



Red Hat OpenShift AI Self-Managed 2.10

Release notes

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

Red Hat OpenShift AI Self-Managed 2.10 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 version 2.10 of Red Hat OpenShift AI.

Table of Contents

CHAPTER 1. OVERVIEW OF OPENSIFT AI	3
CHAPTER 2. NEW FEATURES AND ENHANCEMENTS	4
2.1. NEW FEATURES	4
2.2. ENHANCEMENTS	4
CHAPTER 3. TECHNOLOGY PREVIEW FEATURES	5
CHAPTER 4. DEVELOPER PREVIEW FEATURES	6
CHAPTER 5. LIMITED AVAILABILITY FEATURES	7
CHAPTER 6. SUPPORT REMOVALS	8
6.1. DATA SCIENCE PIPELINES V1 UPGRADED TO V2	8
6.2. EMBEDDED SUBSCRIPTION CHANNEL NO LONGER USED	8
6.3. REMOVAL OF BIAS DETECTION (TRUSTYAI)	8
6.4. VERSION 1.2 NOTEBOOK CONTAINER IMAGES FOR WORKBENCHES ARE NO LONGER SUPPORTED	8
6.5. BETA SUBSCRIPTION CHANNEL NO LONGER USED	9
CHAPTER 7. RESOLVED ISSUES	10
7.1. ISSUES RESOLVED IN RED HAT OPENSIFT AI 2.10	10
CHAPTER 8. KNOWN ISSUES	12
CHAPTER 9. PRODUCT FEATURES	32

CHAPTER 1. OVERVIEW OF OPENSIFT 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-premise 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 enhance your data science projects on OpenShift AI by building portable machine learning (ML) workflows with data science pipelines, using Docker containers. 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 has two deployment options:

- **Self-managed software** that you can install on-premise or in the cloud. You can install OpenShift AI Self-Managed in a self-managed environment such as OpenShift Container Platform, or in Red Hat-managed cloud environments such as Red Hat OpenShift Dedicated (with a Customer Cloud Subscription for AWS or GCP), Red Hat OpenShift Service on Amazon Web Services (ROSA Classic), or Red Hat OpenShift Service on Amazon Web Services with hosted control planes (ROSA HCP).
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](#).
- A **managed cloud service**, installed as an add-on in Red Hat OpenShift Dedicated (with a Customer Cloud Subscription for AWS or GCP) or in Red Hat OpenShift Service on Amazon Web Services (ROSA Classic).
For information about OpenShift AI Cloud Service, see [Product Documentation for Red Hat OpenShift AI](#).

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 2.10.



IMPORTANT

This version of OpenShift AI supports using data science pipelines with DSP 2.0. If you are using OpenShift AI 2.8 and want to continue using data science pipelines with DSP 1.0, Red Hat recommends that you stay on the 2.8 version. For more information, see [Support Removals](#).

2.1. NEW FEATURES

vLLM runtime for KServe

The single-model serving platform (which uses KServe) now includes vLLM, a high-performance, specialized runtime for large language models that uses a special method called paged attention to speed up the execution of several LLMs on certain accelerators.

The vLLM runtime is compatible with the OpenAI API. The initial version of vLLM requires GPUs and is not optimized for CPU inference.

If you have configured monitoring for the single-model serving platform, you can also view vLLM runtime metrics.

Improved dashboard experience

This release introduces a new home page designed for easy access to data science projects, learning resources, and overviews of primary solution areas, making it easier to start using OpenShift AI. The new home page also provides OpenShift AI administrators with access to key functionality, simplifying product configuration. The previous home page is still accessible from the left navigation pane under **Applications > Enabled**.

2.2. ENHANCEMENTS

Removal of internal image registry dependency

You can now create and use workbenches in OpenShift AI without enabling the internal OpenShift Container Platform image registry. If you update a cluster to enable or disable the internal image registry, you must recreate existing workbenches for the registry changes to take effect.

New workbench images for Intel AI Tools integration

The Intel AI Tools (formerly Intel oneAPI AI Analytics Toolkit) integration has been enhanced with three new workbench images, which include optimizations for popular frameworks and libraries such as PyTorch and TensorFlow. These optimizations provide improved performance on Intel hardware, helping data scientists accelerate end-to-end data science and analytics pipelines on Intel architecture.

You must install the Intel AI Tools Operator on your cluster to be able select the new workbench images.

Support for OpenShift AI Self-Managed on ROSA HCP

OpenShift AI for Self-Managed is now [supported](#) on Red Hat OpenShift Service on AWS (ROSA) with hosted control planes (ROSA HCP).

CHAPTER 3. TECHNOLOGY PREVIEW FEATURES



IMPORTANT

This section describes Technology Preview features in Red Hat OpenShift AI 2.10. 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](#).

code-server workbench image

Red Hat OpenShift AI now includes the **code-server** workbench image. See [code-server in GitHub](#) for more information.

With the **code-server** workbench image, you can customize your workbench environment by using a variety of extensions to add new languages, themes, debuggers, and connect to additional services. You can also 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** workbench image.

The **code-server** workbench image is currently available in Red Hat OpenShift AI 2.10 as a Technology Preview feature. This feature was first introduced in OpenShift AI 2.6.

CHAPTER 4. DEVELOPER PREVIEW FEATURES



IMPORTANT

This section describes Developer Preview features in Red Hat OpenShift AI 2.10. Developer Preview features are not supported by Red Hat in any way and are not functionally complete or production-ready. Do not use Developer Preview features for production or business-critical workloads. Developer Preview features provide early access to functionality in advance of possible inclusion in a Red Hat product offering. Customers can use these features to test functionality and provide feedback during the development process. Developer Preview features might not have any documentation, are subject to change or removal at any time, and have received limited testing. Red Hat might provide ways to submit feedback on Developer Preview features without an associated SLA.

For more information about the support scope of Red Hat Developer Preview features, see [Developer Preview Support Scope](#).

KServe Modelcars

The KServe component of OpenShift AI includes Modelcars as a Developer Preview feature. The Modelcars feature streamlines model fetching by using Open Container Initiative (OCI) images that contain your model data. This behavior can decrease startup times for large models, reduce disk space usage, and enhance performance.

The Modelcars feature is not enabled by default. You must modify your KServe configuration to use the feature.

For more information, see [Serving models with OCI images](#) in the KServe documentation and the [Enhancing KServe Model Fetching with Modelcars](#) design document.

Support for AppWrapper in Kueue

AppWrapper support in Kueue is available as a Developer Preview feature. The experimental API enables the use of AppWrapper-based workloads with the distributed workloads feature.

CHAPTER 5. LIMITED AVAILABILITY FEATURES



IMPORTANT

This section describes Limited Availability features in Red Hat OpenShift AI 2.10. Limited Availability means that you can install and receive support for the feature only with specific approval from Red Hat. Without such approval, the feature is unsupported. This applies to all features described in this section.

Tuning in OpenShift AI

Tuning in OpenShift AI is available as a Limited Availability feature. The Kubeflow Training Operator and the Hugging Face Supervised Fine-tuning Trainer (SFT Trainer) enable users to fine-tune and train their models easily in a distributed environment. In this release, you can use this feature for models that are based on the PyTorch machine-learning framework.

CHAPTER 6. SUPPORT REMOVALS

This section describes major changes in support for user-facing features in Red Hat OpenShift AI. For information about OpenShift AI supported software platforms, components, and dependencies, see [Supported configurations](#).

6.1. DATA SCIENCE PIPELINES V1 UPGRADED TO V2

Previously, data science pipelines in OpenShift AI were based on KubeFlow Pipelines v1. Starting with OpenShift AI 2.9, data science pipelines are based on KubeFlow Pipelines v2, which uses a different workflow engine. Data Science Pipelines (DSP) 2.0 is enabled and deployed by default in OpenShift AI. For more information, see [Enabling Data Science Pipelines 2.0](#).



IMPORTANT

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

It is no longer possible to deploy, view, or edit the details of pipelines that are based on Data Science Pipelines (DSP) 1.0 from the dashboard in OpenShift AI 2.10. If you already use data science pipelines, Red Hat recommends that you stay on OpenShift AI 2.8 until full feature parity in DSP 2.0 has been delivered in a stable OpenShift AI release and you are ready to migrate to the new pipeline solution. For a detailed view of the release lifecycle, including the full support phase window, see [Red Hat OpenShift AI Self-Managed Life Cycle](#).



IMPORTANT

If you want to use existing pipelines and workbenches with DSP 2.0 after upgrading to OpenShift AI 2.10, you must update your workbenches to use the 2024.1 notebook image version and then manually migrate your pipelines from DSP 1.0 to 2.0. For more information, see [Upgrading to DSP 2.0](#).

6.2. EMBEDDED SUBSCRIPTION CHANNEL NO LONGER USED

Starting with OpenShift AI 2.8, the **embedded** subscription channel is no longer used. You can no longer select the **embedded** channel for a new installation of the Operator. For more information about subscription channels, see [Installing the Red Hat OpenShift AI Operator](#).

6.3. 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.

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

When you create a workbench, you specify a notebook container image to use with the workbench. Starting with OpenShift AI 2.5, when you create a new workbench, version 1.2 notebook container images are not available to select. Workbenches that are already running with a version 1.2 notebook

image continue to work normally. However, Red Hat recommends that you update your workbench to use the latest notebook container image.

6.5. BETA SUBSCRIPTION CHANNEL NO LONGER USED

Starting with OpenShift AI 2.5, the **beta** subscription channel is no longer used. 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 7. RESOLVED ISSUES

The following notable issues are resolved in Red Hat OpenShift AI 2.10.

Security updates, bug fixes, and enhancements for Red Hat OpenShift AI 2.10 are released as asynchronous errata. All OpenShift AI errata advisories are published on the [Red Hat Customer Portal](#).

7.1. ISSUES RESOLVED IN RED HAT OPENSIFT AI 2.10

For a complete list of updates, see the [RHBA-2024:4069](#) advisory.

RHOAIENG-7312 - Model serving fails during query with token authentication in KServe

Previously, if you enabled both the ModelMesh and KServe components in your **DataScienceCluster** object and added Authorino as an authorization provider, a race condition could occur that resulted in the **odh-model-controller** pods being rolled out in a state that is appropriate for ModelMesh, but not for KServe and Authorino. In this situation, if you made an inference request to a running model that was deployed using KServe, you saw a **404 - Not Found** error. In addition, the logs for the **odh-model-controller** deployment object showed a **Reconciler** error message. This issue is now resolved.

RHOAIENG-7079 (previously documented as RHOAIENG-6317) - Pipeline task status and logs sometimes not shown in OpenShift AI dashboard

Previously, when running pipelines by using Elyra, the OpenShift AI dashboard might not show the pipeline task status and logs, even when the related pods had not been pruned and the information was still available in the OpenShift Console. This issue is now resolved.

RHOAIENG-7070 (previously documented as RHOAIENG-6709) - Jupyter notebook creation might fail when different environment variables specified

Previously, if you started and then stopped a Jupyter notebook, and edited its environment variables in an OpenShift AI workbench, the notebook failed to restart. This issue is now resolved.

RHOAIENG-6853 - Cannot set pod toleration in Elyra pipeline pods

Previously, if you set a pod toleration for an Elyra pipeline pod, the toleration did not take effect. This issue is now resolved.

RHOAIENG-6505 - Disconnected environments: Extra image needed for RayCluster TLS certificate creation

Previously, if you tried to implement mutual Transport Layer Security (mTLS) for distributed workloads in a disconnected environment, you had to add an additional default image (**quay.io/project-codeflare/ray:latest-py39-cu118**) to your mirror registry for TLS certificate creation for Ray Clusters. This issue is now resolved.

RHOAIENG-5314 - Data science pipeline server fails to deploy in fresh cluster due to network policies

Previously, if 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-4252 - Data science pipeline server deletion process fails to remove ScheduledWorkflow resource

Previously, the pipeline server deletion process did not remove the **ScheduledWorkFlow** resource. As a result, new **DataSciencePipelinesApplications** (DSPAs) did not recognize the redundant **ScheduledWorkFlow** resource. This issue is now resolved

RHOAIENG-3411 (previously documented as RHOAIENG-3378) - Internal Image Registry is an undeclared hard dependency for Jupyter notebooks spawn process

Previously, before you could start OpenShift AI notebooks and workbenches, you must have already enabled the internal, integrated container image registry in OpenShift Container Platform. Attempts to start notebooks or workbenches without first enabling the image registry failed with an "InvalidImageName" error. You can now create and use workbenches in OpenShift AI without enabling the internal OpenShift Container Platform image registry. If you update a cluster to enable or disable the internal image registry, you must recreate existing workbenches for the registry changes to take effect.

RHOAIENG-2541 - KServe controller pod experiences OOM because of too many secrets in the cluster

Previously, if your OpenShift cluster had a large number of secrets, the KServe controller pod could continually crash due to an out-of-memory (OOM) error. This issue is now resolved.

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

Previously, the Red Hat OpenShift AI Add-on uninstall did not delete OpenShift AI components when the install was triggered via OCM APIs. This issue is now resolved.

RHOAIENG-307 - Removing the DataScienceCluster deletes all OpenShift Serverless CRs

Previously, if you deleted the DataScienceCluster custom resource (CR), all OpenShift Serverless CRs (including knative-serving, deployments, gateways, and pods) were also deleted. This issue is now resolved.

CHAPTER 8. KNOWN ISSUES

This section describes known issues in Red Hat OpenShift AI 2.10 and any known methods of working around these issues.

RHOAIENG-8819 - `ibm-granite/granite-3b-code-instruct` model fails to deploy on single-model serving platform

If you try to deploy the `ibm-granite/granite-3b-code-instruct` model on the single-model serving platform by using the **vLLM ServingRuntime for KServe** runtime, the model deployment fails with the error shown.

```
ValueError: Head size 80 is not supported by FlashAttention. Supported head sizes are: [32, 64, 96, 128, 160, 192, 224, 256]
```

Workaround

Add the following parameter to the **env** section of the **ServingRuntime** specification for the **vLLM ServingRuntime for KServe** runtime.

```
VLLM_ATTENTION_BACKEND=XFORMERS
```

RHOAIENG-8553 - Workbench created with custom image shows **!Deleted** flag

If you disable the internal image registry on your OpenShift Container Platform cluster and then create a workbench with a custom image that is imported by using the image tag, for example: `quay.io/my-wb-images/my-image:tag`, a **!Deleted** flag is shown in the **Notebook image** column on the **Workbenches** tab of the **Data Science Projects** page. If you stop the workbench, you cannot restart it.

- An OpenShift Container Platform cluster admin can confirm if the internal image registry is enabled on your cluster.
- An OpenShift AI admin user can confirm if a custom image was imported by using the tag notation.

Workaround

Import the custom image using the SHA digest, for example `quay.io/my-repo/my-image@sha256:xxxxxxxxxxxxxx`, and then create the workbench using the custom image.

RHOAIENG-8294 - CodeFlare error when upgrading OpenShift AI 2.8 to version 2.10

If you try to upgrade OpenShift AI 2.8 to version 2.10, the following error message is shown for the CodeFlare component, due to a mismatch with the **AppWrapper** custom resource definition (CRD) version.

```
ReconcileCompletedWithComponentErrors DataScienceCluster resource reconciled with component errors: 1 error occurred: * CustomResourceDefinition.apiextensions.k8s.io "appwrappers.workload.codeflare.dev" is invalid: status.storedVersions[0]: Invalid value: "v1beta1": must appear in spec.versions
```

Workaround

1. Delete the existing **AppWrapper** CRD.

```
$ oc delete crd appwrappers.workload.codeflare.dev
```


-
2. Install the latest version of the **AppWrapper** CRD.

```
$ oc apply -f https://raw.githubusercontent.com/project-codeflare/codeflare-operator/main/config/crd/crd-appwrapper.yml
```

RHOAIENG-8218 - Cannot log in to a workbench created on an OpenShift 4.15 cluster without OCP internal image registry

When you create a workbench on an OpenShift 4.15 cluster that does not have the OpenShift Container Platform internal image registry enabled, the workbench starts successfully, but you cannot log in to it.

A **The authorization server encountered an unexpected condition that prevented it from fulfilling the request** error is displayed.

Workaround

Manually create a service account token secret in OCP for every workbench that you create.

1. From the OpenShift CLI, use the following command to create a service account token secret. Replace **<workbench-name>** with the name of your workbench, and **<project-name>** with the name of your data science project.

```
-----
cat <<'EOF' | oc apply -f -
---
kind: Secret
apiVersion: v1
metadata:
  name: <workbench-name>-token
  namespace: <project-name>
  annotations:
    kubernetes.io/service-account.name: <workbench-name>
type: kubernetes.io/service-account-token
EOF
-----
```

2. Refresh the page in your browser and log in to your workbench.

For information about creating a service account token secret using the web console, see [Creating a service account token secret](#).

RHOAIENG-7947 - Model serving fails during query in KServe

If you initially install the ModelMesh component and enable the multi-model serving platform, but later install the KServe component and enable the single-model serving platform, inference requests to models deployed on the single-model serving platform might fail. In these cases, inference requests return a **404 - Not Found** error and the logs for the **odh-model-controller** deployment object show a **Reconciler** error message.

Workaround

In OpenShift Container Platform, restart the **odh-model-controller** deployment object.

RHOAIENG-7887 - Kueue fails to monitor RayCluster or PyTorchJob resources

When you create a **DataScienceCluster** CR with all components enabled, the Kueue component is installed before the Ray component and the Training Operator component. As a result, the Kueue component does not monitor **RayCluster** or **PyTorchJob** resources.

Workaround

Perform one of the following actions:

- After installing the Ray component and the Training Operator component, restart the Kueue controller pod in the **redhat-ods-applications** namespace.
- Alternatively, edit the **DataScienceCluster** CR to mark the **kueue** component as **Removed**, wait until Kueue is uninstalled, and then mark the **kueue** component as **Managed** again.

RHOAIENG-7716 - Pipeline condition group status does not update

When you run a pipeline that has condition groups, for example, **dsl.If**, the UI displays a **Running** status for the groups, even after the pipeline execution is complete.

Workaround

You can confirm if a pipeline is still running by checking that no child tasks remain active.

1. From the OpenShift AI dashboard, click **Data Science Pipelines → Runs**.
2. From the **Project** drop-down menu, click your data science project.
3. From the **Runs** tab, click the pipeline run that you want to check the status of.
4. Expand the condition group and click a child task.
A panel that contains information about the child task is displayed
5. On the panel, click the **Task** details tab.
The **Status** field displays the correct status for the child task.

RHOAIENG-7346 - Distributed workloads no longer run from existing pipelines after upgrade

After upgrading to OpenShift AI 2.10, a distributed workload no longer runs from an existing pipeline if the cluster is created only inside the pipeline. The pipeline stops working, and the Ray cluster does not start.

Workaround

In the Python code for each existing pipeline for a distributed workload, replace **cluster.wait_ready** with **time.sleep(180)** and recompile the code. Import the pipeline and schedule the pipeline run.

RHOAIENG-7209 - Error displays when setting the default pipeline root

When you set the default pipeline root using the Data Science Pipelines SDK or the OpenShift AI user interface, an error similar to the following appears:

```
F0513 18:25:25.155794 28 main.go:49] failed to execute component: Failed to open bucket "mlpipeline": Failed to retrieve credentials for bucket mlpipeline: Failed to get Bucket credentials from secret name="" namespace="dspa1": resource name may not be empty44
```

Workaround

None.

RHOAIENG-6711 - ODH-model-controller overwrites the `spec.memberSelectors` field in `ServiceMeshMemberRoll` objects

If you add a project or namespace to a `ServiceMeshMemberRoll` resource using the `spec.memberSelectors` field of the `ServiceMeshMemberRoll` resource, the ODH-model-controller overwrites the field.

Workaround

Explicitly add namespaces to the `ServiceMeshMemberRoll` resource using the `spec.members` field as shown in the example:

```
spec:
  members:
  - <namespace 1>
  - <namespace 2>
```

RHOAIENG-6649 - An error is displayed when viewing a model on a model server that has no external route defined

If you use the dashboard to deploy a model on a model server that does not have external routes enabled, a **t.components is undefined** error message might be shown while the model creation is in progress.

Workaround

None.

RHOAIENG-6646 - An error is displayed when viewing the Model Serving page during an upgrade

If you try to use the dashboard to deploy a model while an upgrade of OpenShift AI is in progress, a **t.status is undefined** error message might be shown.

Workaround

Wait until the upgraded OpenShift AI Operator is ready and then refresh the page in your browser.

RHOAIENG-6486 - Pod labels, annotations, and tolerations cannot be configured when using the Elyra JupyterLab extension with the TensorFlow 2024.1 notebook image

When using the Elyra JupyterLab extension with the TensorFlow 2024.1 notebook image, you cannot configure pod labels, annotations, or tolerations from an executed pipeline. This is due to a dependency conflict with the `kfp` and `tf2onnx` packages.

Workaround

If you are working with the TensorFlow 2024.1 notebook image, after you have completed your work, change the assigned workbench notebook image to the Standard Data Science 2024.1 notebook image.

In the **Pipeline properties** tab in the Elyra JupyterLab extension, set the Tensorflow runtime image as the default runtime image for the pipeline node individually, along with the relevant pod label, annotation or toleration, for each pipeline node.

RHOAIENG-6435 - Distributed workloads resources are not included in Project metrics

When you click **Distributed Workloads Metrics > Project metrics** and view the **Requested resources** section, the **Requested by all projects** value currently excludes the resources for distributed workloads that have not yet been admitted to the queue.

Workaround

None.

RHOAIENG-6409 - Cannot save parameter errors appear in pipeline logs for successful runs

When you run a pipeline more than once with Data Science Pipelines 2.0, **Cannot save parameter** errors appear in the pipeline logs for successful pipeline runs. You can safely ignore these errors.

Workaround

None.

RHOAIENG-6376 - Pipeline run creation fails after setting `pip_index_urls` in a pipeline component to a URL that contains a port number and path

When you create a pipeline and set the `pip_index_urls` value for a component to a URL that contains a port number and path, compiling the pipeline code and then creating a pipeline run results in the following error:

```
ValueError: Invalid IPv6 URL
```

Workaround

1. Create a new pip server using only `protocol://hostname`, and update the `pip_index_urls` value for the component with the new server.
2. Recompile the pipeline code.
3. Create a new pipeline run.

RHOAIENG-4812 - Distributed workload metrics exclude GPU metrics

In this release of OpenShift AI, the distributed workload metrics exclude GPU metrics.

Workaround

None.

RHOAIENG-4570 - Existing Argo Workflows installation conflicts with install or upgrade

Data Science Pipelines (DSP) 2.0 contains an installation of Argo Workflows. OpenShift AI does not support direct customer usage of this installation of Argo Workflows. To install or upgrade OpenShift AI with DSP 2.0, ensure that there is no existing installation of Argo Workflows on your cluster. For more information, see [Enabling Data Science Pipelines 2.0](#).

Workaround

Remove the existing Argo Workflows installation or set `datasciencepipelines` to **Removed**, and then proceed with the installation or upgrade.

RHOAIENG-3913 - Red Hat OpenShift AI Operator incorrectly shows **Degraded condition of **False** with an error**

If you have enabled the KServe component in the DataScienceCluster (DSC) object used by the

OpenShift AI Operator, but have not installed the dependent Red Hat OpenShift Service Mesh and Red Hat OpenShift Serverless Operators, the **kserveReady** condition in the DSC object correctly shows that KServe is not ready. However, the **Degraded** condition incorrectly shows a value of **False**.

Workaround

Install the Red Hat OpenShift Serverless and Red Hat OpenShift Service Mesh Operators, and then recreate the DSC.

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-3025 - OVMS expected directory layout conflicts with the KServe StoragePuller layout

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.

Workaround

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.

RHOAIENG-2759 - Model deployment fails when both secured and regular model servers are present in a project

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.

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.

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.

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:

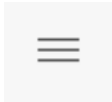
- Deploy the service in another data science project that does not contain models deployed on the single-model serving platform. Or, deploy the service in another OpenShift project.
- In the data science project where the service is, add a [network policy](#) to accept ingress traffic to your application 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.

RHOAIENG-2228 - The performance metrics graph changes constantly when the interval is set to 15 seconds

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.

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.

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-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
)
auth_return = auth.login()
```

**NOTE**

If your cluster uses self-signed certificates, include **ca-cert-path=<path>** in the **TokenAuthentication** parameter list.

RHOAIENG-404 - No Components Found page randomly appears instead of Enabled page in OpenShift AI dashboard

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

Workaround

Refresh the browser page.

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

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

Workaround

Use a different browser.

RHOAIENG-1128 - Unclear error message displays when attempting to increase the size of a Persistent Volume (PV) that is not connected to a workbench

When attempting to increase the size of a Persistent Volume (PV) that is not connected to a workbench, an unclear error message is displayed.

Workaround

Verify that your PV is connected to a workbench before attempting to increase the size.

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.

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.

Workaround

None.

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-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-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-3115 - Model cannot be queried for a few seconds after it is shown as ready

Models deployed using the multi-model serving platform might be unresponsive to queries despite appearing as **Ready** in the dashboard. You might see an "Application is not available" response when querying the model endpoint.

Workaround

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

RHOAIENG-1619 (previously documented as 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.

RHOAIENG-1207 (previously documented as 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.

RHOAIENG-1204 (previously documented as 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.

RHOAIENG-1203 (previously documented as 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.

[RHOAIENG-1201](#) (previously documented as [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.

Workaround

None.

[RHOAIENG-1196](#) (previously documented as [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
```

[RHOAIENG-582](#) (previously documented as [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.

Workaround

None.

[RHOAIENG-432](#) (previously documented as [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.

Workaround

None.

[RHOAIENG-226](#) (previously documented as [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}}}'
```

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.

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.

Workaround

None.

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 a separately installed CodeFlare Operator in your data science cluster](#).

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](#).

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.

Workaround

None.

RHOAIENG-5646 (previously documented as 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.

Workaround

None.

RHOAIENG-1210 (previously documented as 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.

RHOAIENG-1208 (previously documented as **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.

RHOAIENG-1205 (previously documented as **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**.

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.

RHOAIENG-580 (previously documented as **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.

RHOAIENG-583 (previously documented as **RHODS-8921** and **RHODS-6373**) - You cannot create a pipeline server or start a workbench 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. Similarly, when the cumulative character limit of a data science project name and a workbench name exceeds 62 characters, workbenches fail to start.

Workaround

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

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.

RHOAIENG-1157 (previously documented as **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
```

Workaround

None.

RHOAIENG-1132 (previously documented as **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.

RHOAIENG-1152 (previously documented as 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-5763 - Incorrect package version displayed during notebook selection

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

Workaround

None.

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.

RHOAIENG-1149 (previously documented 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.

Workaround

None.

RHOAIENG-1137 (previously documented as 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 whose permissions were revoked.

Workaround

None.

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.

Workaround

None.

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.

Workaround

None.

RHOAIENG-1141 (previously documented as 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.

Workaround

None.

RHOAIENG-1135 (previously documented as 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**.

Workaround

None.

RHOAING-1147 (previously documented as 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.

Workaround

None.

RHOAIENG-1134 (previously documented as 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.

Workaround

None.

RHOAIENG-2305 (previously documented as 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.

Workaround

Contact [Marketplace support](#) for assistance manually configuring Watson Studio on OpenShift Dedicated 4.9 and higher.

CHAPTER 9. 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](#).