message is not shown.

[RHOAIENG-499](#) - **Uninstalling Red Hat OpenShift AI Self Managed by using the CLI does not uninstall**

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

Workaround

Manually delete the remaining OpenShift AI resources as follows:

1. Delete the **DataScienceCluster** CR.

2. Wait until all pods are deleted from the **redhat-ods-applications** namespace.

3. If Serverless was set to **Managed** in the **DataScienceCluster** CR, wait until all pods are deleted from the **knative-serving** namespace.

4. Delete the **DSCInitialization** CR.

5. If Service Mesh was set to **Managed** in the **DSCInitialization** CR, wait until all pods are deleted from the **istio-system** namespace.

6. Uninstall the Red Hat OpenShift AI Operator.

7. Wait until all pods are deleted from the **redhat-ods-operator** namespace and the **redhat-ods-monitoring** namespace.

RHOAIENG-545 – Cannot specify a generic default node runtime image in JupyterLab pipeline editor

When you edit an Elyra pipeline in the JupyterLab IDE pipeline editor, and you click the **PIPELINE PROPERTIES** tab, and scroll to the **Generic Node Defaults** section and edit the **Runtime Image** field, your changes are not saved.

Workaround

Define the required runtime image explicitly for each node. Click the **NODE PROPERTIES** tab, and specify the required image in the **Runtime Image** field.

DATA-SCIENCE-PIPELINES-165 – Poor error message when S3 bucket is not writable

When you set up a data connection and the S3 bucket is not writable, and you try to upload a pipeline, the error message **Failed to store pipelines** is not helpful.

Workaround

Verify that your data connection credentials are correct and that you have write access to the bucket you specified.

KUBEFLOW-177 – Bearer token from application not forwarded by OAuth-proxy

You cannot use an application as a custom workbench image if its internal authentication mechanism is based on a bearer token. The OAuth-proxy configuration removes the bearer token from the headers, and the application cannot work properly.

ODH-DASHBOARD-1335 – Rename Edit permission to Contributor

The term *Edit* is not accurate:

- For *most* resources, users with the **Edit** permission can not only edit the resource, they can also create and delete the resource.

- Users with the **Edit** permission cannot edit the project.

The term *Contributor* more accurately describes the actions granted by this permission.

ODH-DASHBOARD-1758 - Error duplicating OOTB custom serving runtimes several times

If you duplicate a model-serving runtime several times, the duplication fails with the **Serving runtime name "<name>" already exists** error message.

Workaround

> Change the **metadata.name** field to a unique value.

ODH-DASHBOARD-1771 - JavaScript error during Pipeline step initializing

Sometimes the pipeline **Run details** page stops working when the run starts.

Workaround

> Refresh the page.

ODH-DASHBOARD-1781 - Missing tooltip for Started Run status

Data science pipeline runs sometimes don't show the tooltip text for the status icon shown.

Workaround

> For more information, view the pipeline **Run details** page and see the run output.

ODH-DASHBOARD-1908 - Cannot create workbench with an empty environment variable

When creating a workbench, if you click **Add variable** but do not select an environment variable type from the list, you cannot create the workbench. The field is not marked as required, and no error message is shown.

ODH-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-2140 - Package versions displayed in dashboard do not match installed versions

The dashboard might display inaccurate version numbers for packages such as JupyterLab and Notebook. The package version number can differ in the image if the packages are manually updated.

Workaround

> To find the true version number for a package, run the **pip list** command and search for the package name, as shown in the following examples:

```
$ pip list | grep jupyterlab
jupyterlab                3.5.3
$ pip list | grep notebook
notebook                  6.5.3
```

RHODS-12432 - Deletion of the notebook-culler ConfigMap causes Permission Denied on dashboard

If you delete the **notebook-controller-culler-config** ConfigMap in the **redhat-ods-applications** namespace, you can no longer save changes to the **Cluster Settings** page on the OpenShift AI dashboard. The save operation fails with an **HTTP request has failed** error.

Workaround

Complete 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 to update the **OdhDashboardConfig** custom resource in the **redhat-ods-applications** application namespace:

   ```
   $ oc patch OdhDashboardConfig odh-dashboard-config -n redhat-ods-applications --type=merge -p '{"spec": {"dashboardConfig": {"notebookController.enabled": true}}}'
   ```

RHODS-12798 - Pods fail with "unable to init seccomp" error

Pods fail with **CreateContainerError** status or **Pending** status instead of **Running** status, because of a known kernel bug that introduced a **seccomp** memory leak. When you check the events on the namespace where the pod is failing, or run the **oc describe pod** command, the following error appears:

```
runc create failed: unable to start container process: unable to init seccomp: error loading seccomp filter into kernel: error loading seccomp filter: errno 524
```

Workaround

Increase the value of **net.core.bpf_jit_limit** as described in the Red Hat Knowledgebase solution Pods failing with error loading seccomp filter into kernel: errno 524 in OpenShift 4 .

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. The accelerator profiles feature is currently available in Red Hat OpenShift AI as a Technology Preview feature. See Technology Preview features.

RHODS-12903 - Successfully-submitted Elyra pipeline fails to run

If you use a private TLS certificate, and you successfully submit an Elyra-generated pipeline against the data science pipeline server, the pipeline steps fail to execute, and the following error messages are shown:

```
File "/opt/app-root/src/bootstrapper.py", line 747, in <module>
main()
File "/opt/app-root/src/bootstrapper.py", line 730, in main
```

> Actions
> ...
> WARNING: Retrying (Retry (total-4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor.urllib3.connection.HTTPSConnection obj
> In this situation, a new runtime image should be created, to include the correct CA bundle, as well as all the required pip packages.

Workaround

Contact Red Hat Support for detailed steps to resolve this issue.

RHODS-12904 – Pipeline submitted from Elyra might fail when using private certificate

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

- The object storage used for the pipeline server is using private TLS certificates.

- The data science pipeline server API endpoint is using private TLS certificates.

Workaround

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

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

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

Workaround

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

RHODS-12928 – Using unsupported characters can generate Kubernetes resource names with multiple dashes

When you create a resource and you specify unsupported characters in the name, then each space is replaced with a dash and other unsupported characters are removed, which can result in an invalid resource name.

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

In disconnected environments, after upgrade to Red Hat OpenShift AI 2.6, previously deployed model servers might no longer work. The model status might be incorrectly reported as **OK** on the dashboard.

Workaround

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

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

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

Example:

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

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

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

3. Capture the secret key name as follows:

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

4. Update the annotation as follows:

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

[RHODS-12946](#) – **Cannot install from PyPI mirror in disconnected environment or when using private certificates**

In disconnected environments, Red Hat OpenShift AI cannot connect to the public–facing PyPI repositories, so you must specify a repository inside your network. If you are using private TLS certificates, and a data science pipeline is configured to install Python packages, the pipeline run fails.

Workaround

Add the required environment variables and certificates to your pipeline, as described in the Red Hat Knowledgebase solution [Install packages from PyPI Mirror fails on Data Science Pipelines in disconnected installation](#).

[RHODS-12986](#) – **Potential reconciliation error after upgrade to Red Hat OpenShift AI 2.6**

After you upgrade to Red Hat OpenShift AI 2.6, a reconciliation error might appear in the Red Hat OpenShift AI Operator pod logs and in the **DataScienceCluster** custom resource (CR) conditions.

Example error:

```
2023-11-23T09:45:37Z    ERROR    Reconciler error    {"controller": "datasciencecluster",
"controllerGroup": "datasciencecluster.opendatahub.io", "controllerKind": "DataScienceCluster",
"DataScienceCluster": {"name":"default-dsc"}, "namespace": "", "name": "default-dsc", "reconcileID":
"0c1a32ca-7ffd-4310-8259-f6baabf3c868", "error": "1 error occurred:\n\t* Deployment.apps \"rhods-
prometheus-operator\" is invalid: spec.selector: Invalid value:
v1.LabelSelector{MatchLabels:map[string]string{\"app.kubernetes.io/part-of\":\"model-mesh\",
\"app.opendatahub.io/model-mesh\":\"true\", \"k8s-app\":\"rhods-prometheus-operator\"},
MatchExpressions:[]v1.LabelSelectorRequirement(nil)}: field is immutable\n\n"}
```

Workaround

Restart the Red Hat OpenShift AI Operator pod.

RHOAIENG-11 - Separately installed instance of CodeFlare Operator not supported

In Red Hat OpenShift AI, the CodeFlare Operator is included in the base product and not in a separate Operator. Separately installed instances of the CodeFlare Operator from Red Hat or the community are not supported.

Workaround

Delete any installed CodeFlare Operators, and install and configure Red Hat OpenShift AI, as described in the Red Hat Knowledgebase solution How to migrate from installed CodeFlare Operator to Red Hat OpenShift Data Science.

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**.

RHOAIENG-52 - Token authentication fails in clusters with self-signed certificates

If you use self-signed certificates, and you use the Python **codeflare-sdk** in a notebook or in a Python script as part of a pipeline, token authentication will fail.

RHOAIENG-133 - Existing workbench cannot run Elyra pipeline after notebook restart

If you use the Elyra JupyterLab extension to create and run data science pipelines within JupyterLab, and you configure the pipeline server *after* you created a workbench and specified a notebook image within the workbench, you cannot execute the pipeline, even after restarting the notebook.

Workaround

1. Stop the running notebook.

2. Edit the workbench to make a small modification. For example, add a new dummy environment variable, or delete an existing unnecessary environment variable. Save your changes.

3. Restart the notebook.

4. In the left sidebar of JupyterLab, click **Runtimes**.

5. Confirm that the default runtime is selected.

NOTEBOOKS-218 - Data science pipelines saved from the Elyra pipeline editor reference an incompatible runtime

When you save a pipeline in the Elyra pipeline editor with the format **.pipeline** in OpenShift AI version 1.31 or earlier, the pipeline references a runtime that is incompatible with OpenShift AI version 1.32 or later.

As a result, the pipeline fails to run after you upgrade OpenShift AI to version 1.32 or later.

Workaround

After you upgrade to OpenShift AI to version 1.32 or later, select the relevant runtime images again.

NOTEBOOKS-210 - A notebook fails to export as a PDF file in Jupyter

When you export a notebook as a PDF file in Jupyter, the export process fails with an error.

DATA-SCIENCE-PIPELINES-OPERATOR-349 - The Import Pipeline button is prematurely accessible

When you import a pipeline to a workbench that belongs to a data science project, the **Import Pipeline** button is prematurely accessible before the pipeline server is fully available.

Workaround

Refresh your browser page and import the pipeline again.

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.

ODH-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.

RHODS-11791 - Usage data collection is enabled after upgrade

If you previously had the **Allow collection of usage data** option deselected (that is, disabled), this option becomes selected (that is, enabled) when you upgrade OpenShift AI.

Workaround

Manually reset the **Allow collection of usage data** option. To do this, perform the following actions:

1. In the OpenShift AI dashboard, in the left menu, click **Settings → Cluster settings**. The **Cluster Settings** page opens.

2. In the **Usage data collection** section, deselect **Allow collection of usage data**.

3. Click **Save changes**.

ODH-DASHBOARD-1741 - Cannot create a workbench whose name begins with a number

If you try to create a workbench whose name begins with a number, the workbench does not start.

Workaround

Delete the workbench and create a new one with a name that begins with a letter.

RHODS-6913 (ODH-DASHBOARD-1699) – Workbench does not automatically restart for all configuration changes

When you edit the configuration settings of a workbench, a warning message appears stating that the workbench will restart if you make any changes to its configuration settings. This warning is misleading because in the following cases, the workbench does not automatically restart:

- Edit name

- Edit description

- Edit, add, or remove keys and values of existing environment variables

Workaround

Manually restart the workbench.

KUBEFLOW-157 – Logging out of JupyterLab does not work if you are already logged out of the OpenShift AI dashboard

If you log out of the OpenShift AI dashboard before you log out of JupyterLab, then logging out of JupyterLab is not successful. For example, if you know the URL for a Jupyter notebook, you are able to open this again in your browser.

Workaround

Log out of JupyterLab before you log out of the OpenShift AI dashboard.

RHODS-9789 – Pipeline servers fail to start if they contain a custom database that includes a dash in its database name or username field

When you create a pipeline server that uses a custom database, if the value that you set for the **dbname** field or **username** field includes a dash, the pipeline server fails to start.

Workaround

Edit the pipeline server to omit the dash from the affected fields.

RHODS-9412 – Elyra pipeline fails to run if workbench is created by a user with edit permissions

If a user who has been granted edit permissions for a project creates a project workbench, that user sees the following behavior:

- During the workbench creation process, the user sees an **Error creating workbench** message related to the creation of Kubernetes role bindings.

- Despite the preceding error message, OpenShift AI still creates the workbench. However, the error message means that the user will not be able to use the workbench to run Elyra data science pipelines.

- If the user tries to use the workbench to run an Elyra pipeline, Jupyter shows an **Error making request** message that describes failed initialization.

  Workaround

A user with administrator permissions (for example, the project owner) must create the workbench on behalf of the user with edit permissions. That user can then use the workbench to run Elyra pipelines.

RHODS-8921 – **You cannot create a pipeline server when cumulative character limit is exceeded**

When the cumulative character limit of a data science project name and a pipeline server name exceeds 62 characters, you are unable to successfully create a pipeline server.

Workaround

Rename your data science project so that it does not exceed 30 characters.

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-7718 – **User without dashboard permissions is able to continue using their running notebooks and workbenches indefinitely**

When a Red Hat OpenShift AI administrator revokes a user's permissions, the user can continue to use their running notebooks and workbenches indefinitely.

Workaround

When the OpenShift AI administrator revokes a user's permissions, the administrator should also stop any running notebooks and workbenches for that user.

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-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-6955 – **An error can occur when trying to edit a workbench**

When editing a workbench, an error similar to the following can occur:

> Error creating workbench
> Operation cannot be fulfilled on notebooks.kubeflow.org "workbench-name": the object has been modified; please apply your changes to the latest version and try again

RHODS-6383 – **An ImagePullBackOff error message is not displayed when required during the workbench creation process**

Pods can experience issues pulling container images from the container registry. If an error occurs, the relevant pod enters into an **ImagePullBackOff** state. During the workbench creation process, if an **ImagePullBackOff** error occurs, an appropriate message is not displayed.

Workaround

> Check the event log for further information on the **ImagePullBackOff** error. To do this, click on the workbench status when it is starting.

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-6356 – The notebook creation process fails for users who have never logged in to the dashboard

The dashboard's notebook **Administration** page displays users belonging to the user group and admin group in OpenShift. However, if an administrator attempts to start a notebook server on behalf of a user who has never logged in to the dashboard, the server creation process fails and displays the following error message:

> Request invalid against a username that does not exist.

Workaround

> Request that the relevant user logs into the dashboard.

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-5906 – The NVIDIA GPU Operator is incompatible with OpenShift 4.11.12

Provisioning a GPU node on a OpenShift 4.11.12 cluster results in the **nvidia-driver-daemonset** pod getting stuck in a CrashLoopBackOff state. The NVIDIA GPU Operator is compatible with OpenShift 4.11.9 and 4.11.13.

RHODS-5763 – Incorrect package version displayed during notebook selection

The **Start a notebook server** page displays an incorrect version number for the Anaconda notebook image.

RHODS-5543 – When using the NVIDIA GPU Operator, more nodes than needed are created by the Node Autoscaler

When a pod cannot be scheduled due to insufficient available resources, the Node Autoscaler creates a new node. There is a delay until the newly created node receives the relevant GPU workload. Consequently, the pod cannot be scheduled and the Node Autoscaler's continuously creates additional new nodes until one of the nodes is ready to receive the GPU workload. For more information about this issue, see the Red Hat Knowledgebase solution When using the NVIDIA GPU Operator, more nodes than needed are created by the Node Autoscaler.

Workaround

> Apply the **cluster-api/accelerator** label in **machineset.spec.template.spec.metadata**. This causes the autoscaler to consider those nodes as unready until the GPU driver has been deployed.

RHODS-5216 – The application launcher menu incorrectly displays a link to OpenShift Cluster Manager

Red Hat OpenShift AI incorrectly displays a link to the OpenShift Cluster Manager from the application launcher menu. Clicking this link results in a "Page Not Found" error because the URL is not valid.

RHODS-5251 - Notebook server administration page shows users who have lost permission access

If a user who previously started a notebook server in Jupyter loses their permissions to do so (for example, if an OpenShift AI administrator changes the user's group settings or removes the user from a permitted group), administrators continue to see the user's notebook servers on the server **Administration** page. As a consequence, an administrator is able to restart notebook servers that belong to the user who's permissions were revoked.

RHODS-4799 - Tensorboard requires manual steps to view

When a user has TensorFlow or PyTorch notebook images and wants to use TensorBoard to display data, manual steps are necessary to include environment variables in the notebook environment, and to import those variables for use in your code.

Workaround

When you start your notebook server, use the following code to set the value for the TENSORBOARD_PROXY_URL environment variable to use your OpenShift AI user ID.

```
import os
os.environ["TENSORBOARD_PROXY_URL"]= os.environ["NB_PREFIX"]+"/proxy/6006/"
```

RHODS-4718 - The Intel® oneAPI AI Analytics Toolkits quick start references nonexistent sample notebooks

The Intel® oneAPI AI Analytics Toolkits quick start, located on the **Resources** page on the dashboard, requires the user to load sample notebooks as part of the instruction steps, but refers to notebooks that do not exist in the associated repository.

RHODS-4627 - The CronJob responsible for validating Anaconda Professional Edition's license is suspended and does not run daily

The CronJob responsible for validating Anaconda Professional Edition's license is automatically suspended by the OpenShift AI operator. As a result, the CronJob does not run daily as scheduled. In addition, when Anaconda Professional Edition's license expires, Anaconda Professional Edition is not indicated as disabled on the OpenShift AI dashboard.

RHODS-4502 - The NVIDIA GPU Operator tile on the dashboard displays button unnecessarily

GPUs are automatically available in Jupyter after the NVIDIA GPU Operator is installed. The **Enable** button, located on the NVIDIA GPU Operator tile on the **Explore** page, is therefore redundant. In addition, clicking the **Enable** button moves the NVIDIA GPU Operator tile to the **Enabled** page, even if the Operator is not installed.

RHODS-3985 - Dashboard does not display *Enabled page content after ISV operator uninstall

After an ISV operator is uninstalled, no content is displayed on the **Enabled** page on the dashboard. Instead, the following error is displayed:

```
Error loading components
HTTP request failed
```

Workaround

Wait 30-40 seconds and then refresh the page in your browser.

RHODS-3984 - Incorrect package versions displayed during notebook selection

In the OpenShift AI interface, the **Start a notebook server page** displays incorrect version numbers for the JupyterLab and Notebook packages included in the oneAPI AI Analytics Toolkit notebook image. The page might also show an incorrect value for the Python version used by this image.

Workaround

When you start your oneAPI AI Analytics Toolkit notebook server, you can check which Python packages are installed on your notebook server and which version of the package you have by running the **!pip list** command in a notebook cell.

RHODS-2956 - Error can occur when creating a notebook instance

When creating a notebook instance in Jupyter, a **Directory not found** error appears intermittently. This error message can be ignored by clicking **Dismiss**.

RHODS-2881 - Actions on dashboard not clearly visible

The dashboard actions to revalidate a disabled application license and to remove a disabled application tile are not clearly visible to the user. These actions appear when the user clicks on the application tile's **Disabled** label. As a result, the intended workflows might not be clear to the user.

RHODS-2879 - License revalidation action appears unnecessarily

The dashboard action to revalidate a disabled application license appears unnecessarily for applications that do not have a license validation or activation system. In addition, when a user attempts to revalidate a license that cannot be revalidated, feedback is not displayed to state why the action cannot be completed.

RHODS-2650 - Error can occur during Pachyderm deployment

When creating an instance of the Pachyderm operator, a webhook error appears intermittently, preventing the creation process from starting successfully. The webhook error is indicative that, either the Pachyderm operator failed a health check, causing it to restart, or that the operator process exceeded its container's allocated memory limit, triggering an Out of Memory (OOM) kill.

Workaround

Repeat the Pachyderm instance creation process until the error no longer appears.

RHODS-2096 - IBM Watson Studio not available in OpenShift AI

IBM Watson Studio is not available when OpenShift AI is installed on OpenShift Dedicated 4.9 or higher, because it is not compatible with these versions of OpenShift Dedicated. Contact Marketplace support for assistance manually configuring Watson Studio on OpenShift Dedicated 4.9 and higher.

CHAPTER 7. PRODUCT FEATURES

Red Hat OpenShift AI provides a rich set of features for data scientists and IT operations administrators. To learn more, see Introduction to Red Hat OpenShift AI .