



Red Hat OpenShift AI Self-Managed 2.19

Release notes

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

Red Hat OpenShift AI Self-Managed 2.19 Release notes

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

Legal Notice

Copyright © 2025 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 2.19 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	5
CHAPTER 3. TECHNOLOGY PREVIEW FEATURES	7
CHAPTER 4. DEVELOPER PREVIEW FEATURES	10
CHAPTER 5. LIMITED AVAILABILITY FEATURES	11
CHAPTER 6. SUPPORT REMOVALS	12
6.1. DEPRECATED FUNCTIONALITY	12
6.1.1. Deprecated Text Generation Inference Server (TGIS)	12
6.1.2. Deprecated accelerator profiles	12
6.1.3. Deprecated OpenVINO Model Server (OVMS) plugin	12
6.1.4. OpenShift AI dashboard user management moved from ODHDashboardConfig to Auth resource	12
6.1.5. Deprecated cluster configuration parameters	12
6.2. REMOVED FUNCTIONALITY	13
6.2.1. Standalone script for InstructLab removed	13
6.2.2. Anaconda removal	13
6.2.3. Data science pipelines v1 support removed	14
6.2.4. Pipeline logs for Python scripts running in Elyra pipelines are no longer stored in S3	14
6.2.5. Embedded subscription channel no longer used	14
6.2.6. Version 1.2 notebook container images for workbenches are no longer supported	15
6.2.7. Beta subscription channel no longer used	15
6.2.8. HabanaAI workbench image removal	15
6.3. PLANNED SUPPORT REMOVAL	15
6.3.1. Multi-model serving platform (ModelMesh)	15
CHAPTER 7. RESOLVED ISSUES	16
7.1. ISSUES RESOLVED IN RED HAT OPENSIFT AI 2.19	16
CHAPTER 8. KNOWN ISSUES	18
CHAPTER 9. PRODUCT FEATURES	36

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 Intel Gaudi AI accelerators.

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 Intel Gaudi AI accelerators.

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 ROSA HCP), or Microsoft Azure Red Hat OpenShift.
- 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 the [Red Hat OpenShift AI: Supported Configurations](#) Knowledgebase article.

For a detailed view of the 2.19 release lifecycle, including the full support phase window, see the [Red Hat OpenShift AI Self-Managed Life Cycle](#) Knowledgebase article.

CHAPTER 2. NEW FEATURES AND ENHANCEMENTS

This section describes new features and enhancements in Red Hat OpenShift AI 2.19.

2.1. NEW FEATURES

Guardrails Orchestrator Framework

The Guardrails Orchestrator Framework is now generally available.

The Guardrails Orchestrator is a TrustyAI service that adds safety and policy checks (guardrails) to Large Language Models (LLMs). Managed by the TrustyAI Operator, it lets you define rules (detectors) to filter LLM input/output.

Why does it matter?

- LLMs can generate harmful, biased, or inaccurate content. Guardrails Orchestrator mitigates these risks, preventing reputational damage, ethical issues, and legal liabilities.
- It helps ensure your LLM applications are safe, reliable, and policy-compliant. Key benefits include detecting harmful content, enforcing policies, and improving security and quality.

Kubeflow Training Operator (KFTO) for Distributed PyTorch Jobs in OpenShift

This feature enables users to run distributed training jobs using Kubeflow Training Operator (KTFO) using PyTorch jobs, supporting NVIDIA and AMD accelerators.

View installed components and versions

You can now view a list of the installed OpenShift AI components, their corresponding upstream components, and the versions of the installed components. You can access the list of installed components from the **Help** → **About** menu in the Red Hat OpenShift AI dashboard.

OCI containers for model storage

You can use OCI storage as an alternative to cloud storage services for model serving. First, create an OCI container image to contain the model. The image is uploaded to an OCI-compatible registry, such as Quay. When deploying a model, the model serving platform references the repository of the containerized model.

Using an OCI container can provide the following advantages:

- Reduced startup times, because the cluster keeps a cache of downloaded images. Restarting the model pod does not download the model again.
- Lower disk space usage, because the model is not downloaded on each pod replica, assuming pods are scheduled on the same node.
- Enhanced performance when pre-fetching images or asynchronous loading.
- Compatibility and integration, because it can be easily integrated with KServe. No additional dependencies are required and the infrastructure might already be available.

For more information, see [Using OCI containers for model storage](#) .

Multi-node PyTorch distributed training with KFTO

Distributed PyTorch training across multiple nodes and GPUs using the Kubeflow Training Operator is now supported. This feature enables the following functionality:

- Configuration for single or multiple GPUs per node using the PyTorchJob API
- Support for the **kubeflow-training** SDK
- Supports NCCL, RCCL, and GLOO backends for GPU and CPU workloads, with configurable resource allocation
- Training scripts can be mounted using ConfigMaps or included in custom container images.
- Support for both DDP and FSDP distributed training approaches.
- Job scheduling through Distributed Workloads capabilities, or Kueue
- Runtime metrics accessible using OpenShift monitoring

NVIDIA GPUDirect RDMA support for distributed model training

NVIDIA GPUDirect RDMA, which uses Remote Direct Memory Access (RDMA) to provide direct GPU interconnect, is now supported for distributed model training with KFTO. This feature enables NCCL-based collective communication with RDMA over Converged Ethernet (RoCE) and InfiniBand on compatible NVIDIA accelerated networking platforms.

The Kubeflow Training image for CUDA has been updated to include RDMA userspace libraries.

Support for OpenShift AI Self-Managed on Oracle Cloud Infrastructure (OCI)

OpenShift AI Self-Managed is now supported on Red Hat OpenShift Container Platform on Oracle Cloud Infrastructure (OCI). For more information about OpenShift AI supported software platforms, see the [Red Hat OpenShift AI: Supported Configurations](#) Knowledgebase article.

For more information about OpenShift Container Platform on OCI, see [Installing on OCI](#).

2.2. ENHANCEMENTS

Deploying models in standard deployment mode

You can deploy models in either **advanced** or **standard** deployment mode. Standard deployment mode uses KServe RawDeployment mode and does not require the Red Hat OpenShift Serverless Operator, Red Hat OpenShift Service Mesh, or Authorino.

Benefits to standard deployment mode include:

- Enables deployment with Kubernetes resources, such as **Deployment**, **Service**, **Route**, and **Horizontal Pod Autoscaler**. The resulting model deployment has a smaller resource footprint compared to advanced mode.
- Enables traditional Deployment/Pod configurations, such as mounting multiple volumes, which is not available using Knative. This is beneficial for applications requiring complex configurations or multiple storage mounts.

For more information, see [About KServe deployment modes](#).

Updated naming for vLLM serving runtime templates

Template naming has been updated to better distinguish the vLLM templates based on the accelerator supported. All vLLM templates now reflect the accelerator name in the title and description of the template:

- NVIDIA GPU

- AMD GPU
- Gaudi accelerators
- (Technology preview only): CPU (IBM Power and IBM Z)

Support for vLLM Inference Server with Intel Gaudi 1.20

Support for vLLM Inference Server with Intel Gaudi 1.20 accelerators is now available.

Upgraded OpenVINO Model Server

The OpenVINO Model Server has been upgraded to version 2025.0. For information on the changes and enhancements, see [OpenVINO™ Model Server 2025.0](#).

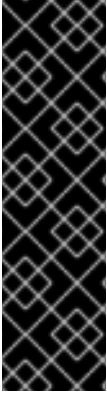
Updated workbench images

A new set of workbench images (2025.1) are now available. This update includes major version upgrades for most pre-built Python packages and updated IDEs for RStudio and Code-server.

Support for Kubeflow Pipelines 2.4.0 in data science pipelines

To keep Red Hat OpenShift AI updated with the latest features, data science pipelines have been upgraded to Kubeflow Pipelines (KFP) version 2.4.0. For more information, see the [Kubeflow Pipelines documentation](#).

CHAPTER 3. TECHNOLOGY PREVIEW FEATURES



IMPORTANT

This section describes Technology Preview features in Red Hat OpenShift AI 2.19. 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](#).

IBM Power and IBM Z architecture support

IBM Power (ppc64le) and IBM Z (s390x) architectures are now supported as a Technology Preview feature. Currently, you can only deploy models in standard mode on these architectures.

Support for vLLM in IBM Power and IBM Z architectures vLLM runtime templates are available for use in IBM Power and IBM Z architectures as Technology Preview.

Enable targeted deployment of workbenches to specific worker nodes in Red Hat OpenShift AI Dashboard using node selectors.

Hardware profiles are now available as a Technology Preview. The hardware profiles feature enables users to target specific worker nodes for workbenches or model-serving workloads. It allows users to target specific accelerator types or CPU-only nodes.

This feature replaces the current accelerator profiles feature and container size selector field, offering a broader set of capabilities for targeting different hardware configurations. While accelerator profiles, taints, and tolerations provide some capabilities for matching workloads to hardware, they do not ensure that workloads land on specific nodes, especially if some nodes lack the appropriate taints.

The hardware profiles feature supports both accelerator and CPU-only configurations, along with node selectors, to enhance targeting capabilities for specific worker nodes. Administrators can configure hardware profiles in the settings menu. Users can select the enabled profiles using the UI for workbenches, model serving, and Data Science Pipelines where applicable.

Distributed InstructLab training

InstructLab is an open-source project for enhancing large language models (LLMs) in generative artificial intelligence (gen AI) applications. It fine-tunes models using synthetic data generation (SDG) techniques and a structured taxonomy to create diverse, high-quality training datasets. The InstructLab pipeline is now available as a Technology Preview feature, enabling you to run the full InstructLab workflow through a data science pipeline in OpenShift AI. For prerequisites and setup instructions to run this pipeline, see [InstructLab on Red Hat OpenShift AI](#).



IMPORTANT

You must have NVIDIA GPU Operator 24.6 installed to use the InstructLab pipeline in OpenShift AI 2.19.

Mandatory Kueue local-queue labeling policy for Ray cluster creation

Cluster administrators can use the Validating Admission Policy feature to enforce the mandatory labeling of Ray cluster resources with Kueue local-queue identifiers. This labeling ensures that workloads are properly categorized and routed based on queue management policies, which prevents resource contention and enhances operational efficiency.

When the local-queue labeling policy is enforced, Ray clusters are created only if they are configured to use a local queue, and the Ray cluster resources are then managed by Kueue. The local-queue labeling policy is enforced for all projects by default, but can be disabled for some or all projects. For more information about the local-queue labeling policy, see [Enforcing the use of local queues](#).



NOTE

This feature might introduce a breaking change for users who did not previously use Kueue local queues to manage their Ray cluster resources.

RStudio Server notebook image

With the RStudio Server notebook image, you can access the RStudio IDE, an integrated development environment for R. The R programming language is used for statistical computing and graphics to support data analysis and predictions.

To use the RStudio Server notebook image, you must first build it by creating a secret and triggering the **BuildConfig**, and then enable it in the OpenShift AI UI by editing the **rstudio-rhel9** image stream. For more information, see [Building the RStudio Server workbench images](#).



IMPORTANT

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

CUDA - RStudio Server notebook image

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

To use the CUDA - RStudio Server notebook image, you must first build it by creating a secret and triggering the **BuildConfig**, and then enable it in the OpenShift AI UI by editing the **rstudio-rhel9** image stream. For more information, see [Building the RStudio Server workbench images](#).



IMPORTANT

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

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

Model Registry

OpenShift AI now supports the Model Registry Operator. The Model Registry Operator is not installed by default in Technology Preview mode. The model registry is a central repository that contains metadata related to machine learning models from inception to deployment.

Support for multinode deployment of very large models

Serving models over multiple graphical processing unit (GPU) nodes when using a single-model serving runtime is now available as a Technology Preview feature. Deploy your models across multiple GPU nodes to improve efficiency when deploying large models such as large language models (LLMs). For more information, see [Deploying models across multiple GPU nodes](#) .

Guardrails Orchestrator Service configurations

The optional Guardrails Orchestrator configurations are now available as a Technology Preview feature:

- Regex detector (Built-in detector)
- Guardrails gateway (through the **vllmGateway** field of the custom resource)

CHAPTER 4. DEVELOPER PREVIEW FEATURES



IMPORTANT

This section describes Developer Preview features in Red Hat OpenShift AI 2.19. 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](#).

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 2.19. 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 the [Red Hat OpenShift AI: Supported Configurations](#) Knowledgebase article.

6.1. DEPRECATED FUNCTIONALITY

6.1.1. Deprecated Text Generation Inference Server (TGIS)

Starting with OpenShift AI version 2.19, the Text Generation Inference Server (TGIS) is deprecated. TGIS will continue to be supported through the OpenShift AI 2.16 EUS lifecycle. Caikit-TGIS and Caikit are not affected and will continue to be supported. The out-of-the-box serving runtime template will no longer be deployed. vLLM is recommended as a replacement runtime for TGIS.

6.1.2. Deprecated accelerator profiles

Accelerator profiles are now deprecated. To target specific worker nodes for workbenches or model serving workloads, use hardware profiles.

6.1.3. Deprecated OpenVINO Model Server (OVMS) plugin

The CUDA plugin for the OpenVINO Model Server (OVMS) is now deprecated and will no longer be available in future releases of OpenShift AI.

6.1.4. OpenShift AI dashboard user management moved from `OdhdashboardConfig` to `Auth` resource

Previously, cluster administrators used the `groupsConfig` option in the `OdhdashboardConfig` resource to manage the OpenShift groups (both administrators and non-administrators) that can access the OpenShift AI dashboard. Starting with OpenShift AI 2.17, this functionality has moved to the `Auth` resource. If you have workflows (such as GitOps workflows) that interact with `OdhdashboardConfig`, you must update them to reference the `Auth` resource instead.

Table 6.1. Updated configurations

Resource	2.16 and earlier	2.17 and later versions
<code>apiVersion</code>	<code>opendatahub.io/v1alpha</code>	<code>services.platform.opendatahub.io/v1alpha1</code>
<code>kind</code>	<code>OdhdashboardConfig</code>	<code>Auth</code>
<code>name</code>	<code>odh-dashboard-config</code>	<code>auth</code>
Admin groups	<code>spec.groupsConfig.adminGroups</code>	<code>spec.adminGroups</code>
User groups	<code>spec.groupsConfig.allowedGroups</code>	<code>spec.allowedGroups</code>

6.1.5. Deprecated cluster configuration parameters

When using the CodeFlare SDK to run distributed workloads in Red Hat OpenShift AI, the following parameters in the Ray cluster configuration are now deprecated and should be replaced with the new parameters as indicated.

Deprecated parameter	Replaced by
<code>head_cpus</code>	<code>head_cpu_requests</code> , <code>head_cpu_limits</code>
<code>head_memory</code>	<code>head_memory_requests</code> , <code>head_memory_limits</code>
<code>min_cpus</code>	<code>worker_cpu_requests</code>
<code>max_cpus</code>	<code>worker_cpu_limits</code>
<code>min_memory</code>	<code>worker_memory_requests</code>
<code>max_memory</code>	<code>worker_memory_limits</code>
<code>head_gpus</code>	<code>head_extended_resource_requests</code>
<code>num_gpus</code>	<code>worker_extended_resource_requests</code>

You can also use the new `extended_resource_mapping` and `overwrite_default_resource_mapping` parameters, as appropriate. For more information about these new parameters, see the [CodeFlare SDK documentation](#) (external).

6.2. REMOVED FUNCTIONALITY

6.2.1. Standalone script for InstructLab removed

The standalone script for running Distributed InstructLab training has been removed. To run the InstructLab training flow, use the InstructLab pipeline. For more information, see [InstructLab on Red Hat OpenShift AI](#).

6.2.2. Anaconda removal

Anaconda is an open source distribution of the Python and R programming languages. Starting with OpenShift AI version 2.18, Anaconda is no longer included in OpenShift AI, and Anaconda resources are no longer supported or managed by OpenShift AI.

If you previously installed Anaconda from OpenShift AI, a cluster administrator must complete the following steps from the OpenShift command-line interface to remove the Anaconda-related artifacts:

1. Remove the secret that contains your Anaconda password:
oc delete secret -n redhat-ods-applications anaconda-ce-access
2. Remove the **ConfigMap** for the Anaconda validation cronjob:
oc delete configmap -n redhat-ods-applications anaconda-ce-validation-result

3. Remove the Anaconda image stream:
oc delete imagestream -n redhat-ods-applications s2i-minimal-notebook-anaconda
4. Remove the Anaconda job that validated the downloading of images:
oc delete job -n redhat-ods-applications anaconda-ce-periodic-validator-job-custom-run
5. Remove any pods related to Anaconda cronjob runs:
oc get pods -n redhat-ods-applications --no-headers=true | awk '/anaconda-ce-periodic-validator-job-custom-run*/'

6.2.3. Data science pipelines v1 support removed

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 2.0 is enabled and deployed by default in OpenShift AI.

Starting with OpenShift AI 2.16, data science pipelines 1.0 resources are no longer supported or managed by OpenShift AI. It is no longer possible to deploy, view, or edit the details of pipelines that are based on data science pipelines 1.0 from either the dashboard or the KFP API server.

OpenShift AI does not automatically migrate existing data science pipelines 1.0 instances to 2.0. If you are upgrading to OpenShift AI 2.16 or later, you must manually migrate your existing data science pipelines 1.0 instances. For more information, see [Migrating to data science pipelines 2.0](#).



IMPORTANT

Data science pipelines 2.0 contains an installation of Argo Workflows. Red Hat does not support direct customer usage of this installation of Argo Workflows. To install or upgrade to OpenShift AI 2.16 or later with data science pipelines 2.0, ensure that there is no existing installation of Argo Workflows on your cluster.

6.2.4. Pipeline logs for Python scripts running in Elyra pipelines are no longer stored in S3

Logs are no longer stored in S3-compatible storage for Python scripts which are running in Elyra pipelines. From OpenShift AI version 2.11, you can view these logs in the pipeline log viewer in the OpenShift AI dashboard.



NOTE

For this change to take effect, you must use the Elyra runtime images provided in workbench images at version 2024.1 or later.

If you have an older workbench image version, update the **Version selection** field to a compatible workbench image version, for example, 2024.1, as described in [Updating a project workbench](#).

Updating your workbench image version will clear any existing runtime image selections for your pipeline. After you have updated your workbench version, open your workbench IDE and update the properties of your pipeline to select a runtime image.

6.2.5. 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](#).

6.2.6. 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.

6.2.7. 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](#).

6.2.8. HabanaAI workbench image removal

Support for the HabanaAI 1.10 workbench image has been removed. New installations of OpenShift AI from version 2.14 do not include the HabanaAI workbench image. However, if you upgrade OpenShift AI from a previous version, the HabanaAI workbench image remains available, and existing HabanaAI workbench images continue to function.

6.3. PLANNED SUPPORT REMOVAL

6.3.1. Multi-model serving platform (ModelMesh)

The multi-model serving platform based on ModelMesh is now deprecated. ModelMesh is planned to be supported until at least April 2026. You can continue to deploy models on the multi-model serving platform, but it is recommended that you migrate to the single-model serving platform.

For more information or for help on using the single-model serving platform, contact your account manager.

CHAPTER 7. RESOLVED ISSUES

The following notable issues are resolved in Red Hat OpenShift AI 2.19. Security updates, bug fixes, and enhancements for Red Hat OpenShift AI 2.19 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 2.19

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

Previously, using TensorFlow-based workbench images did not allow users to use pod labels, annotations, and tolerations when using the Elyra JupyterLab extension. With the 2025.1 images, the TensorFlow-based workbench is upgraded with the KubeFlow pipeline SDK (kfp). With the upgraded SDK, you can set pod labels, annotations, and tolerations when using the Elyra extension to schedule the Data Science pipelines.

RHOAIENG-21197 - Deployment failure when using vLLM runtime on AMD GPU accelerators in a FIPS-enabled cluster

Previously, when deploying a model by using the vLLM runtime on AMD GPU accelerators in a FIPS-enabled cluster, the deployment could fail. This issue is now resolved.

RHOAIENG-20245 - Certain model registry operations remove custom properties from the registered model and version

Previously, editing the description, labels, or properties of a model version removed labels and custom properties from the associated model. Deploying a model version, or editing its model source format, removed labels and custom properties from the version and from the associated model. This issue is now resolved.

RHOAIENG-19954 - Kueue alerts not monitored in OpenShift

Previously, in the OpenShift console, Kueue alerts were not monitored. The new **ServiceMonitor** resource rejected the usage of the **BearerTokenFile** field, which meant that Prometheus did not have the required permissions to scrape the target. As a result, the Kueue alerts were not shown on the **Observe → Alerting** page, and the Kueue targets were not shown on the **Observe → Targets** page. This issue is now resolved.

RHOAIENG-19716 - The **system-authenticated** user group cannot be removed by using the dashboard

Previously, after installing or upgrading Red Hat OpenShift AI, the **system-authenticated** user group displayed in **Settings > User management** under **Data science user groups**. If you removed this user group from **Data science user groups** and saved the changes, the group was erroneously added again. This issue is now resolved.

RHOAIENG-18238 - Inference endpoints for deployed models return 403 error after upgrading the Authorino Operator

Previously, after upgrading the Authorino Operator, the automatic Istio sidecar injection may not have been reapplied. Without the sidecar, Authorino was not correctly integrated into the service mesh, and caused inference endpoint requests to fail with an HTTP 403 error. This issue is now resolved.

RHOAIENG-11371 - Incorrect run status reported for runs using ExitHandler

Previously, when using pipeline exit handlers (**`dsl.ExitHandler`**), if a task inside the handle failed but the exit task succeeded, the overall pipeline run status was inaccurately reported as **Succeeded** instead of **Failed**. This issue is now resolved.

RHOAIENG-16146 - Connection sometimes not preselected when deploying a model from model registry

Previously, when deploying a model from a model registry, the object storage *connection* (previously called *data connection*) might not have been preselected. This issue is now resolved.

RHOAIENG-21068 - InstructLab pipeline run cannot be created when the parameter **`sdg_repo_pr`** is left empty

Previously, when creating a pipeline run of the InstructLab pipeline, if the parameter **`sdg_repo_pr`** was left empty, the pipeline run could not be created and an error message appeared. This issue is now resolved.

CHAPTER 8. KNOWN ISSUES

This section describes known issues in Red Hat OpenShift AI 2.19 and any known methods of working around these issues.

[RHOAIENG-24886](#) - Cannot deploy OCI model when Model URI field includes prefix

When deploying an OCI model, if you paste the complete URI in the **Model URI** field and then move the cursor to another field, the URL prefix (for example, **http://**) is removed from the **Model URI** field, but it is included in the **storageUri** value in the **InferenceService** resource. As a result, you cannot deploy the OCI model.

Workaround

To *prevent* the issue, manually remove the URI prefix from the **Model URI** field before you move the cursor from that field, or copy and paste the URI text excluding the prefix.

To *recover from* the issue, open the **InferenceService** resource and edit the **storageUri** value to change the prefix from **http** to **oci**.

[RHOAIENG-24104](#) - KServe reconciler should only deploy certain resources when Authorino is installed

When Authorino is not installed, Red Hat OpenShift AI applies the **AuthorizationPolicy** and **EnvoyFilter** resources to the KServe serverless deployment mode. This could block some inference requests.

Workaround

Install Authorino, then restart the OpenShift AI operator, KServe controller, and **odh-model-controller** by deleting the existing pods.

[RHOAIENG-23596](#) - Inference requests on IBM Power with longer prompts to the inference service fail with a timeout error

When using the IBM Power architecture to send longer prompts of more than 100 input tokens to the inference service, there is no response from the inference service. An error message similar to the following appears:

504 Gateway Time-out - The server didn't respond in time.

Workaround

There are two options for working around this issue:

- When creating an inference service, set the environment variable **OMP_NUM_THREADS** to **16**.
- Use more CPUs.

[RHOAIENG-23562](#) - TrustyAIService TLS handshake error in FIPS clusters

When using a FIPS cluster that uses an external route to send a request to the TrustyAIService, a TLS handshake error appears in the logs and the request is not processed.

Workaround

Use the UI to send a request to the TrustyAIService.

RHOAIENG-23475 - Inference requests on IBM Power in a disconnected environment fail with a timeout error

When using the IBM Power architecture, if prompts sent to the inference service are created in a disconnected environment, there is no response from the inference service. An error message similar to the following appears:

504 Gateway Time-out - The server didn't respond in time.

Workaround

There are two options for working around this issue:

- When creating an inference service, set the environment variable **OMP_NUM_THREADS** to **16**.
- Use more CPUs.

RHOAIENG-23169 - StorageInitializer fails to download models from Hugging Face repository

Deploying models from Hugging Face in a KServe environment using the **hf://** protocol fails when the cluster lacks built-in support for this protocol. Additionally, the storage initializer **InitContainer** in KServe can encounter a **PermissionError** because of insufficient write permissions in the default cache directory (**/.cache**).

Workaround

Install a **ClusterStorageContainer** resource in the KServe cluster with the following values:

- Enable support for the **hf://** URI format in the **supportedUriFormats** section.
- Set the **HF_HOME** environment variable in the **storage-initializer** container to a writable directory, such as **/tmp/hf_home**.

The following code sample shows a **ClusterStorageContainer** resource with these values set:

```
apiVersion: "serving.kserve.io/v1alpha1"
kind: ClusterStorageContainer
metadata:
  name: default
spec:
  container:
    name: storage-initializer
    env:
      - name: HF_HOME
        value: /tmp/hf_home
  supportedUriFormats:
    - prefix: hf://
    - prefix: gs://
    - prefix: s3://
    - prefix: hdfs://
    - prefix: webhdfs://
    - regex: "https://(.+?).blob.core.windows.net/(.+)"
    - regex: "https://(.+?).file.core.windows.net/(.+)"
    - regex: "https?://(.+)/(.+)"
```

RHOAIENG-22965 - Data science pipeline task fails when optional input parameters are not set

When a pipeline has optional input parameters, creating a pipeline run and tasks with unset parameters fails with the following error:

```
failed: failed to resolve inputs: resolving input parameter optional_input with spec
component_input_parameter:"parameter_name": parent DAG does not have input parameter
```

This issue also affects the InstructLab pipeline.

Workaround

Provide values for all optional parameters.

RHOAIENG-22439 - cuda-rstudio-rhel9 cannot be built

When building the RStudio Server workbench images, building the **cuda-rstudio-rhel9** fails with the following error:

```
Package supervisor-4.2.5-6.el9.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
.M...U... /run/supervisor
error: build error: building at STEP "RUN yum -y module enable nginx:$NGINX_VERSION &&
INSTALL_PKGS="nss_wrapper bind-utils gettext hostname nginx nginx-mod-stream nginx-mod-http-
perl fcgiwrap initscripts chkconfig supervisor" && yum install -y --setopt=tsflags=nodocs
$INSTALL_PKGS && rpm -V $INSTALL_PKGS && nginx -v 2>&1 | grep -qe
"nginx/$NGINX_VERSION\." && echo "Found VERSION $NGINX_VERSION" && yum -y clean all
--enablerepo='*': while running runtime: exit status 1
```

Workaround

Run the following command to re-tag **cuda-rstudio-rhel9**:

```
oc tag --source=imagestreamtag redhat-ods-applications/cuda-rhel9:latest redhat-ods-
applications/cuda-rstudio-rhel9:latest
```

Output similar to the following appears:

```
Tag redhat-ods-applications/cuda-rstudio-rhel9:latest set to redhat-ods-applications/cuda-
rhel9@sha256:c03a3017364f311ad41f3a3677e0a532020cdd9f57fd98578288eb789cffbf4f.
```

RHOAIENG-21274 - Connection type changes back to S3 or URI when deploying a model with an OCI connection type

If you deploy a model that is using the S3 or URI connection type in a project with no matching connections, the **Create new connection** section is pre-populated using data from the model location of the S3 or URI storage location. If you change the connection type to OCI and enter a value in the **Model URI** field, the connection type changes back to S3 or URI.

Workaround

To deploy a model with OCI connection type, either register a model with OCI connections or use a model that has an OCI connection. Do not change the connection type while deploying from the model registry.

RHOAIENG-20595 - Pipeline fails to run if the `http_proxy` or `https_proxy` environment variable is set

If you set the `http_proxy` or `https_proxy` environment variable in a pipeline task, the pipeline fails with the following error:

```
Connecting to cache endpoint ds-pipeline-dspa.project-name.svc.cluster.local:8887 panic: runtime
error: invalid memory address or nil pointer dereference
```

Workaround

Set the `no_proxy` environment variable to the following value and the pipeline will run as expected:

```
my-task.set_env_variable("no_proxy",
"localhost,127.0.0.1,svc.cluster.local,0,1,2,3,4,5,6,7,8,9")
```

RHOAIENG-20209 - Warning message not displayed when requested resources exceed threshold

When you click **Distributed workloads** → **Project metrics** and view the **Requested resources** section, the charts show the requested resource values and the total shared quota value for each resource (**CPU** and **Memory**). However, when the **Requested by all projects** value exceeds the **Warning threshold** value for that resource, the expected warning message is not displayed.

Workaround

None.

RHOAIENG-18590 - The **KnativeServing** resource fails after disabling and enabling KServe

After disabling and enabling the **kserve** component in the DataScienceCluster, the **KnativeServing** resource might fail.

Workaround

Delete all **ValidatingWebhookConfiguration** and **MutatingWebhookConfiguration** webhooks related to Knative:

1. Get the webhooks:

```
oc get ValidatingWebhookConfiguration,MutatingWebhookConfiguration | grep -i knative
```

2. Ensure KServe is disabled.

3. Get the webhooks:

```
oc get ValidatingWebhookConfiguration,MutatingWebhookConfiguration | grep -i knative
```

4. Delete the webhooks.

5. Enable KServe.

6. Verify that the KServe pod can successfully spawn, and that pods in the **knative-serving** namespace are active and operational.

RHOAIENG-16247 - Elyra pipeline run outputs are overwritten when runs are launched from OpenShift AI dashboard

When a pipeline is created and run from Elyra, outputs generated by the pipeline run are stored in the folder **bucket-name/pipeline-name-timestamp** of object storage.

When a pipeline is created from Elyra and the pipeline run is started from the OpenShift AI dashboard, the timestamp value is not updated. This can cause pipeline runs to overwrite files created by previous pipeline runs of the same pipeline.

This issue does not affect pipelines compiled and imported using the OpenShift AI dashboard because **runid** is always added to the folder used in object storage. For more information about storage locations used in data science pipelines, see [Storing data with data science pipelines](#).

Workaround

When storing files in an Elyra pipeline, use different subfolder names on each pipeline run.

[OCPBUGS-49422](#) - AMD GPUs and AMD ROCm workbench images are not supported in a disconnected environment

This release of OpenShift AI does not support AMD GPUs and AMD ROCm workbench images in a disconnected environment because installing the AMD GPU Operator requires internet access to fetch dependencies needed to compile GPU drivers.

Workaround

None.

[RHOAIENG-15982](#) - Replicas not created correctly in multinode deployment when **pipelineParallelSize** parameter updated

When you update the **pipelineParallelSize** parameter in a multinode deployment, new replicas are not created correctly. Instead, the existing ReplicaSet's replicas are modified, causing the new deployment to malfunction.

Workaround

Remove the existing InferenceService CR and create an InferenceService CR with the updated **pipelineParallelSize** value.

[RHOAIENG-14271](#) - Compatibility errors occur when using different Python versions in Ray clusters with Jupyter notebooks

When you use Python version 3.11 in a Jupyter notebook, and then create a Ray cluster, the cluster defaults to a workbench image that contains Ray version 2.35 and Python version 3.9. This mismatch can cause compatibility errors, as the Ray Python client requires a Python version that matches your workbench configuration.

Workaround

Use a workbench image with your Ray cluster that contains a matching Python version with the **ClusterConfiguration** argument.

[RHOAIENG-12516](#) - **fast** releases are available in unintended release channels

Due to a known issue with the stream image delivery process, **fast** releases are currently available on unintended streaming channels, for example, **stable**, and **stable-x.y**. For accurate release type, channel, and support lifecycle information, refer to the **Life-cycle Dates** table on the [Red Hat OpenShift AI Self-Managed Life Cycle](#) page.

Workaround

None.

RHOAIENG-12233 - The `chat_template` parameter is required when using the `/v1/chat/completions` endpoint

When configuring the **vLLM ServingRuntime for KServer** runtime and querying a model using the `/v1/chat/completions` endpoint, the model fails with the following **400 Bad Request** error:

As of transformers v4.44, default chat template is no longer allowed, so you must provide a chat template if the tokenizer does not define one

Transformers v4.44.2 is shipped with vLLM v0.5.5. As of vLLM v0.5.5, you must provide a chat template while querying a model using the `/v1/chat/completions` endpoint.

Workaround

If your model does not include a predefined chat template, you can use the **chat-template** command-line parameter to specify a chat template in your custom vLLM runtime, as shown in the example. Replace `<CHAT_TEMPLATE>` with the path to your template.

```
containers:
  - args:
    - --chat-template=<CHAT_TEMPLATE>
```

You can use the chat templates that are available as **.jinja** files in <https://github.com/opendatahub-io/vllm/tree/main/examples> or with the vLLM image under `/apps/data/template`. For more information, see [Chat templates](#).

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

If you try to upgrade OpenShift AI 2.8 to version 2.10 or later, 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. Wait for about 20 seconds, and then ensure that a new **AppWrapper** CRD is automatically applied, as shown in the following example:

```
$ oc get crd appwrappers.workload.codeflare.dev
NAME                                CREATED AT
appwrappers.workload.codeflare.dev  2024-11-22T18:35:04Z
```

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, restart the **odh-model-controller** deployment object.

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** list, 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-6435 - Distributed workloads resources are not included in Project metrics

When you click **Distributed workloads > 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-12294 (previously documented as 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 2.0 contains an installation of Argo Workflows. Red Hat does not support direct customer usage of this installation of Argo Workflows. To install or upgrade OpenShift AI with data science pipelines 2.0, ensure that there is no existing installation of Argo Workflows on your cluster. For more information, see [Migrating to 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-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-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-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-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-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-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-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.

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-16568 (previously documented as **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.

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.

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-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-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.

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.

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](#).