Red Hat OpenShift AI Self-Managed 2.21

Release notes

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

Last Updated: 2025-06-18

Red Hat OpenShift AI Self-Managed 2.21 Release notes

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

Legal Notice

Abstract

These release notes provide an overview of new features, enhancements, resolved issues, and known issues in version 2.21 of Red Hat OpenShift AI.

Table of Contents

CHAPTER 1. OVERVIEW OF OPENSHIFT AI

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

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

For data scientists, OpenShift AI includes Jupyter and a collection of default workbench 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 workbenches 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 2.21 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 2.21.

2.1. NEW FEATURES

Red Hat OpenShift AI designed for FIPS

OpenShift AI is now designed for FIPS. All components provided by Red Hat use the Red Hat Enterprise Linux (RHEL) cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 validation. Additionally, all base container images have been updated to use UBI 9.

For more information about the NIST validation program, see Cryptographic Module Validation Program. For the latest NIST status for the individual versions of RHEL cryptographic libraries that have been submitted for validation, see Red Hat Product compliance .

IMPORTANT

When installing or upgrading to OpenShift AI 2.21 on a cluster running in FIPS mode, any custom container images used in data science pipelines must also be based on UBI 9 or RHEL 9. This ensures compatibility with the FIPS-approved data science pipelines components and prevents errors related to mismatched OpenSSL or GNU C Library (glibc) versions.

View project-specific workbench images, serving runtimes, and hardware profiles

This feature introduces a project-level view of OpenShift AI items, including workbench images, serving runtimes, and hardware profiles. When selecting these resources in the dashboard, you can choose from project-scoped and global options, based on what is available in your environment. These options let you customize your dashboard to show and use only the items relevant to your project, without impacting others.

Cluster administrators make project-specific items available by creating the appropriate resources in the target project namespace. This feature improves dashboard functionality by enhancing customization and user self-sufficiency.

Hugging Face Detector runtime available for the**odh-model-controller** deployment object

The Hugging Face Detector runtime is now available from the **odh-model-controller** deployment object. You can now deploy the Hugging Face sentence and token classifier models such as IBM’s Granite-Guardian-HAP model as guardrail detectors.

2.2. ENHANCEMENTS

Updated terminology from notebooks to workbenches

The OpenShift AI terminology has been updated to use the term "workbenches" instead of "notebooks" in many areas of the product. This change better reflects the broader scope of capabilities available. The term "notebooks" now specifically refers to Jupyter notebooks. While Jupyter notebooks are a type of workbench, workbenches also include other tools, such as code-server (VS Code) images, that are not notebooks. This is a terminology change only—the underlying functionality remains unchanged.

Distributed workloads: additional training image tested and verified

An additional image is tested and verified:

- **CUDA-compatible Ray cluster image**
  A new Ray-based training image, **quay.io/modh/ray:2.44.1-py311-cu121** is tested and verified. This image is compatible with AMD accelerators that are supported by CUDA 12.1.

- **ROCm-compatible Ray cluster image**
  The ROCm-compatible Ray cluster image **quay.io/modh/ray:2.44.1-py311-rocm62** is tested and verified. This image is compatible with AMD accelerators that are supported by ROCm 6.2.

NOTE

These images are AMD64 images, which might not work on other architectures. For more information about the latest available training images in Red Hat OpenShift AI, see Red Hat OpenShift AI Supported Configurations .

Simplified model deployment from pipelines using KServe

You can now deploy models as inference services directly from pipeline steps using the KServe SDK. The default service account for data science pipelines has been updated with the required RBAC permissions to create **InferenceService** objects, which removes the need to manually update roles or create custom service accounts.

Ability to use the latest pipeline version in scheduled runs

Now, you can configure a scheduled run to automatically use the latest version so that each recurring run always uses the most recent pipeline version without requiring manual updates after a pipeline change or upgrade.

Updated vLLM component versions

OpenShift AI 2.21 supports the following vLLM versions for each listed component:

- vLLM CUDA v0.9.0.1 (designed for FIPS)
- vLLM ROCm v0.8.4.3 (designed for FIPS)
- vLLM Power v0.8.5
- vLLM Z v0.8.5 (designed for FIPS)
- vLLM Gaudi v0.6.6.post1

For more information, see **vllm** in GitHub.

CHAPTER 3. TECHNOLOGY PREVIEW FEATURES

IMPORTANT

This section describes Technology Preview features in Red Hat OpenShift AI 2.21. 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 .

Model customization with LAB-tuning

LAB-tuning is now available as a Technology Preview feature, enabling data scientists to run an end-to-end workflow for customizing large language models (LLMs). The LAB (Large-scale Alignment for chatBots) method offers a more efficient alternative to traditional fine-tuning by leveraging taxonomy-guided synthetic data generation (SDG) and a multi-phase training approach.

Data scientists can run LAB-tuning workflows directly from the OpenShift AI dashboard by using the new preconfigured InstructLab pipeline, which simplifies the tuning process. For details on enabling and using LAB-tuning, see Enabling LAB-tuning and Customizing models with LAB-tuning.

IMPORTANT

The LAB-tuning feature is not currently supported for disconnected environments.

Red Hat OpenShift AI Model Catalog

The Red Hat OpenShift AI Model Catalog is now available as a Technology Preview feature. This functionality starts with connecting users with the Granite family of models, as well as the teacher and judge models used in LAB-tuning.

NOTE

The model catalog feature is not currently supported for disconnected environments.

New Feature Store component

You can now install and manage Feature Store as a configurable component in the Red Hat OpenShift AI Operator. Based on the open-source Feast project, Feature Store acts as a bridge between ML models and data, enabling consistent and scalable feature management across the ML lifecycle.

This Technology Preview release introduces the following capabilities:

- Centralized feature repository for consistent feature reuse
- Python SDK and CLI for programmatic and command-line interactions to define, manage, and retrieve features for ML models
- Feature definition and management

- Support for a wide range of data sources
- Data ingestion via feature materialization
- Feature retrieval for both online model inference and offline model training
- Role-Based Access Control (RBAC) to protect sensitive features
- Extensibility and integration with third-party data and compute providers
- Scalability to meet enterprise ML needs
- Searchable feature catalog
- Data lineage tracking for enhanced observability
  For configuration details, see Configuring Feature Store.

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.

Mandatory Kueue local-queue labeling policy for Ray cluster and PyTorchJob creation

Cluster administrators can use the Validating Admission Policy feature to enforce the mandatory labeling of Ray cluster and PyTorchJob 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 and PyTorchJobs are created only if they are configured to use a local queue, and the Ray cluster and PyTorchJob 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 and PyTorchJob resources.

RStudio Server workbench image

With the RStudio Server workbench 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 workbench 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 workbench image

With the CUDA - RStudio Server workbench 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 workbench 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 workbench 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 2.21. 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.

LLM Compressor integration

LLM Compressor capabilities are now available in Red Hat OpenShift AI as a Developer Preview feature. A new workbench image with the **llm-compressor** library and a corresponding data science pipelines runtime image make it easier to compress and optimize your large language models (LLMs) for efficient deployment with vLLM. For more information, see **llm-compressor** in GitHub.

You can use LLM Compressor capabilities in two ways:

- Use a Jupyter notebook with the workbench image available at Red Hat Quay.io: **opendatahub / llmcompressor-workbench**.
  For an example Jupyter notebook, see **examples/llmcompressor/workbench_example.ipynb** in the **red-hat-ai-examples** repository.

- Run a data science pipeline that executes model compression as a batch process with the runtime image available at Red Hat Quay.io: **opendatahub / llmcompressor-pipeline-runtime**.
  For an example pipeline, see **examples/llmcompressor/oneshot_pipeline.py** in the **red-hat-ai-examples** repository.

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 2.21. 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.

Embedded subscription channel

In OpenShift AI 2.21, use of the **embedded** subscription channel is a Limited Availability feature. For more information about subscription channels, see Installing the Red Hat OpenShift AI Operator .

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
apiVersion	**opendatahub.io/v1alpha**	**services.platform.opendatahub.io/v1alpha1**
kind	**OdhDashboardConfig**	**Auth**
name	**odh-dashboard-config**	**auth**
Admin groups	**spec.groupsConfig.adminGroups**	**spec.adminGroups**
User groups	**spec.groupsConfig.allowedGroups**	**spec.allowedGroups**

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
head_cpus	**head_cpu_requests**, **head_cpu_limits**
head_memory	**head_memory_requests**, **head_memory_limits**
min_cpus	**worker_cpu_requests**
max_cpus	**worker_cpu_limits**
min_memory	**worker_memory_requests**
max_memory	**worker_memory_limits**
head_gpus	**head_extended_resource_requests**
num_gpus	**worker_extended_resource_requests**

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. Embedded subscription channel not used in some versions

For OpenShift AI 2.8 to 2.20 inclusive, the **embedded** subscription channel is not used. You cannot select the **embedded** channel for a new installation of the Operator for those versions. For more information about subscription channels, see Installing the Red Hat OpenShift AI Operator .

6.2.2. Standalone script for InstructLab removed

The standalone script for running Distributed InstructLab training has been removed. To run the InstructLab training flow, use the LAB-tuning Technology Preview feature. For more information, see Enabling LAB-tuning and Customizing models with LAB-tuning.

IMPORTANT

The LAB-tuning feature is currently not supported for disconnected environments.

6.2.3. 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*/'
   ```

6.2.4. 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.

6.2.5. 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.

6.2.6. Version 1.2 container images for workbenches are no longer supported

When you create a workbench, you specify a container image to use with the workbench. Starting with OpenShift AI 2.5, when you create a new workbench, version 1.2 container images are not available to select. Workbenches that are already running with a version 1.2 image continue to work normally. However, Red Hat recommends that you update your workbench to use the latest 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 2.21. Security updates, bug fixes, and enhancements for Red Hat OpenShift AI 2.21 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 OPENSHIFT AI 2.21

RHOAIENG-24682 - [vLLM-Cuda] Unable to deploy model on FIPS enabled cluster

Previously, if you deployed a model by using the **vLLM NVIDIA GPU ServingRuntime for KServe**or **vLLM ServingRuntime Multi-Node for KServe**runtimes on NVIDIA accelerators in a FIPS-enabled cluster, the deployment could fail. This issue is now resolved.

RHOAIENG-23596 - Inference requests on IBM Power with longer prompts to the inference service fail with a timeout error

Previously, when using the IBM Power architecture to send longer prompts of more than 100 input tokens to the inference service, there was no response from the inference service. This issue no longer occurs.

CHAPTER 8. KNOWN ISSUES

This section describes known issues in Red Hat OpenShift AI 2.21 and any known methods of working around these issues.

RHOAIENG-27676 - Accelerator profile does not work correctly with deleted case

If you delete your Accerator profile after you created a workbench, deployment, or model server, the **Edit** page does not use existing settings and shows the wrong Accelerator profile.

Workaround

None.

RHOAIENG-26537 - Users cannot access the dashboard after installing OpenShift AI 2.21

After you install OpenShift AI 2.21 and create a **DataScienceCluster** on a new cluster, users cannot access the dashboard due to the **Auth** custom resource being created without the default group configuration.

Workaround

Manually add the default group configuration to the **Auth** resource in the **services.platform.opendatahub.io** API group:

1. Log in to the OpenShift console as a cluster administrator.
2. In the web console, click **Operators** → **Installed Operators** and then click the Red Hat OpenShift AI Operator.
3. Click the **All instances** tab.
4. Click the **auth** resource, and then click the **YAML** tab.
5. Edit the YAML file to add the following information:

```
spec:
  adminGroups:
 - rhods-admins
  allowedGroups:
 - 'system:authenticated'
```

6. Click **Save**.

RHOAIENG-26464 - InstructLab training phase 1 pods fail due to insufficient memory when using the default value

When you run the InstructLab pipeline using the default value for the **train_memory_per_worker** input parameter (100 GiB), the phase1 training task fails because of insufficient pod memory.

Workaround

In this release, the base image used for the training phase has been updated to Red Hat Enterprise Linux AI 1.5, which generates better output models but requires more memory during the training phase. Set **train_memory_per_worker** to 250 GiB or higher for a successful pipeline run.

RHOAIENG-26263 - Node selector not cleared when changing the hardware profile for a workbench or model deployment

If you edit an existing workbench or model deployment to change the hardware profile from one that includes a node selector to one that does not, the previous node placement settings might not be removed. As a result, your workload is still scheduled based on the old node selector, leading to an inefficient use of resources.

Workaround

Manually remove the node selector setting:

1. Log in to the OpenShift console as a cluster administrator.
2. Click Home → Search.
3. Select your project from the Project list.
4. In the Resources list, select Notebook for a workbench or InferenceService for a model deployment.
5. Click the workbench or model deployment to open the details page.
6. Click the YAML tab and set the nodeSelector field to **{}** as shown in the following examples:

   Notebook details (workbench) YAML

   ```
   spec:
     template:
    spec:
      nodeSelector: {}
   ```

   InferenceService details (model deployment) YAML

   ```
   spec:
    predictor:
      nodeSelector: {}
   ```

7. Click **Save**.

RHOAIENG-25734 - Duplicate name issue with notebook images

When you delete a workbench after you have created a workbench, deployment, or model server and use the same name for both the product-scoped and global-scoped Imagestreams, the workbench displays an incorrect name in the workbench table and in the **Edit workbench** form.

Workaround

Do not use the same name for your project-scoped and global-scoped Accelerator profiles.

RHOAIENG-25733 - Accelerator profile does not work correctly with duplicate name

When you create a workbench, deployment, or model and use the same name for the project-scoped Accelerator profile as the global-scoped Accelerator profile, the **Edit** page and server form display incorrect labels in the respective tables and form.

Workaround

Do not use the same name for your project-scoped and global-scoped Accelerator profiles.

RHOAIENG-24545 - Runtime images are not present in the workbench after the first start

The list of runtime images does not properly populate the first running workbench instance in the namespace, therefore no image is shown for selection in the Elyra pipeline editor.

Workaround

Restart the workbench. After restarting the workbench, the list of runtime images populates both the workbench and the select box for the Elyra pipeline editor.

RHOAIENG-25090 - InstructLab **prerequisites-check-op** task fails when the model registration option is disabled

When you start a LAB-tuning run without selecting the **Add model to <model registry name>**checkbox, the InstructLab pipeline starts, but the **prerequisites-check-op** task fails with the following error in the pod logs:

```
failed: failed to resolve inputs: the resolved input parameter is null: output_model_name
```

Workaround

Select the **Add model to <model registry name>**checkbox when you configure the LAB-tuning run.

RHOAIENG-25056 - Data science pipeline task fails when optional input parameters used in nested pipelines are not set

When a pipeline has optional input parameters, if values for those parameters are not provided and they are used in a nested pipeline, the tasks using them fail with the following error:

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

Workaround

Provide values for all optional parameters when using nested pipeline tasks.

RHOAIENG-24786 - Upgrading the Authorino Operator from Technical Preview to Stable fails in disconnected environments

In disconnected environments, upgrading the Red Hat Authorino Operator from Technical Preview to Stable fails with an error in the **authconfig-migrator-qqttz** pod.

Workaround

1. Update the Red Hat Authorino Operator to the latest version in the **tech-preview-v1** update channel (v1.1.2).

2. Run the following script:

```
#!/bin/bash
set -xe

# delete the migrator job
oc delete job -n openshift-operators authconfig-migrator || true

# run the migrator script
authconfigs=$(oc get authconfigs -A -o custom-
columns='NAMESPACE:.metadata.namespace,NAME:.metadata.name' --no-headers)

if [[ ! -z "${authconfigs}" ]]; then
  while IFS=" " read -r namespace name; do
    oc get authconfig "$name" -n "$namespace" -o yaml >
"/tmp/${name}.${namespace}.authconfig.yaml"
    oc apply -f "/tmp/${name}.${namespace}.authconfig.yaml"
  done <<< "${authconfigs}"
fi

oc patch crds authconfigs.authorino.kuadrant.io --type=merge --subresource status --
patch '{"status":{"storedVersions":["v1beta2"]}}'
```

3. Update the Red Hat Authorino Operator subscription to use the **stable** update channel.

4. Select the update option for Authorino 1.2.1.

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-20595 - Pipeline fails to run if thehttp_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.

SRVKS-1301 (previously documented as RHOAIENG-18590) - TheKnativeServing 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-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 KServe** 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 described in vLLM NVIDIA GPU ServingRuntime for KServe.

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 loops (**dsl.ParallelFor**) or condition groups (**dsl.If**), the UI displays a Running status for the loops and 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 workbench 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 asODH-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 asODH-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 asRHODS-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 asRHODS-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 workbench 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 workbench image within the workbench, you cannot execute the pipeline, even after restarting the workbench.

Workaround

1. Stop the running workbench.
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 workbench.
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 asNOTEBOOKS-210) - A Jupyter notebook fails to export as a PDF file in Jupyter

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

Workaround

None.

RHOAIENG-1210 (previously documented asODH-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 asODH-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 asRHODS-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 workbenches indefinitely

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

Workaround

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

RHOAIENG-1157 (previously documented asRHODS-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 asRHODS-6356) - The basic-workbench creation process fails for users who have never logged in to the dashboard

The dashboard’s **Administration** page for basic workbenches displays users who belong to the user group and admin group in OpenShift. However, if an administrator attempts to start a basic workbench on behalf of a user who has never logged in to the dashboard, the basic-workbench 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) - Administration page for basic workbenches shows users who have lost permission access

If a user who previously started a basic workbench 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 basic workbench on the **Administration** page. As a consequence, an administrator is able to restart a basic workbench that belongs to a user whose permissions were revoked.

Workaround

None.

RHODS-4799 - Tensorboard requires manual steps to view

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

Workaround

When you start your basic workbench, 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 cluster administrators. To learn more, see Introduction to Red Hat OpenShift AI .