HITACHI

Hitachi Vantara

Accelerate your AI Journey with Red Hat AI and NVIDIA with the Hitachi Integrated Systems Platform and Virtual Storage Platform One

Reference Architecture Guide

MK-SL-373-01
November 2025

The open source content used in Hitachi Vantara products may be found within the Product documentation or you may request a copy of such information (including source code and/or modifications to the extent the license for any open source requires Hitachi make it available) by sending an email to OSS_licensing@hitachivantara.com.

Feedback

Hitachi Vantara welcomes your feedback. Please share your thoughts by sending an email message to Docs-Feedback@hitachivantara.com. To assist the routing of this message, use the paper number in the subject and the title of this white paper in the text.

Revision history

Changes	Date
Minor updates	November 2025
Initial release	August 2025

Reference Architecture Guide

Red Hat AI is a portfolio of products, including both Red Hat Enterprise Linux AI and Red Hat OpenShift AI, and services that accelerate the development and deployment of AI solutions across hybrid cloud environments.

This paper provides a reference configuration of Red Hat OpenShift with the OpenShift AI feature enabled, deployed on the Hitachi Integrated Systems platform leveraging VSP One Storage Platforms and NVIDIA certified data center servers. It leverages the latest capabilities and services provided by OpenShift AI and NVIDIA NIM microservices.

Red Hat OpenShift AI includes the components of Red Hat Enterprise Linux AI and extends and scales the capabilities with Kubernetes-based containerized orchestration and the ability to distribute AI workloads across multiple nodes. It is a platform that empowers enterprises to develop and deploy AI-driven solutions efficiently across hybrid cloud environments, facilitating the creation and delivery of AI-enabled applications at scale with high reliability and security. Additionally, Red Hat OpenShift AI supports model serving and hosting, making real-time AI inference and integration seamless.

A key element in the successful deployment of an AI platform is having a robust and flexible infrastructure (compute, storage, network) that can meet a wide variety of requirements in a highly dynamic environment. Hitachi infrastructure with Red Hat OpenShift AI and NVIDIA AI Enterprise provides a highly available and high-performance infrastructure for AI workloads.

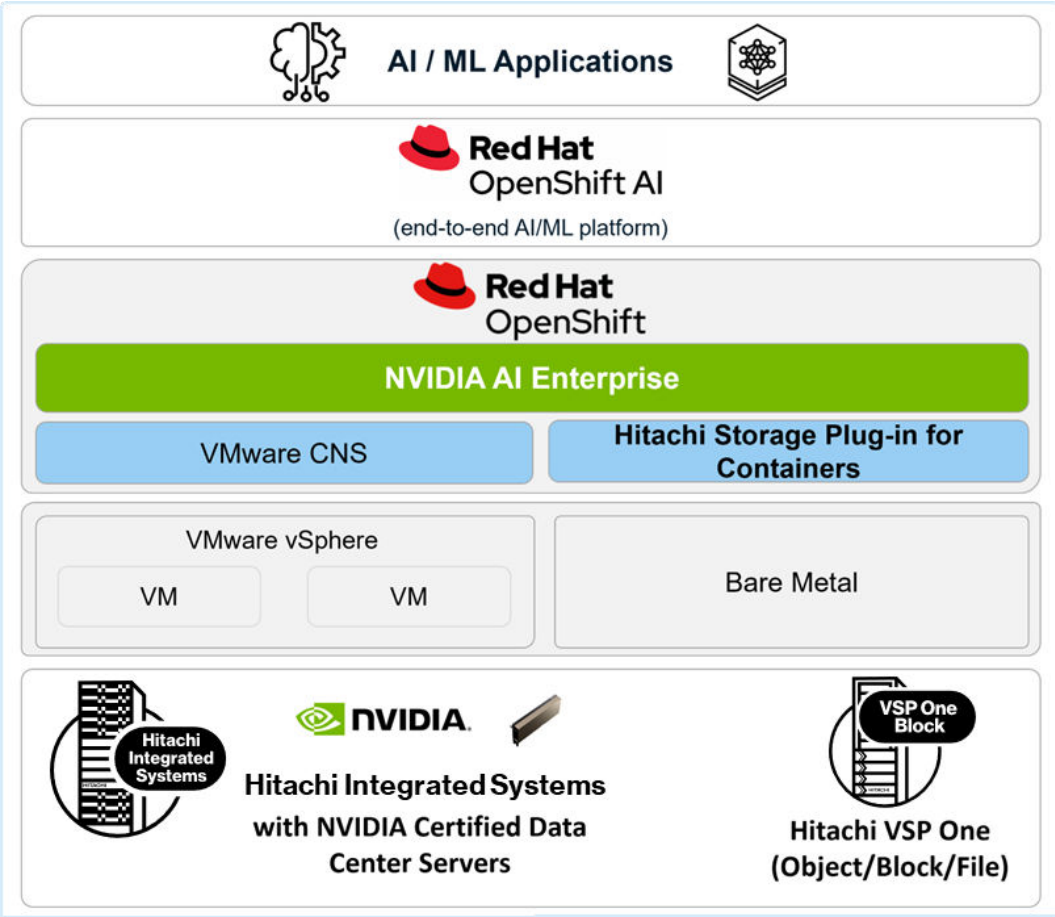
The Hitachi Integrated Systems platform includes a Hitachi Storage CSI driver supported with Hitachi Storage Plug-in for Containers (HSPC). Using these proven CSI (Container Storage Interface) storage integrations, you can provide persistent reliable storage for AI workloads.

This reference architecture also provides the reference design for a build-your-own AI platform using Hitachi Integrated Systems with Red Hat OpenShift and NVIDIA GPUs.

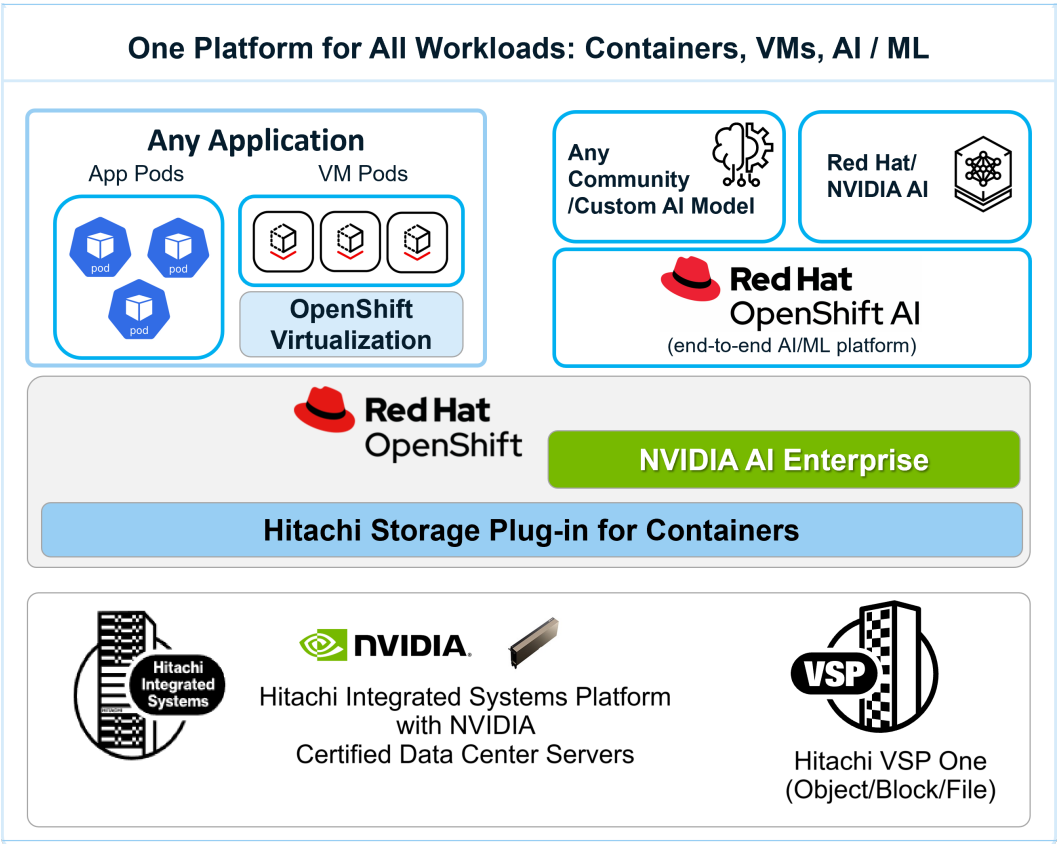
The intended audience of this document is IT administrators, system and AI/ML architects, consultants, and sales engineers to assist in planning, designing, and implementing a Hitachi Integrated Systems platform with OpenShift Container Platform solutions.

Overview

Red Hat OpenShift Container Platform is a successful container orchestration platform and is one of the container orchestration and virtualization solutions available in the Hitachi Integrated Systems solution portfolio. The following figure shows a high-level diagram of Red Hat OpenShift Container Platform with OpenShift AI alongside NVIDIA AI Enterprise. There is flexibility in the hypervisor that can be used. This reference guide leverages KVM that as part of the OpenShift Container Platform/OpenShift Virtualization deployment. It also integrates Hitachi Virtual Storage Platform for persistent storage, with Hitachi Storage Plug-in for Containers enabling automated storage provisioning to the Kubernetes clusters.



The integration of the Hitachi Integrated Systems solution with Red Hat OpenShift and NVIDIA provides a unified and full-stack hybrid platform to deploy and manage all your containerized applications, virtual machines (VMs), and AI/ML workloads across a hybrid environment, reducing operational complexity and paving the way for modernization. A high-level architecture of this platform for all workloads can be seen in the following illustration.



Red Hat OpenShift AI is a powerful artificial intelligence (AI) platform for data scientists and developers of AI applications. Red Hat OpenShift AI provides the tools that let you rapidly develop, train, test, deploy and monitor AI/ML models on-premises and/or in the public cloud.

A key benefit we concluded with this reference pattern is that it helps to accelerate the time to value of AI projects. Before evaluating Red Hat AI, we observed that our customers had to spend resources to find the right combination of software packages suitable to run AI/ML workloads. This is proven to be time consuming, and many times required significant expertise to have an environment up and running to start prototyping/deploying AI/ML applications.

The very interactive and collaborative UI enables data scientists and data engineers to eliminate spending too much time on the infrastructure/DevOps processes and instead use that time to focus on developing/training, fine-tuning their models, and deploying/ integrating those models into workloads/applications in production environments.

We see significant benefits in the distributed workloads capability for more efficient model tuning. Other areas such as model monitoring and data pipelines were not extensively evaluated in this phase of paper.

This reference architecture was validated by the following use cases:

- Generative AI deployment with NVIDIA NIM microservices on OpenShift AI (on page 19)
- Model training from a pretrained model using OpenShift AI (on page 29)
- Testing a retrained model (on page 36)

Follow the steps in the solution design, implementation, and use case validation sections to learn how to fast track and enable these AI accelerated capabilities for your business when using Hitachi Integrated Systems with Red Hat OpenShift Container Platform and OpenShift AI.

Solution components

This section outlines the components used in this reference architecture.

Hitachi Integrated Systems

The Hitachi Integrated Systems platform is optimized, preconfigured, and pretested to host workloads and platform stacks such as VMware vSphere/ VMware Cloud Foundation, Red Hat OpenShift, Kubernetes, Azure Local, Google Anthos, and bare metal. It offers a broad range of compute, network and storage components that can be scaled and configured independently to eliminate overprovisioning. You have a choice of operating environments to maximize your flexibility.

With Hitachi Integrated Systems, you can choose between single-rack configurations and multi-rack configurations with flexible local and VSP One Storage options across block, file, and object.

Hitachi Virtual Storage Platform One Block

VSP One Block arrives pre-configured, including Dynamic Drive Protection (DDP) groups. DDP replaces traditional RAID groups in VSP One Block storage systems, providing the resilience of RAID 6 with distributed spare space and support for an arbitrary number of drives (from 9-32 per group). Adding drives one (or more) at a time is also supported. DDP improves the resilience of the appliance by dramatically lowering rebuilding times.

Hitachi Virtual Storage Platform One File

Hitachi Virtual Storage Platform One File (VSP One File) is a modernized Hitachi NAS that is focused on improving customer experience, delivering on operational simplification, and enabling better agility.

The VSP One File series provides an easy and modern way to manage, monitor, and configure file systems and data across the organization.

Hitachi Virtual Storage Platform One File includes the following 3 dedicated models:

- VSP One File 32 – intended for cost sensitive deployments, and/or hybrid flash deployment that have no short/medium term plan to move away from their 10GbE network infrastructure.
- VSP One File 34 – targeted for technology buyers that desire the best price/performance, an all-flash deployment, and have standardized on 25GbE network connectivity or are planning to move short/medium term to 25GbE.
- VSP One File 38 – a performance model and built for leading technology pioneers that require the most file performance, and highest rack network port density to address all-flash use cases and applications.

For high SLA workloads, all VSP One File models support SMB3 multichannel, NFS 4.1 multi-pathing, Nconnect, and link aggregation control protocol (LACP) to improve availability and/or higher bandwidth for Windows and Linux clients beyond the limit of a single connection.

Hitachi Virtual Storage Platform One Object

VSP One Object offers massive scale, cloud capabilities, and broad protocol support. It supports all-flash configurations and policy-driven tiered storage, allowing performance and capacity to scale independently. It quickly ingests large volumes of small objects and cost-effectively stores very large objects.

Customers benefit from a diverse array of configurations and deployment options. These include traditional archive and backup solutions, as well as modern cloud-native environments, AI/ML applications, and analytics workloads. The platform is optimized for a compact form factor, fitting an entire cluster into a 4U footprint. This enables real-time analytics at the edge, eliminating the need to transfer data to a central location for analysis.

Additional hardware components

The tested solution used specific features based on the following hardware. You can use any qualified server platform defined in the Product Compatibility Guide. For more information, see https://compatibility.hitachivantara.com/.

Hardware	Description	Version	Quantity
Hitachi Advanced Server HA800 series (for OCP bare metal cluster)	▪ 2 × Intel(R) Xeon(R) Gold 6454S processors ▪ 16 × 32 GB DIMM, 512 GB memory ▪ NS204-u RAID1 for boot ▪ SN1610E 32 Gb 2p FC HBA ▪ 2 × Intel(R) Eth E810-XXVDA2 NICs ▪ 1 × NVIDIA L40s GPU (only on worker nodes)	iLO 6 BIOS:U54 v1.30	6
Hitachi VSP One Block (for OCP cluster)	▪ 1 TB cache ▪ 24 × 3.8 TB NVMe drives ▪ 4 × 32 Gbps Fibre Channel ports	A3-04-01-40/00	1
Cisco Nexus 93180YC-FX switch (leaf)	▪ 48-port 10/25 GbE ▪ 6-port 40/100 GbE	NXOS 10.3(7)	2
Brocade G720	▪ 48-port 16/32Gbps Fibre Channel switch	9.1.1c	2

Software components

Software	Version
Hitachi Storage Microcode/Firmware Version	A3-04-01-40/00
Hitachi Storage Plug-in for Containers	3.15.0
Red Hat OpenShift Container Platform (OCP)	4.16.28
Red Hat OpenShift AI	2.16.1
NVIDIA GPU Operator	24.9.2

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform (OCP) provides a single platform to build, deploy, and manage applications consistently across on-premises and hybrid cloud deployments. OCP provides the control plane and data plane within the same interface. OCP provides administrator views to deploy operators, monitor container resources, manage container health, manage users, work with operators, manage pods and deployment configurations, as well as define storage resources.

OCP also provides a developer view to allow users to deploy application resources from various pre-defined resources such as YAML files, Docker files, Catalogs, or GIT within user-defined namespaces. With OCP `kubectl`, a native binary of Kubernetes is complemented by the `oc` command, which provides further support for OCP resources, such as deployment and build configurations, routes, image streams, and tags. OCP provides a GUI and a CLI interface.

Red Hat OpenShift AI

Red Hat OpenShift AI provides an AI platform for building, training, tuning, deploying and monitoring AI-enabled applications and predictive and foundation models at scale across hybrid cloud environments. Red Hat OpenShift AI helps accelerate AI innovation, drive operational consistency, and optimize access to resources when implementing trusted AI solutions.

See the following for more details of OpenShift AI and RHEL AI:

https://www.redhat.com/en/resources/openshift-ai-overview

https://www.redhat.com/en/resources/foundation-model-platform-gen-ai-datasheet

NVIDIA GPU Operator

The NVIDIA GPU Operator uses the operator framework within Kubernetes to automate the management of all NVIDIA software components needed to provision GPUs. These components include the NVIDIA drivers (to enable CUDA), Kubernetes device plugins for GPUs, the NVIDIA Container Runtime, automatic node labeling, DCGM-based monitoring, and more.

NVIDIA AI Enterprise

NVIDIA AI Enterprise is a comprehensive, cloud-native software platform designed to accelerate data science pipelines and streamline the development and deployment of production-grade co-pilots and generative AI applications. It offers easy-to-use microservices that deliver optimized model performance, coupled with enterprise-grade security, support, and stability. This ensures a seamless transition from prototype to production for businesses that rely on AI to operate.

NVIDIA NIM for Large Language Models (LLMs)

NVIDIA NIM, part of NVIDIA AI Enterprise, is a set of easy-to-use microservices for accelerating the deployment of generative AI models across the cloud, data center, and workstations. It supports everything from open-source community models to NVIDIA AI Foundation models, as well as custom AI models.

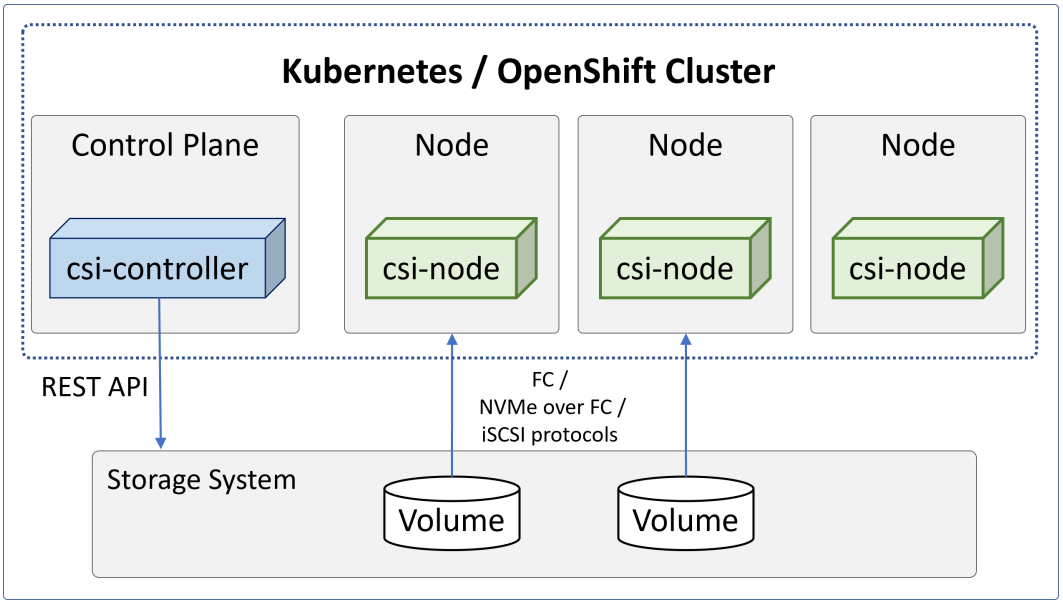
NVIDIA NIMs make it easy for IT and DevOps teams to self-host large language models (LMMs) in their own managed environments allowing developers to quickly build enterprise-grade AI applications with minimum effort, reducing development time and cost. NIM offers the fastest path to inference with unparalleled performance.

Hitachi Storage Plug-in for Containers

Hitachi Storage Plug-in for Containers (HSPC) is a software component that contains libraries, settings, and commands that you can use to create a container to run stateful applications. It enables stateful applications to persist and maintain data after the lifecycle of the container has ended. Storage Plug-in for Containers provides persistent volumes from Hitachi Dynamic Provisioning (HDP) or Hitachi Thin Image (HTI) pools to bare metal or hybrid deployments using Fibre Channel, NVMe over Fibre Channel, or iSCSI protocols. iSCSI protocol is supported for both bare metal and virtual environments.

Storage Plug-in for Containers integrates Kubernetes or OpenShift with Hitachi storage systems using Container Storage Interface (CSI).

The following diagram illustrates a container environment where Storage Plug-in for Containers is deployed.



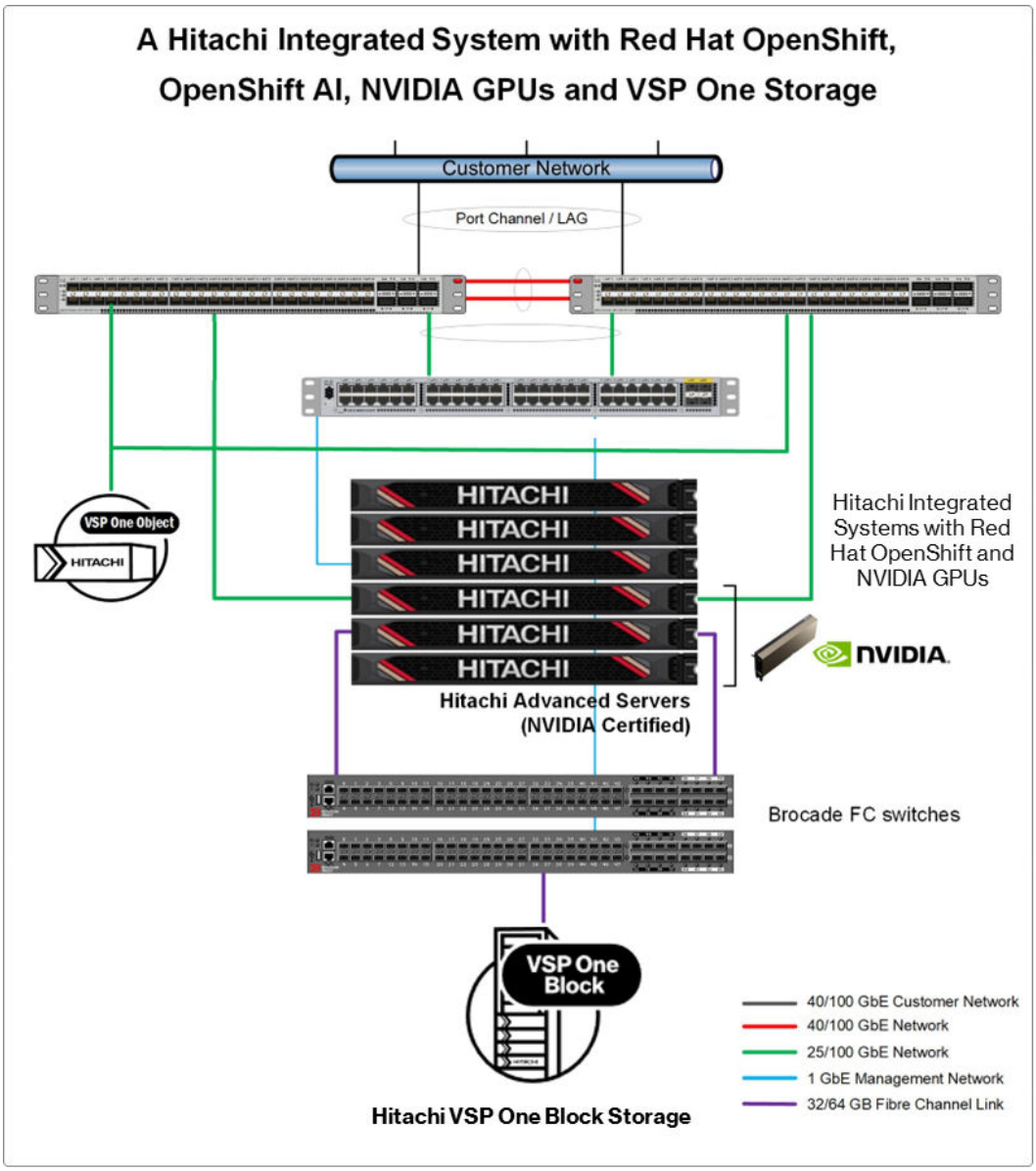
Solution design

This section outlines a detailed solution example for the Hitachi Integrated Systems solution with Red Hat OpenShift and NVIDIA GPUs.

Hitachi Integrated Systems platform infrastructure components

The following figure shows a high-level architecture of Hitachi Integrated Systems clusters used to validate the Red Hat OpenShift AI. It includes the following components:

- Hitachi Integrated Systems platform with Red Hat OpenShift Container Platform and OpenShift AI:
  - OpenShift cluster bare metal configured with six Hitachi Advanced Server nodes: three nodes for the control plane and three for worker nodes to run AI/ML workloads.
  - A Hitachi Virtual Storage Platform One storage system for persistent storage for standard Pods and VM Pods (a VSP E1090, VSP 5200, or VSP 5600 storage systems can be used as well).
- The following network infrastructure is used for these clusters:
  - Two Cisco spine Ethernet switches
  - Two Cisco leaf Ethernet switches
  - One Cisco management switch



The clusters used for this reference architecture were configured following the standard guidelines from Hitachi Integrated Systems.

As stated earlier, the solution in this reference architecture is based on the Hitachi Integrated Systems platform with OpenShift and VSP/VSP One storage which supports multiple protocols with options for Fibre Channel (FC), FC-NVMe, and iSCSI. In addition, this solution can be integrated/deployed together with Hitachi VSP One File and VSP One Object to offer all the storage capabilities to meet the needs for the AI/ML workloads.

This reference design is targeted at medium and enterprise customers requiring an accelerated computing solution reducing the operational burden to develop and deploy AI model enhanced application/workload services. For even larger enterprises with extreme scalability requirements, the Hitachi iQ solution with NVIDIA HGX offers pre-built platforms with bespoke customization and delivery, combining outstanding software, infrastructure, distributed parallel file systems, and expertise in unified and scalable AI development solutions.

For building a large enterprise datacenter using the latest high-speed, low latency solution, follow the Hitachi iQ guides and reference architecture:

- Hitachi's AI Solutions Stack for Enterprise Workloads: https://www.hitachivantara.com/en-us/pdf/datasheet/hitachi-iq-with-nvidia-hgx.pdf
- Hitachi iQ with NVIDIA HGX - Reference Architecture: https://www.hitachivantara.com/en-us/gated-forms/iq-with-hgx-h100-powered-with-hcsf-hcp
- More details about the Hitachi Integrated Systems infrastructure can be found at https://www.hitachivantara.com/en-us/products/integrated-systems/convergedinfrastructure

Deploy OpenShift Container Platform

This guide does not cover the step-by-step details of how to implement OCP. Follow Red Hat OCP documentation for the setup of the cluster.

One OCP bare metal cluster with Hitachi VSP One storage and Hitachi Storage Plug-in for Containers has been configured to support the different use cases described in this guide. In addition, the worker nodes have been equipped with NVIDIA L40s GPUs.

The following are additional details about the cluster used for this validation.

```
[root@jputilitysrv1 ocpjpai]# oc version
Client Version: 4.16.28
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: 4.16.28
Kubernetes Version: v1.29.10+67d3387
[root@jputilitysrv1 ocpjpai]#
[root@jputilitysrv1 ocpjpai]# oc get nodes
NAME                                  STATUS   ROLES                  AGE   VERSION
ocpjpai-master-1.ocpjpai.ocp.sce.lab  Ready    control-plane,master   22d   v1.29.10+67d3387
ocpjpai-master-2.ocpjpai.ocp.sce.lab  Ready    control-plane,master   22d   v1.29.10+67d3387
ocpjpai-master-3.ocpjpai.ocp.sce.lab  Ready    control-plane,master   22d   v1.29.10+67d3387
ocpjpai-worker-1.ocpjpai.ocp.sce.lab  Ready    worker                 22d   v1.29.10+67d3387
ocpjpai-worker-2.ocpjpai.ocp.sce.lab  Ready    worker                 22d   v1.29.10+67d3387
ocpjpai-worker-3.ocpjpai.ocp.sce.lab  Ready    worker                 22d   v1.29.10+67d3387
[root@jputilitysrv1 ocpjpai]#
```

For detailed OpenShift Container Platform installation procedures, see the Red Hat documentation at https://docs.redhat.com/en/documentation/openshift_container_platform/4.16/.

Deploy Hitachi Storage Plug-in for Containers

Hitachi Storage Plug-in for Containers (HSPC) is easily deployed to OpenShift using the Operator, which can be installed from OperatorHub.

This guide does not cover the step-by-step how to install HSPC, follow the https://docs.hitachivantara.com/v/u/en-us/adapters-and-drivers/3.15.x/mk-92adptr142 or the steps described in the https://docs.hitachivantara.com/v/u/en-us/application-optimized-solutions/mk-sl-210.

In summary, these steps include:

- Install Hitachi Storage Plug-in for Containers.
- Configure Secret settings to access Hitachi VSP One storage systems.

- Configure StorageClass settings.
- Configure Multipathing (Fibre Channel or iSCSI).

Specific steps for how to configure Storage Classes will be covered as part of each of the use cases in the Solution implementation and validation section.

Note: If there is a previous version of Storage Plug-in for Containers, remove it before performing the installation procedure.

HSPC and VSP Host Groups

Host groups required for Storage Plug-in for Containers are automatically created by Storage Plug-in for Containers. Storage Plug-in for Containers automatically searches host groups and iSCSI targets based on the name.

To use existing host groups, rename them according to the naming rule. For details, see *Host group and iSCSI target naming rules* in the *Hitachi Storage Plug-in for Containers Quick Reference Guide* at https://docs.hitachivantara.com/v/u/en-us/adapters-and-drivers/3.14.x/mk-92adptr142.

Note: Storage Plug-in for Containers will overwrite host mode options even if existing host groups have other host mode options.

HSPC and VSP Resource Groups

You can partition storage system resources by limiting the LDEV ID range added to the resource group for a specific Kubernetes cluster. You can also isolate impacts between Kubernetes clusters. The following requirements should be met:

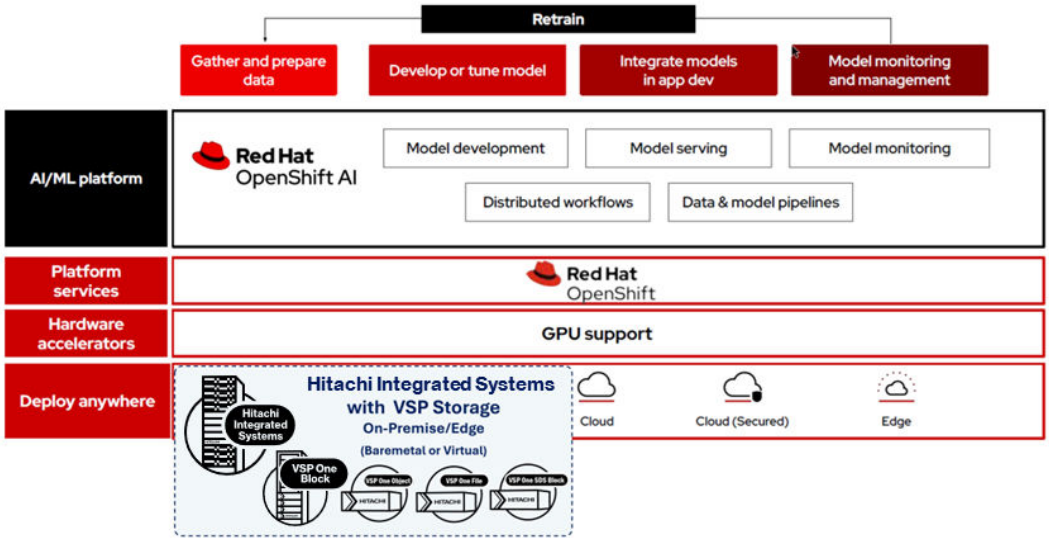
- Multiple Kubernetes clusters can share one resource group.
- Storage system users must have access only to the resource group that they created. The storage system user must not have access to other resource groups.
- Create a pool from pool volumes with the resource group that you have created.
- Allocate the necessary number of undefined LDEV IDs to the resource group.
- Allocate the necessary number of undefined host group IDs to the resource group for each storage system port defined in StorageClass. The number of host group IDs must be equal to the number of hosts for all ports.

For details, see the Hitachi Storage Plug-in for Containers documentation at https://docs.hitachivantara.com/search/all?query=Hitachi%252BStorage%252BPlug-in%252Bfor%252BContainers&content-lang=en-US

Red Hat OpenShift AI with Hitachi Integrated Systems and VSP One Storage

Before you begin

- You must have a subscription for Red Hat OpenShift AI.
- If you are planning to use NVIDIA NIMs with OpenShift AI, you must have NVIDIA AI Enterprise subscription or be an NVIDIA Developer Program member.
- Have RedHat OpenShift cluster 4.14 or later installed on a bare metal or virtual infrastructure:
  - You must have at least 2 worker nodes, and at least one of them with an NVIDIA GPU.
- You can use any of the four installation methods (user-provisioned, installer-provisioned, assisted, agent-based installer) of an OCP cluster on bare metal.
- Use Hitachi Virtual Storage Platform as the backend storage to the OCP cluster and OpenShift Virtualization.
- Hitachi Storage Plug-in for Containers (HSPC). A `storageClass` defined on the OCP cluster with HSPC as the provisioner, and makes sure the storage class is configured as default. Follow the OpenShift Container Platform documentation to configure a default storage class: Changing the default storage class.



Install and configure OpenShift AI and its components

After the prerequisites have been met, you can follow these steps to prepare and install Red Hat OpenShift AI and its components.

Before you begin

After OpenShift cluster is ready, log in to the console and install the following operators from OperatorHub:

- Node Feature Discovery Operator
  - The Node Feature Discovery Operator manages the detection of hardware features and configuration in a Kubernetes cluster by labeling the nodes with hardware-specific information. The Node Feature Discovery (NFD) will label the host with node-specific attributes, like PCI cards, kernel, or OS version, and more.
- NVIDIA GPU Operator
  - The NVIDIA GPU Operator uses the operator framework within Kubernetes to automate the management of all NVIDIA software components needed to provision and monitor GPUs. These components include the NVIDIA drivers (to enable CUDA), Kubernetes device plugin for GPUs, the NVIDIA Container Runtime, automatic node labelling and NVIDIA DCGM exporter.
  - After the GPU Operator and cluster policy have been created, you can check the status with the following command:

    ```
    oc get pods -n nvidia-gpu-operator
    ```

```
[root@jputilitysrv1 ocpjpai]# oc get pods -n nvidia-gpu-operator
NAME                                          READY   STATUS      RESTARTS        AGE
gpu-feature-discovery-z7flc                   2/2     Running     0               10d
gpu-operator-5bf9695f4c-dpnm6                 1/1     Running     0               10d
nvidia-container-toolkit-daemonset-jrnrh      1/1     Running     0               10d
nvidia-cuda-validator-kk79x                   0/1     Completed   0               10d
nvidia-dcgm-69vcp                             1/1     Running     0               10d
nvidia-dcgm-exporter-nks4v                    1/1     Running     0               10d
nvidia-device-plugin-daemonset-cj5lr          2/2     Running     0               10d
nvidia-driver-daemonset-416.94.202412170927-0-wmfdj   2/2     Running     0               10d
nvidia-node-status-exporter-5b92f             1/1     Running     0               10d
nvidia-operator-validator-drdfm               1/1     Running     0               10d
[root@jputilitysrv1 ocpjpai]#
```

For details how to install the NFD and GPU Operator, see the instructions at NVIDIA GPU Operator on Red Hat OpenShift Container Platform.

In addition to the NFD and GPU Operators, to support the KServe component, which is used by the single-model serving platform to serve large language models, you must install the following Operators:

- Red Hat OpenShift Serverless
- Red Hat OpenShift Service Mesh

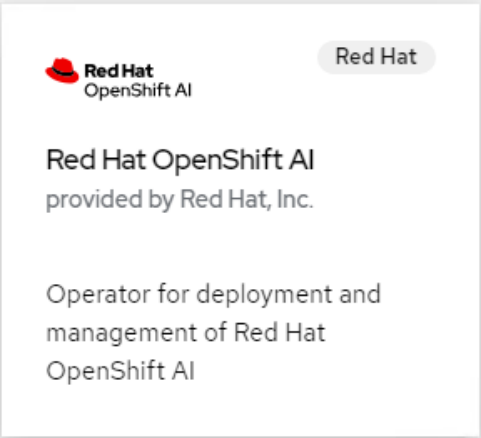
For more details, see About the single-model serving platform.

If there is a need to add an authorization provider for the single-model serving platform, install the Red Hat Authorino (Technical Preview). For more details, see Adding an authorization provider for the single-model serving platform.

After these operators are installed and in a ready state, now proceed to install Red Hat OpenShift AI operator using the following procedure. This operator can be installed either from the CLI or web console. In this guide we are following the installation using the OpenShift web console.

Procedure

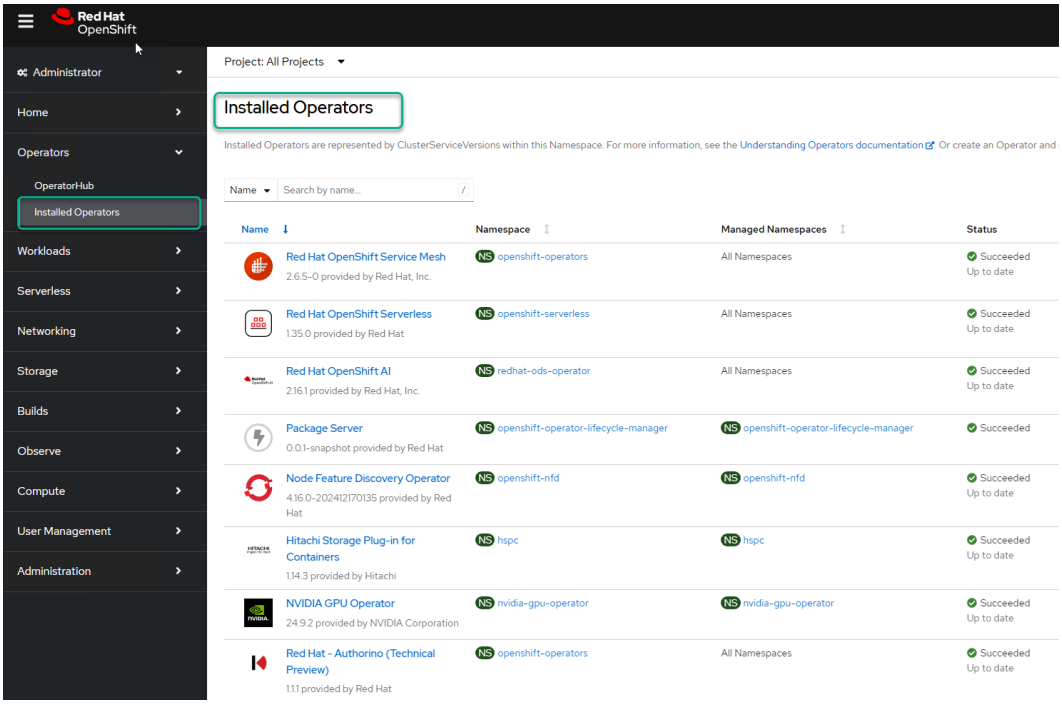
1. In the Red Hat OpenShift web console, navigate to **Operators** > **OperatorHub** and filter by the keyword *Red Hat OpenShift AI*.
2. Select the Red Hat OpenShift AI Operator tile to open the RedHat OpenShift AI information pane.



3. Click **Install**.
4. Keep the default parameters. To start the installation, click **Install** again.
5. After the installation completes successfully, select **Create DataScienceCluster**.
6. Keep the default values and click **Create**.
7. Wait until the **All instances** tab shows the status **Ready** for all components.

Result

After all the components/operators are installed, you can verify all the installed operators.



Access OpenShift AI dashboard

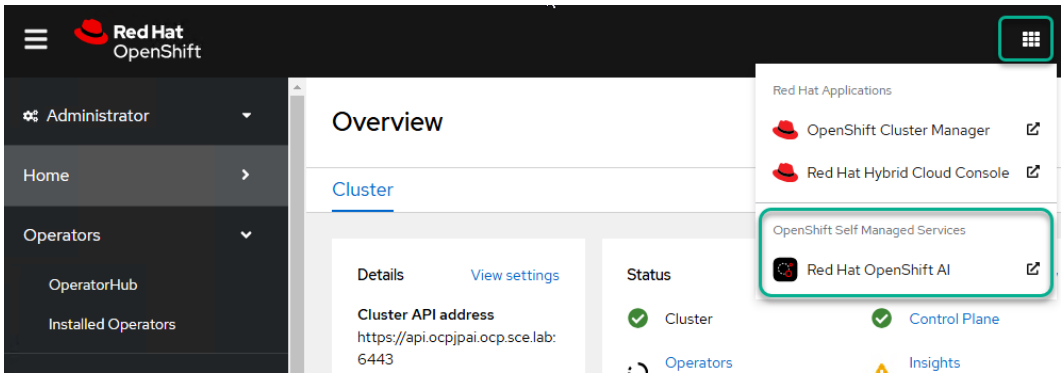
Before we access OpenShift AI dashboard, let us get familiar with some components of a data science project:

- Workbenches are instances of your development and experimentation environment. They typically contain IDEs, such as JupyterLab, RStudio, and Visual Studio Code.
- A Cluster storage is a volume that persists the files and data you're working on within a workbench. A workbench has access to one or more cluster storage instances.
- Data connections contain configuration parameters that are required to connect to a data source, such as an S3 object bucket.
- Pipelines contain the Data Science pipelines that are executed within the project.
- Models and model servers allow you to quickly serve a trained model for real-time inference. You can have multiple model servers per data science project. One model server can host multiple models.

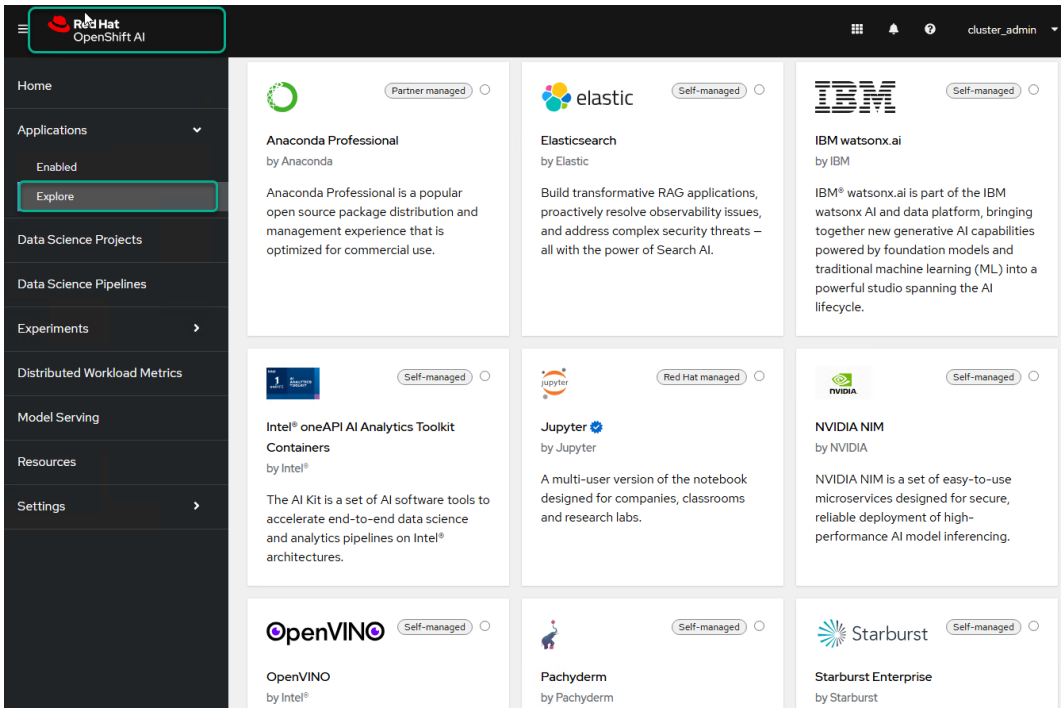
After OpenShift AI is installed, you can access the URL for your OpenShift AI console and share the URL with users to let them log in and work on their models.

Procedure

1. To access the OpenShift AI web console, click the application launcher ( ) from the OCP console, located on the right side of the top navigator bar. Click and select Red Hat OpenShift AI from the drop-down menu.



2. Log in with your OpenShift credentials, and you will be presented with the OpenShift AI landing page where you can enable applications and create Data Science Projects.



All the data science projects you create will benefit from HSPC, which provides robust and high-performance persistent volumes. Additionally, HSPC together with HRPC (Hitachi Replication Plug-in for Containers) offers advanced disaster recovery (DR) protection, ensuring that your AI projects and applications maintain high availability and resilience. This means you can focus on innovation and development, knowing that your data and applications are secure and reliable.

Solution implementation and validation – use cases

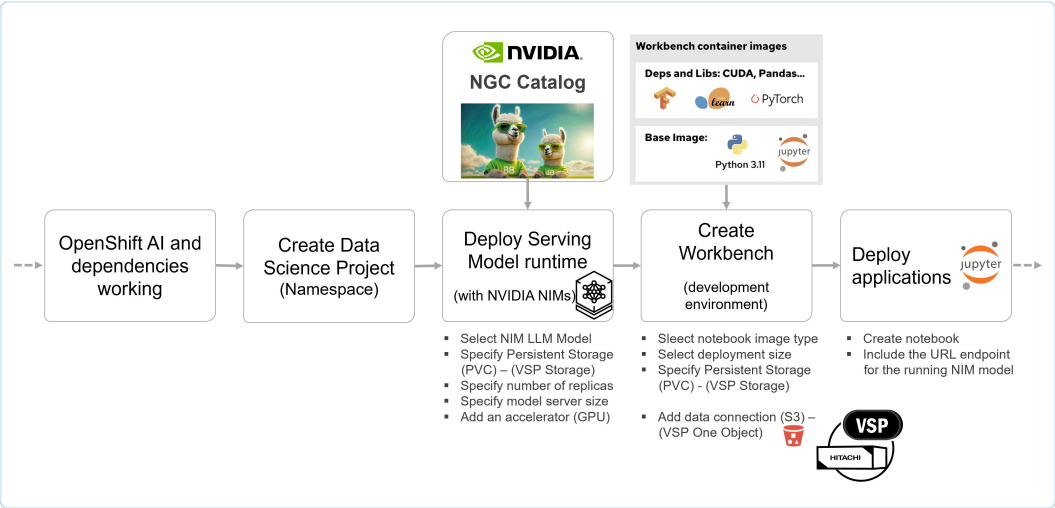
This reference architecture was validated by the following use cases:

- Generative AI deployment with NVIDIA NIM microservices on OpenShift AI (on page 19)
- Model training from a pretrained model using OpenShift AI (on page 29)
- Testing a retrained model (on page 36)

Generative AI deployment with NVIDIA NIM microservices on OpenShift AI

In this test case we are going to show how to deploy generative AI applications leveraging NVIDIA NIM microservices and OpenShift AI on top of the Hitachi Integrated Systems platform and VSP One infrastructure.

See the following flow diagram that showcases the high-level steps to deploy a GenAI application using NVIDIA NIM microservices with OpenShift AI.

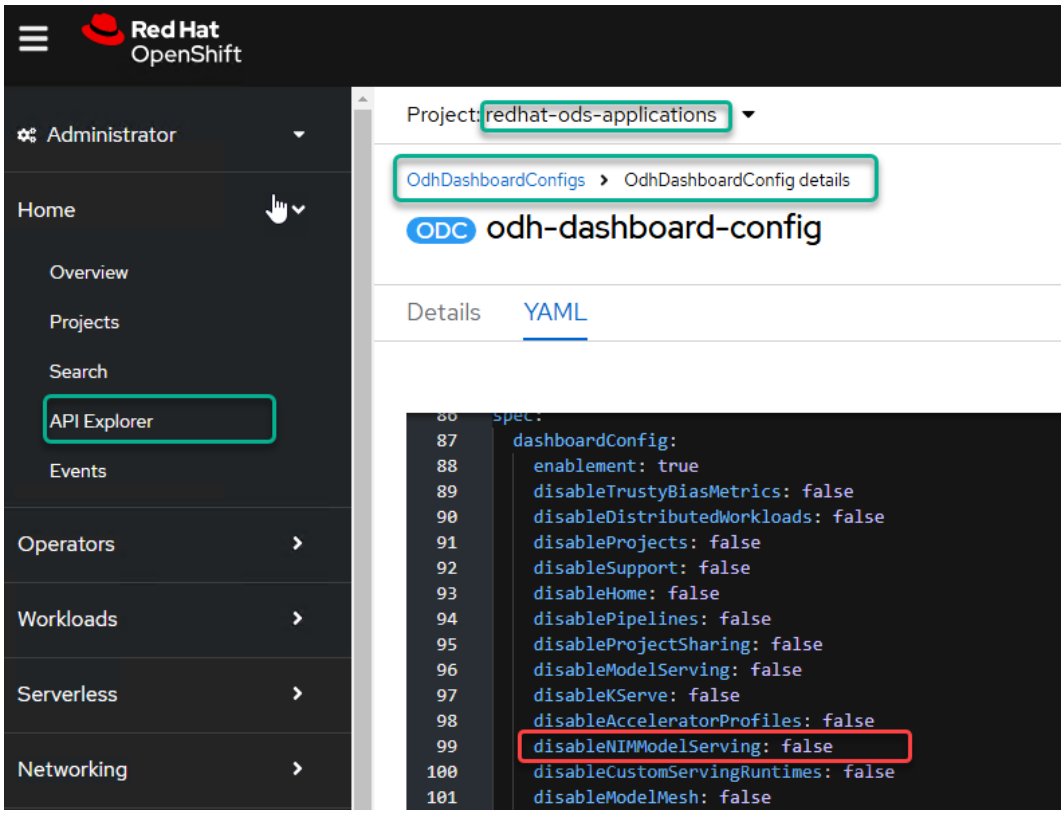


Enable the NVIDIA NIM application

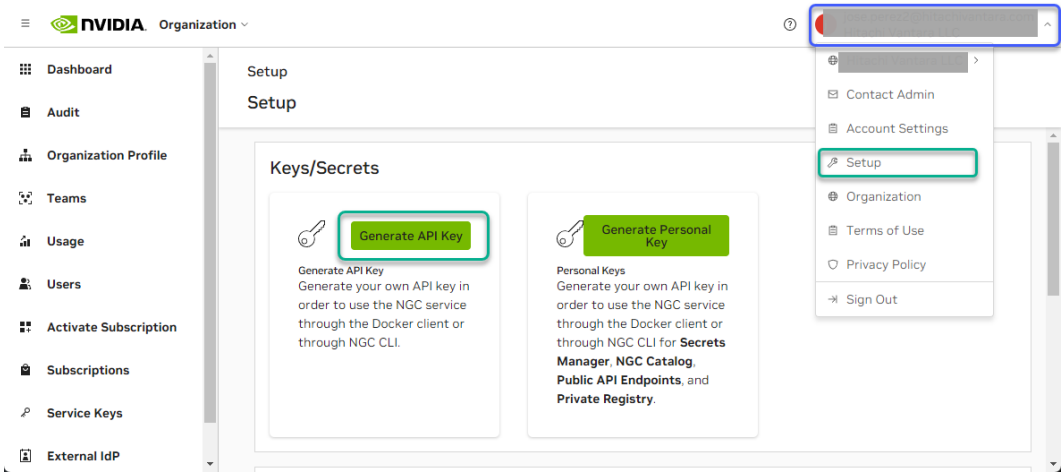
Procedure

1. Before enabling the NIM application on the OpenShift dashboard, make sure to enable it in the `OdhDashboardConfigs` instance under **Home** > **API Explorer** in the Administrator perspective of the OpenShift console. The requirement is to set `disableNIMModelServing` to `false`.

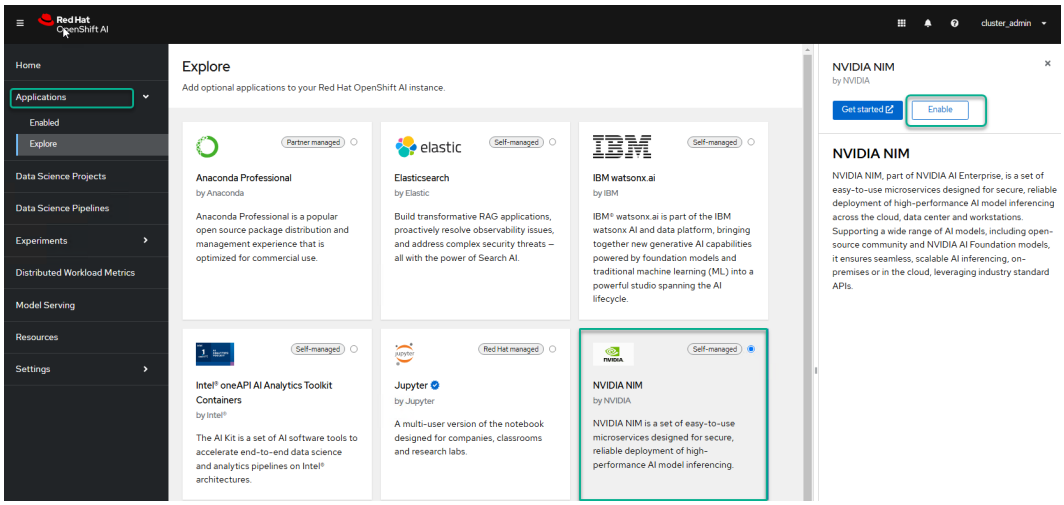
   For more details, see Editing the dashboard configuration file.



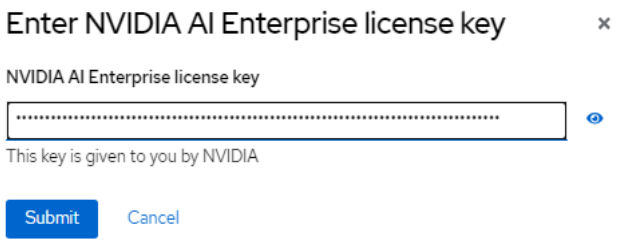
2. Generate an API key from NVIDIA. To do this, log in to the NVIDIA NGC catalog and generate an API key using the option **Generate API Key** under **profile menu** > **Setup**.



3. From the Red Hat OpenShift AI dashboard, select the **Explore** option under Applications, and select the **NVIDIA NIM** tile.

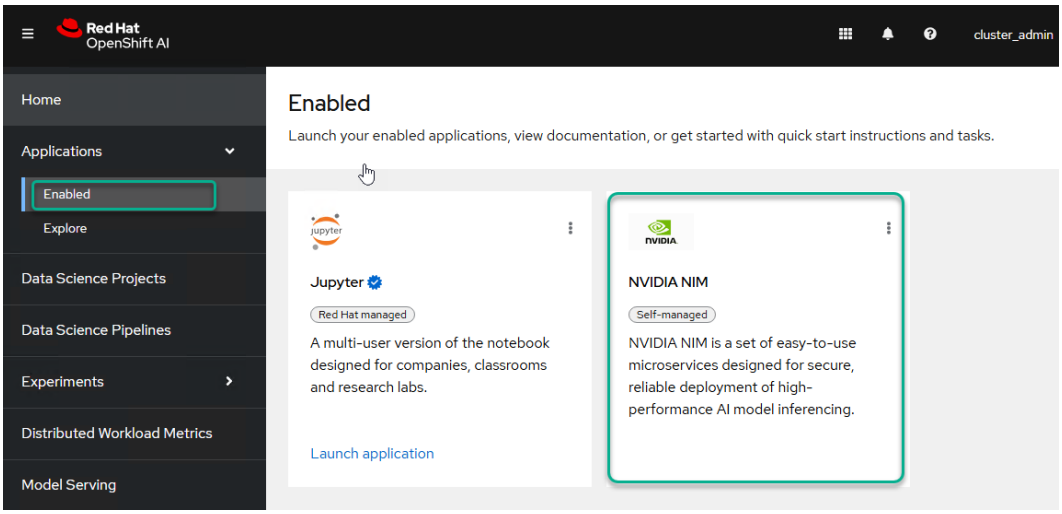


4. Click **Enable**, and enter the API key obtained from NVIDIA NGC catalog, and then click **Submit** to enable the NVIDIA NIM tile.



Result

When enabled, you can verify it by selecting the Enabled option under Applications.



Create a Data Science Project

To implement a data science workflow, you need to create a data science project. A data science project allows you and your team to organize and collaborate on resources within separated namespaces. From a project you can create multiple workbenches, each with their own Jupyter notebook environment, and each with their own data connections and cluster storage. In addition, the workbenches can share models and data with pipelines and model servers, all in a single project.

Procedure

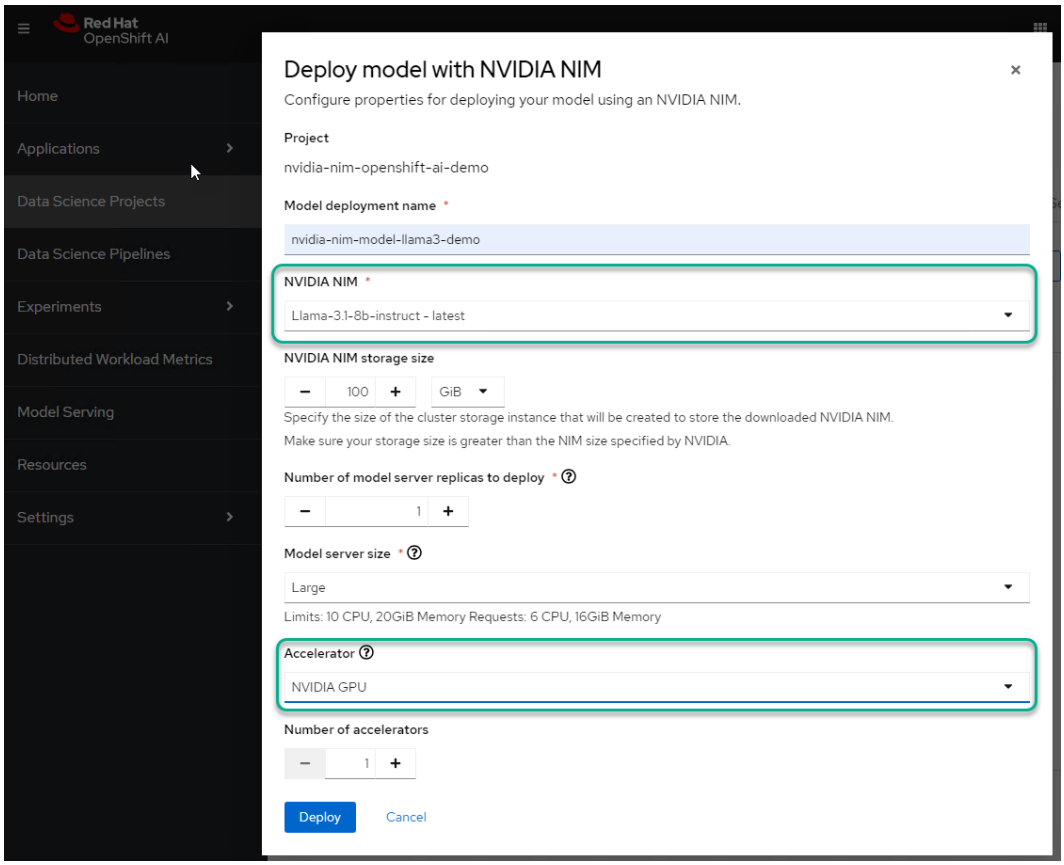
1. From the OpenShift AI dashboard, navigate to the **Data Science Projects** tab.
2. Click **Create Project**, and enter a name and description for the project.
3. Click **Create**.

Deploy an NVIDIA NIM serving model

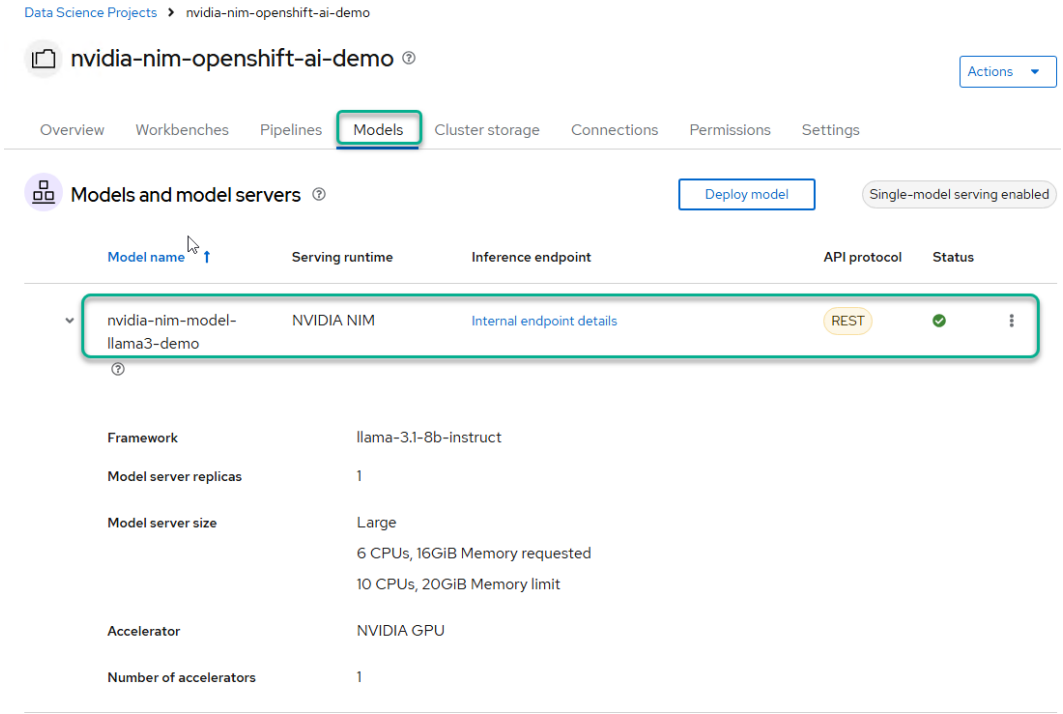
Inside the created project is where all the additional resources for this test case will be deployed.

Procedure

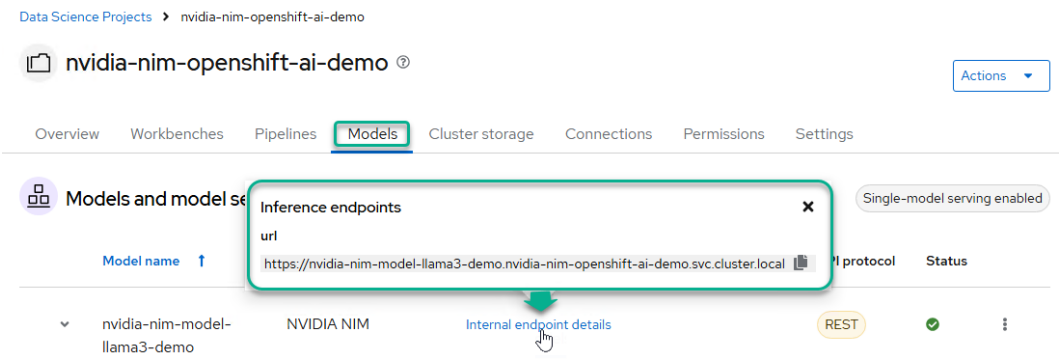
1. Select the project created in the previous step.
2. Specify NIM as the serving model. This can be done from either the **Overview** tab or from the **Models** tab. Click **Deploy model** under the NVIDIA NIM model serving platform tile.
3. On the new window, enter the details for the NVIDIA NIM model.
   - Select the NIM model, in this case we will use the Llama-3.1-8b-instruct-latest.
   - Enter the size of the persistent volume (PVC).
   - Select the number of NVIDIA GPU accelerator cards, based on the capacity of your cluster.
   - Select the model serving size and number of replicas.



4. After the model is deployed, you can check the status.



5. Take note of the internal endpoint URL, this will be used in next steps.



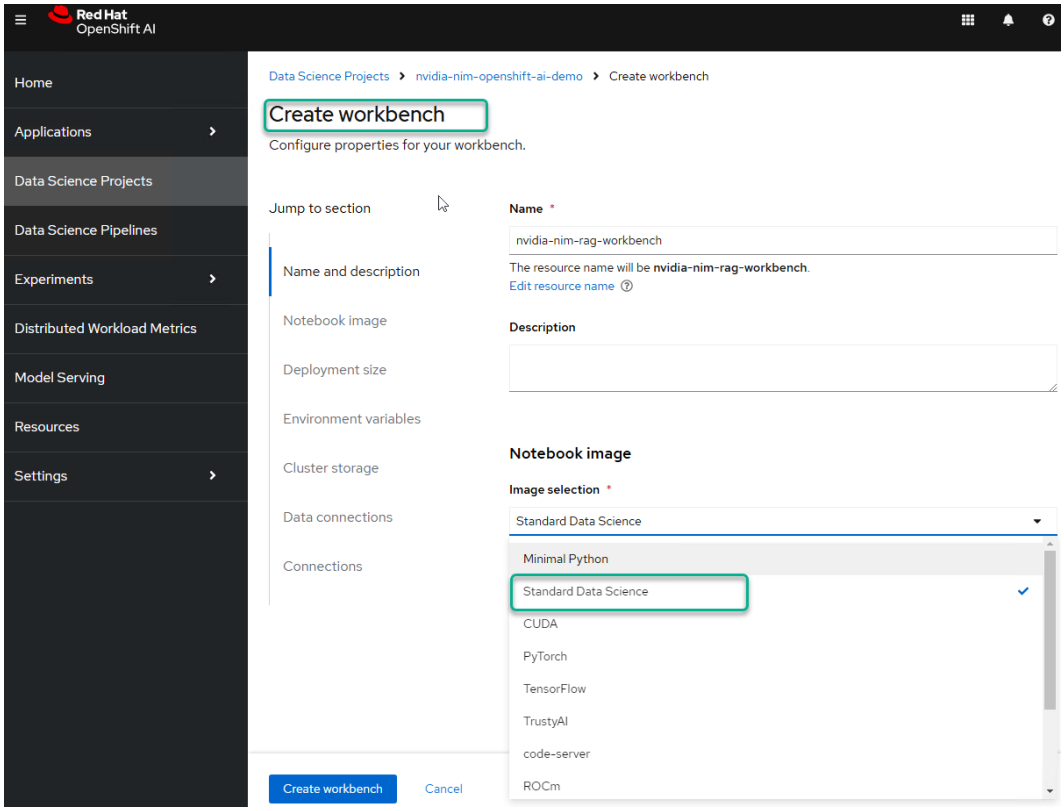
Configure and create a Workbench

After the NVIDIA NIM serving model is deployed and we have the endpoint URL, the next step is to create a workbench.

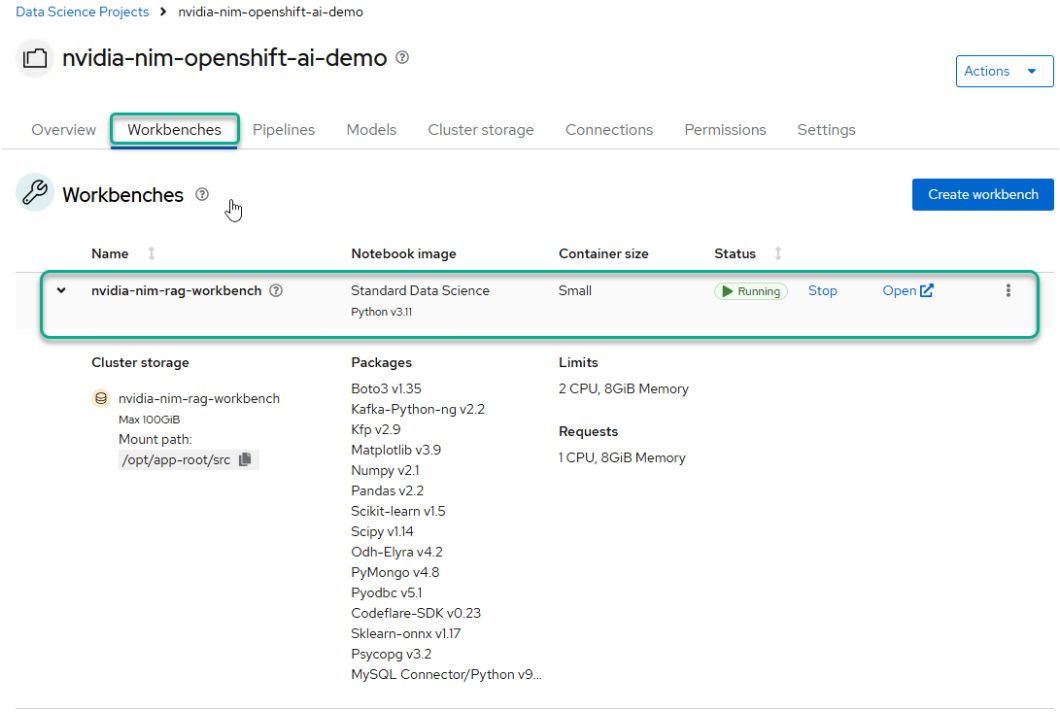
A workbench is an instance of your development and experimentation environment. They typically contain IDEs, such as JupyterLab, RStudio, and Visual Studio Code. When deploying a workbench you can select a notebook image, such as PyTorch, Standard Data Science, TensorFlow, etc., depending on the data science work that is required.

Procedure

1. From the **Workbenches** tab, click **Create workbench**, and then enter a name and description.
2. Select one of the existing notebook images. In this case we select “Standard Data Science” and version 2024.2.



3. Select the container image and accelerator. In this case we are using a Small container size and NVIDIA GPU.
4. Enter the size of the Persistent Volume. The PVC will be created on the storage being used by the default Storage Class. For Persistent storage we are using Hitachi VSP storage and Hitachi Storage Plugin for Containers (HSPC).
5. Click **Create workbench** and wait for the workbench to be in a running state.

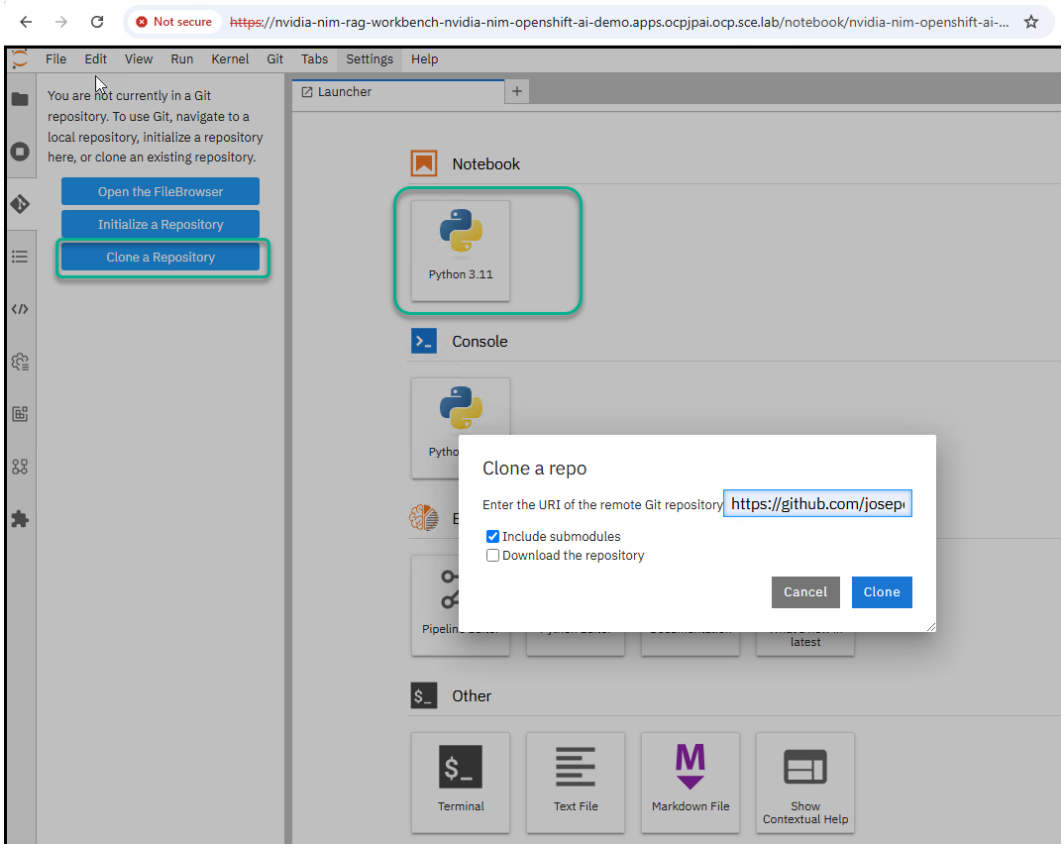


Create a notebook and verify access to the NVIDIA NIM model

The next step after creating the workbench is to deploy a notebook and work with an AI model.

Procedure

1. To access the workbench interface, click the **Open** link. Log in with the OpenShift credentials.
2. The first window that opens after opening the workbench is a Jupyter environment.



3. In this window you can create a notebook from scratch, or you use git to clone an existing repo. In this case, we are cloning an existing repo which contains a notebook that is ready to use.

   This is an example of how to invoke NVIDIA NIM running internally in a serving model inside the same project as the workbench.

   Make sure to use the URL as shown on the NVIDIA serving model end-point. Change the value of the `nim_url` parameter accordingly.

```
import json
import requests
import urllib3
urllib3.disable_warnings()
nim_url = "https://nvidia-nim-model-llama3-demo.nvidia-nim-openshift-ai-
demo.svc.cluster.local"
```

   This cell is to query the list of available models. Keep note of this model "`meta/llama-3.1-8b-instruct`" because it will be used in the next step.

```
response = requests.get(nim_url + "/v1/models", verify=False)
print(json.dumps(response.json()['data'], indent=2))
```

   Example of the response:

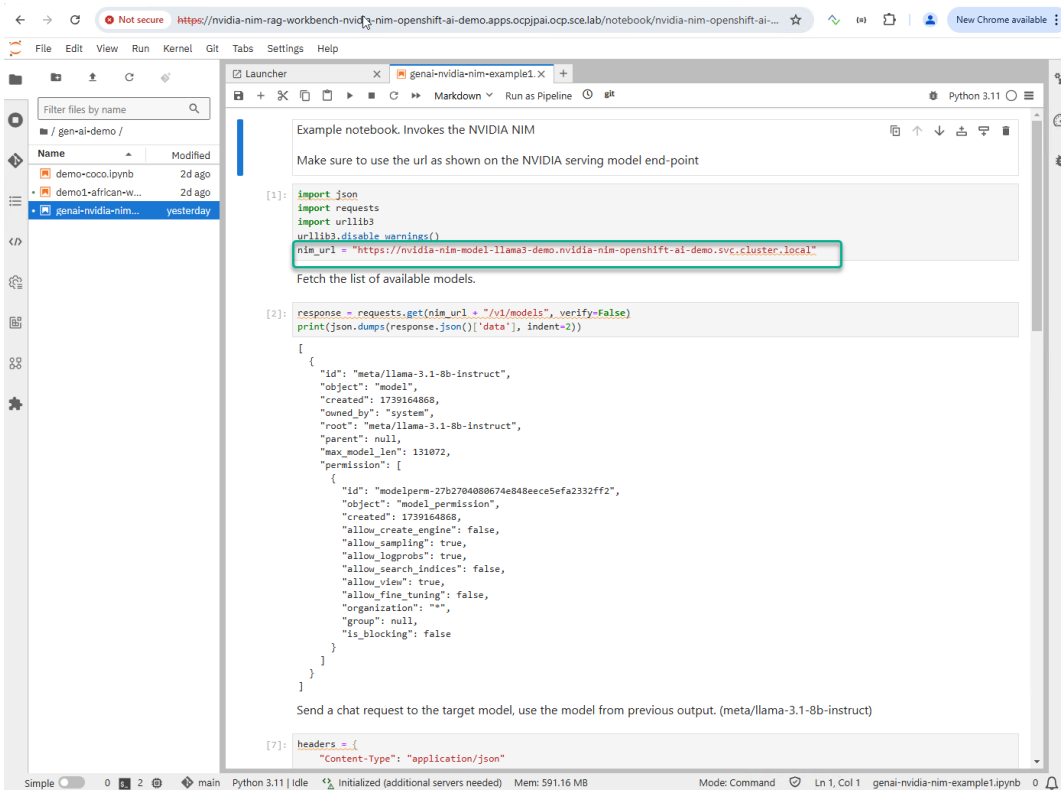
```
[
  {
```

```
    "id": "meta/llama-3.1-8b-instruct",
    "object": "model",
    "created": 1739164868,
    "owned_by": "system",
    "root": "meta/llama-3.1-8b-instruct",
    "parent": null,
    "max_model_len": 131072,
    "permission": [
      {
        "id": "modelperm-27b2704080674e848eece5efa2332ff2",
        "object": "model_permission",
        "created": 1739164868,
        "allow_create_engine": false,
        "allow_sampling": true,
        "allow_logprobs": true,
        "allow_search_indices": false,
        "allow_view": true,
        "allow_fine_tuning": false,
        "organization": "*",
        "group": null,
        "is_blocking": false
      }
    ]
  }
]
```

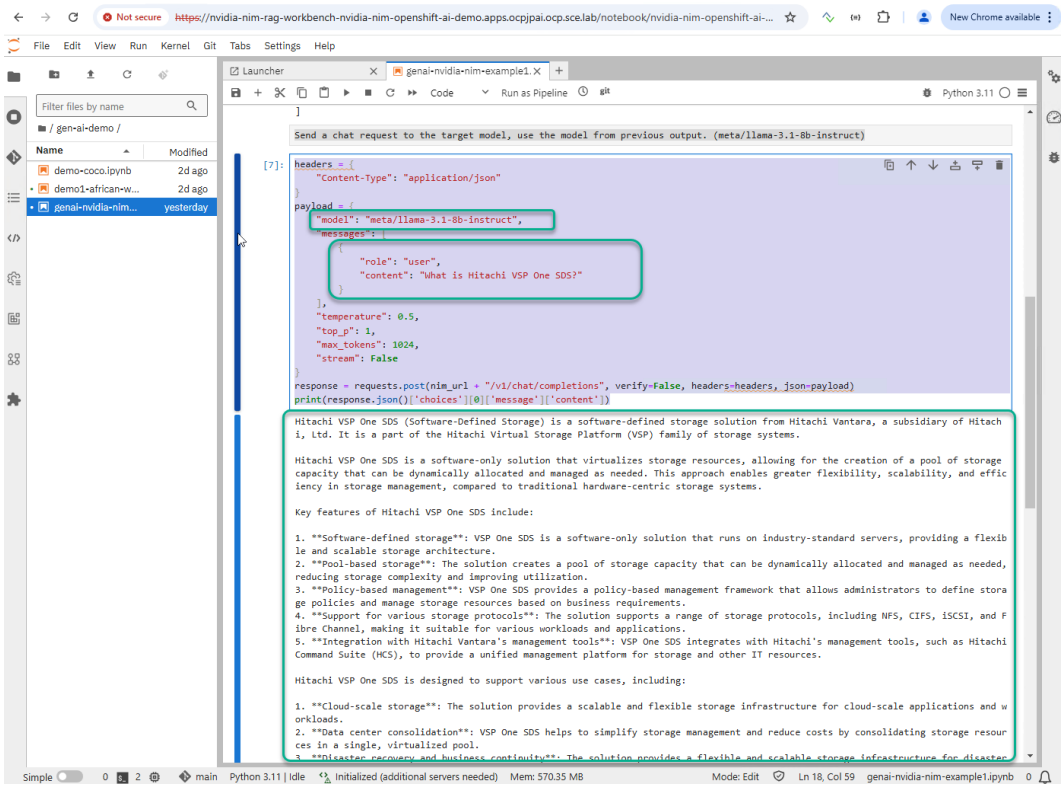

The following cell contains the code required to send a chat request to the target NIM model. Make sure to use the model from previous output "`meta/llama-3.1-8b-instruct`". In this case we are asking a question about Hitachi VSP One SDS storage.

```
headers = {
    "Content-Type": "application/json"
}
payload = {
    "model": "meta/llama-3.1-8b-instruct",
    "messages": [
        {
            "role": "user",
            "content": "What is Hitachi VSP One SDS?"
        }
    ],
    "temperature": 0.5,
    "top_p": 1,
    "max_tokens": 1024,
    "stream": False
}
response = requests.post(nim_url + "/v1/chat/completions", verify=False,
headers=headers, json=payload)
print(response.json()['choices'][0]['message']['content'])
```

The following screenshot shows the code snippets inside the notebook.



Here we can see the response:



As we can see from this test case, it is very easy to setup an NVIDIA NIM serving model and access it using Red Hat OpenShift AI. This integration allows development teams to

rapidly build and scale modern enterprise applications leveraging a set of easy-to-use inference microservices included as part of the NVIDIA AI Enterprise software platform.

Model training from a pretrained model using OpenShift AI

For this test case, we are going to show the process to re-train a model using a pretrained YOLO model (Ultralytics YOLO11 is a real-time object detection model) and a dataset of football players, export and reuse the trained model. The original dataset was obtained from RoboFlow and it has annotated images, split into test, train, and validation datasets.

The test case consists of the following:

- Retrain the model using the imported dataset.
- Verify that the retrained model can do object detection.
- Export the model to ONNX format (Open Neural Network Exchange).
- Upload the model to an S3 bucket to be reused in subsequent tests.
- Download and test the retrained model in a separate environment.

Create a Data Science Project

Procedure

1. From the OpenShift AI dashboard, navigate to the **Data Science Projects** tab.
2. Click **Create Project**, and enter a name and description for the project.
3. Click **Create**.

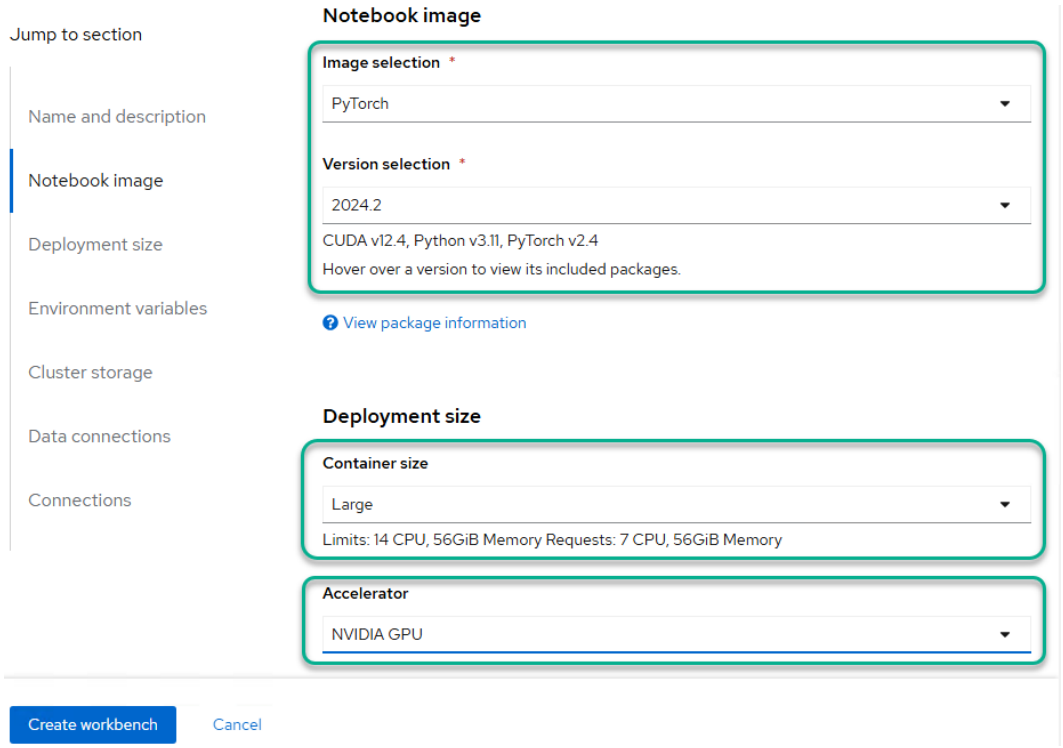
   Inside the created project is where all the additional resources for this test case will be deployed.

Configure and create a Workbench

The next step is to create a workbench which is an instance of development and experimentation environment. Within a workbench we can select a notebook image for your data science work.

Procedure

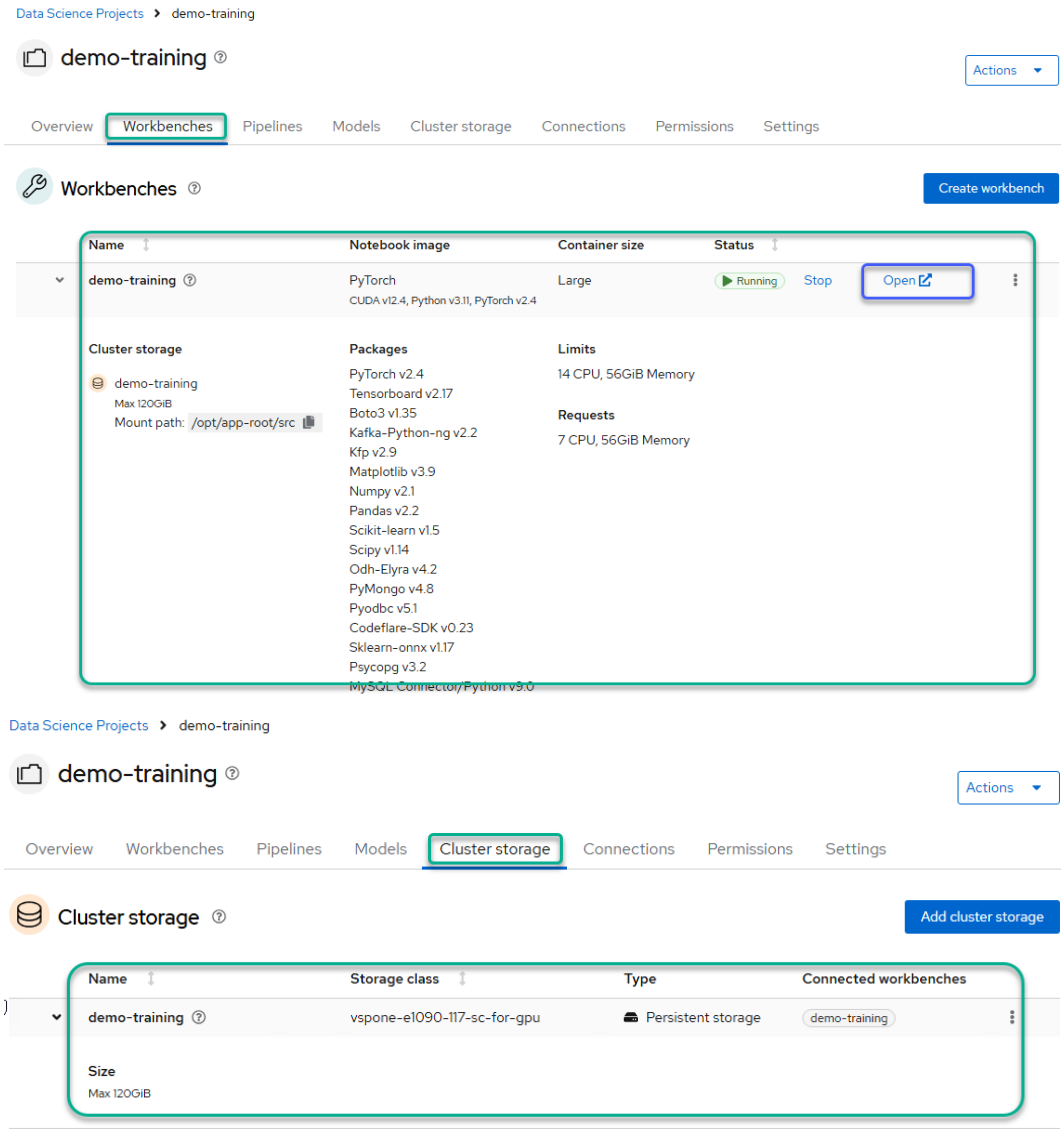
1. Select the newly created project, then click the Workbenches tab.
2. Click **Create workbench**, and then enter a name and description.
3. Select one of the existing notebook images. In this case we select “PyTorch” and version 2024.2.
4. Select the container image and accelerator. In this case we are using a Large container size and NVIDIA GPU.



5. On the cluster storage enter the size of the persistent volume to be created. The PVC will be created on the storage being used by the default Storage Class. For Persistent storage we are using Hitachi VSP storage and Hitachi Storage Plugin for Containers (HSPC).
6. (Optional) In addition to the cluster storage, add a connection to S3 storage. This can be used later to upload or download files from an S3 bucket.

   Note: A data connection is used to store configuration parameters that are required to connect to a data source, such as an S3 object bucket. This can be used for storing your models and data. You can reuse this bucket and its connection for your notebooks and model servers.

   To create data connections to your existing S3-compatible storage buckets, you need the following credential information for the storage buckets: Endpoint URL, Access key, Secret key, Region, Bucket name.
7. It can take a couple of minutes to have the workbench in a running state, and then you can access the workbench interface by clicking the **Open** link. Log in with the OpenShift credentials.

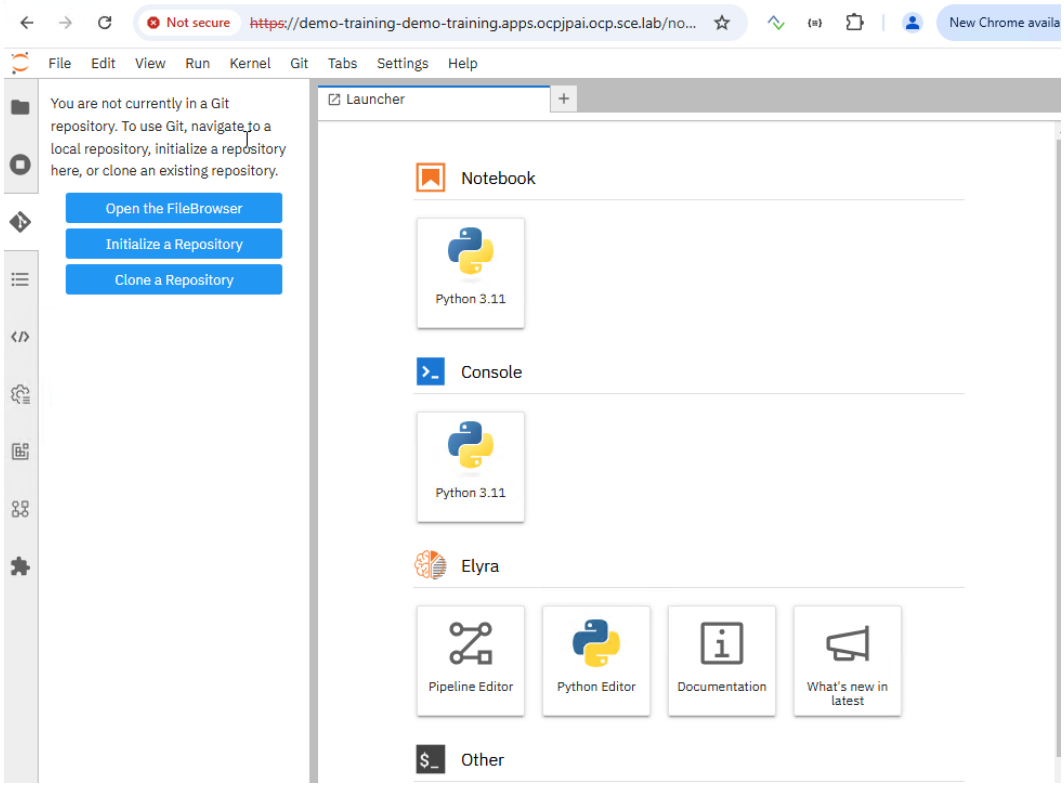


Create a notebook to retrain a model

The next step after creating the workbench is to deploy a notebook and work with an AI model. In this section we are going to show some of these steps:

Procedure

1. The first window that opens after opening the workbench is a Jupyter environment similar to the following.



2. While we can always write a notebook from scratch, the easiest way is to import a Git repository. In this case, we are cloning the same repository used in the previous use case.
3. After the repo has been cloned, and before running the notebook to retrain a model, the first step is to import the dataset.
4. Import dataset:

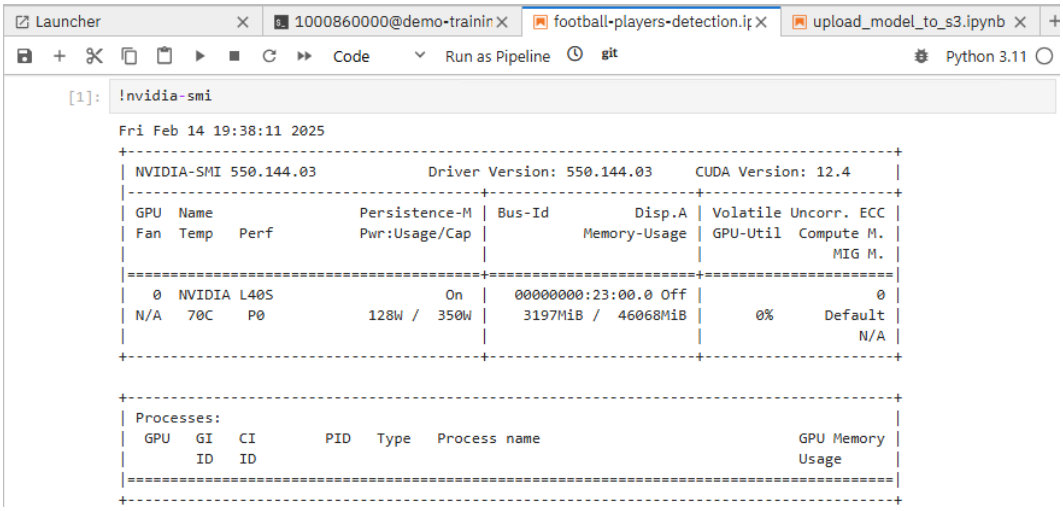
   From the cloned repo, navigate to notebooks folder and open the “download_extract_dataset” notebook. Make sure the update the URL based on the location of the dataset. In this case we are using a dataset obtained from RoboFlow.

   Run the notebook and wait until the dataset is downloaded and extracted to the corresponding folder.
5. Retrain the model:

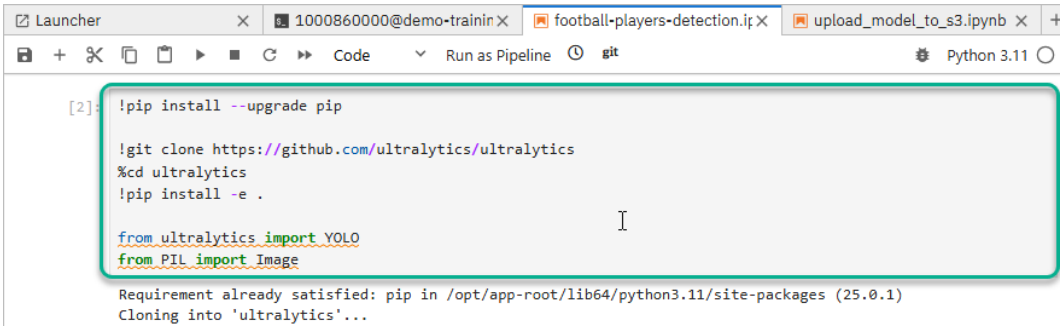
   Now it is time to retrain the model. Open the “football-players-detection” notebook.

   This notebook example shows how to retrain a pre-trained YOLO model using a dataset obtained from Roboflow. Also, a set of unseen images has been included to test the retrained model.

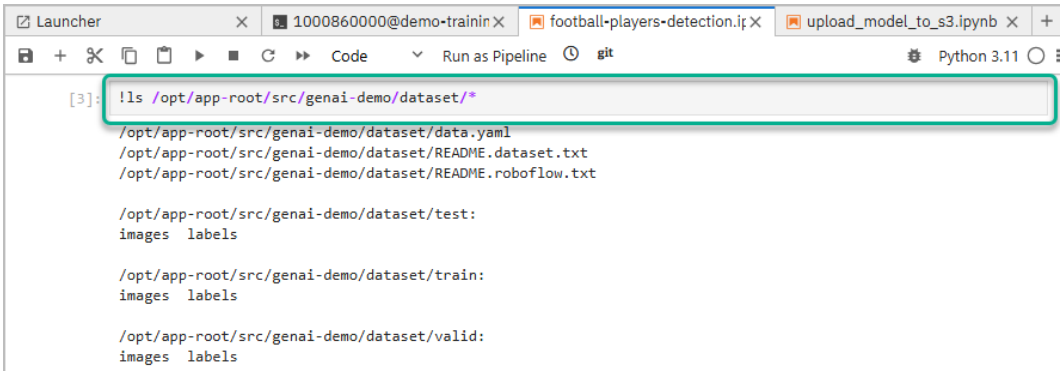
   The first cell is to verify that we can access the NVIDIA GPU.



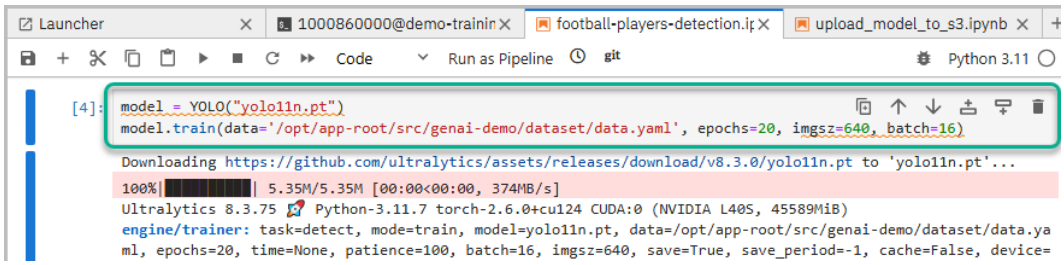
The next step is to clone the Ultralytics GitHub repository, install, and import YOLO.



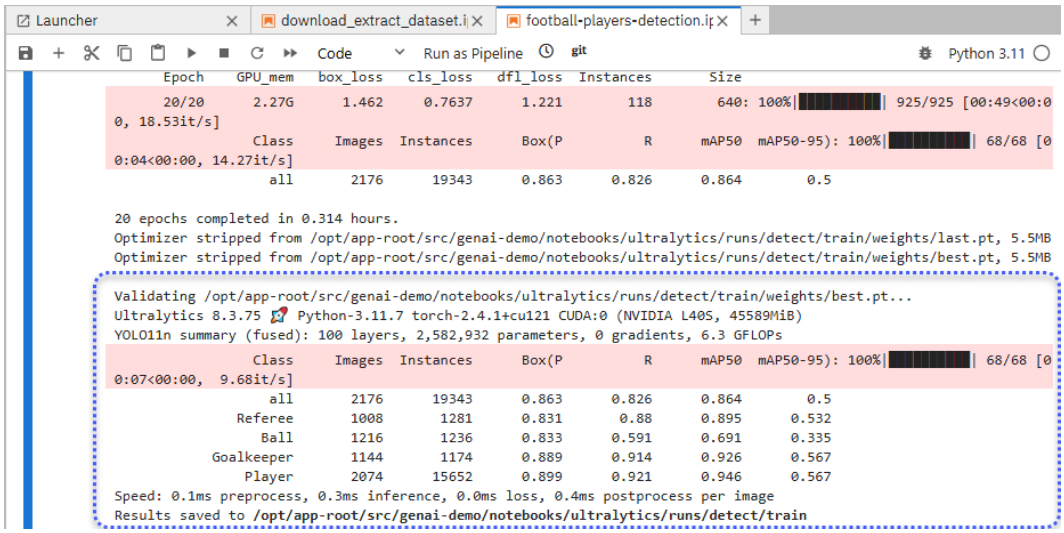
After the required packages have been installed, the next cell is to verify the dataset. The output shows the folder for the dataset, including the test, training, and validation subfolders.



Here we start retraining the model using the imported dataset. As seen below, the first step is to load a pretrained model, in this case are using YOLO11n which is the smallest YOLO11 models. Next, we call the “train” function, change the parameters according to your test. For example, you can adjust the parameter “epoch” depending how many iterations you want to tune during the training.



From the output we can see which GPU is being used, in this case we are using NVIDIA L40 GPU. At the completion of the training, the weights will automatically be saved in the "`runs/detect/train`" folder.



After we have the re-trained model we can start making predictions with new images. For this purpose, we have included a set of unseen images on the test folder outside of the dataset.

Because we retrained this model with the football players dataset, the new re-trained model should be able to detect players, goalkeepers, and even referees on an image of a football (or soccer) field.

In this case we are testing with a sample image "football_image1.jpg". Modify the path if needed.

Here we are loading the updated weights file to the model and passing a sample image, and as we can see the model is able to predict the images correctly, detecting goalkeeper, players, and ball.

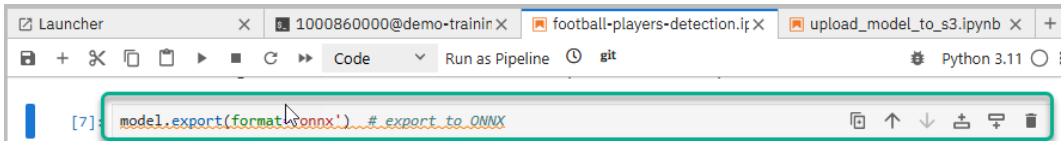
Export the model and upload to S3 storage

Now that we have verified that the retrained model works as expected, we can export and save the model in ONNX format so it can be used for inferencing in another container image on prem or in cloud. ONNX stands for Open Neural Network Exchange which is an open standard for representing machine learning models.

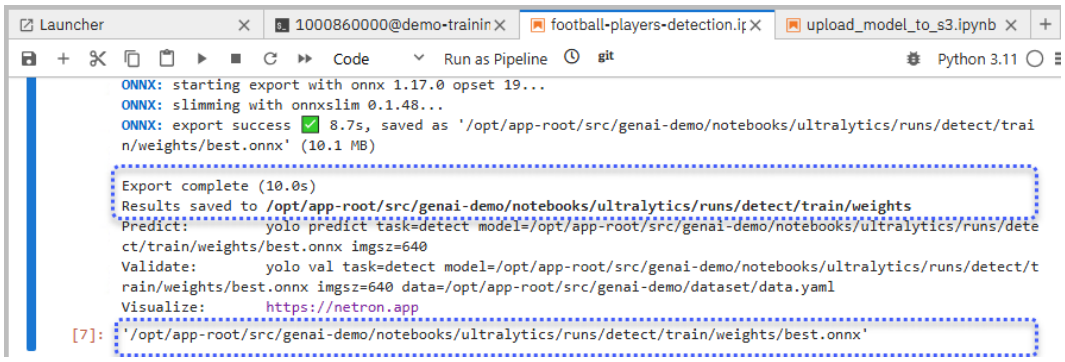
Procedure

1. Export and save the model.

   To export the model in ONNX format we use the following command:

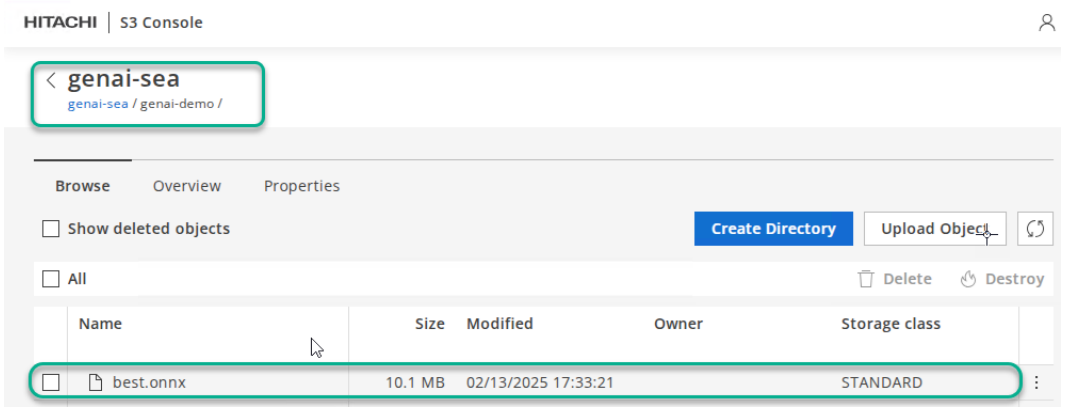


   The following output shows the path where the ONNX file is stored. This file can easily be copied to another location so it can be used in another container.



2. Upload the model to Hitachi VSP One Object (S3 storage).

   Inside the cloned git repository is a notebook file that can be used to upload files to an S3 storage. We are using it to upload to Hitachi S3 storage so it can be used in the next use case in a different workbench.

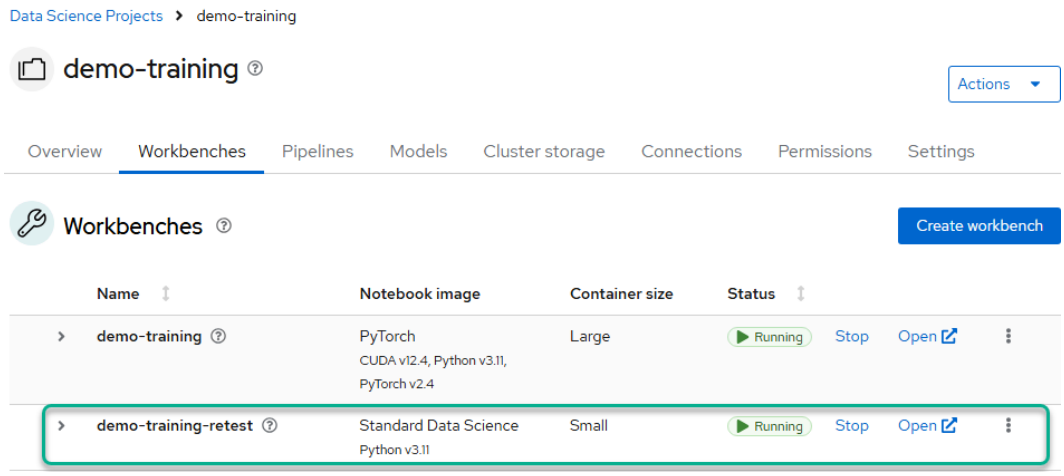


Testing a retrained model

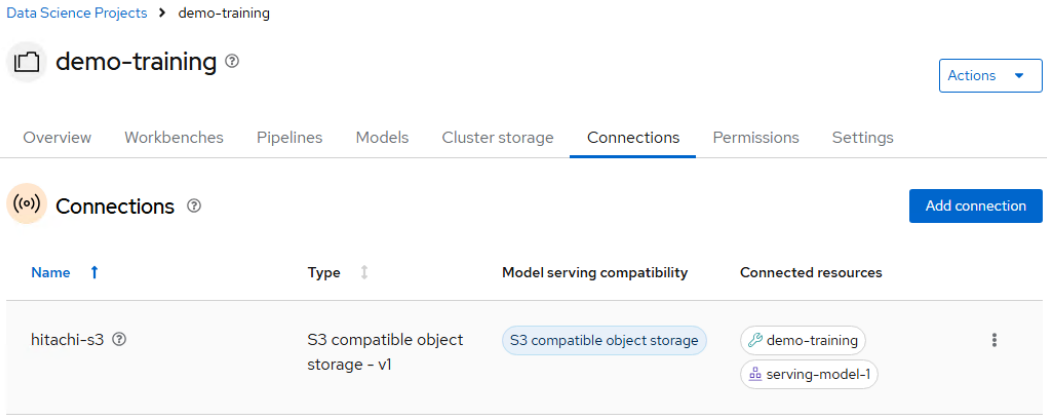
Now that we have retrained, exported the ONNX model, and placed a copy within an S3 bucket we are going to test the retrained model against another set of unseen images using a new workbench/container.

Procedure

1. Create a new workbench inside the same project. The project is called “demo-training-retest”, and we have chosen the notebook image “Standard Data Science” and no GPU.



2. To verify the S3 connections, click the **Connections** tab and you will see that the S3 connections are used by both workbenches.

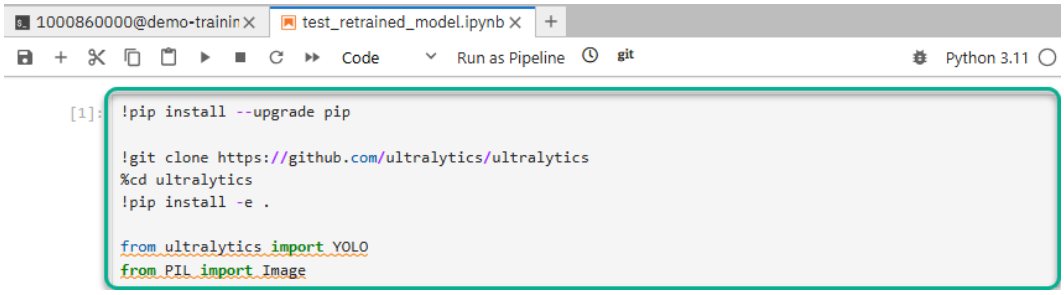


3. Click **Open** and it will redirect to a new window with a Jupyter environment.
4. Use Git to clone the same repository.

   To test the retrained model, use the notebook “test_retrained_model” from the cloned repository.

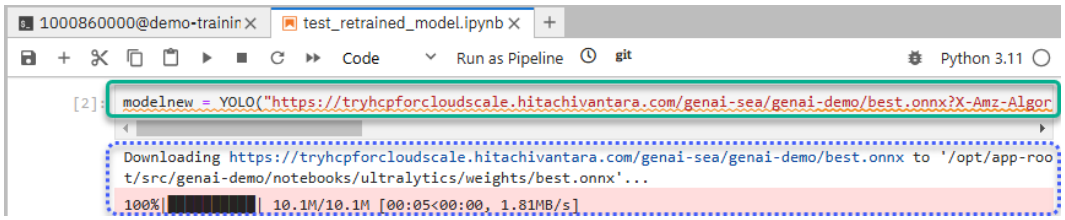
   Here are some details of the cells of this notebook:

   The first cell is to install the required packages.

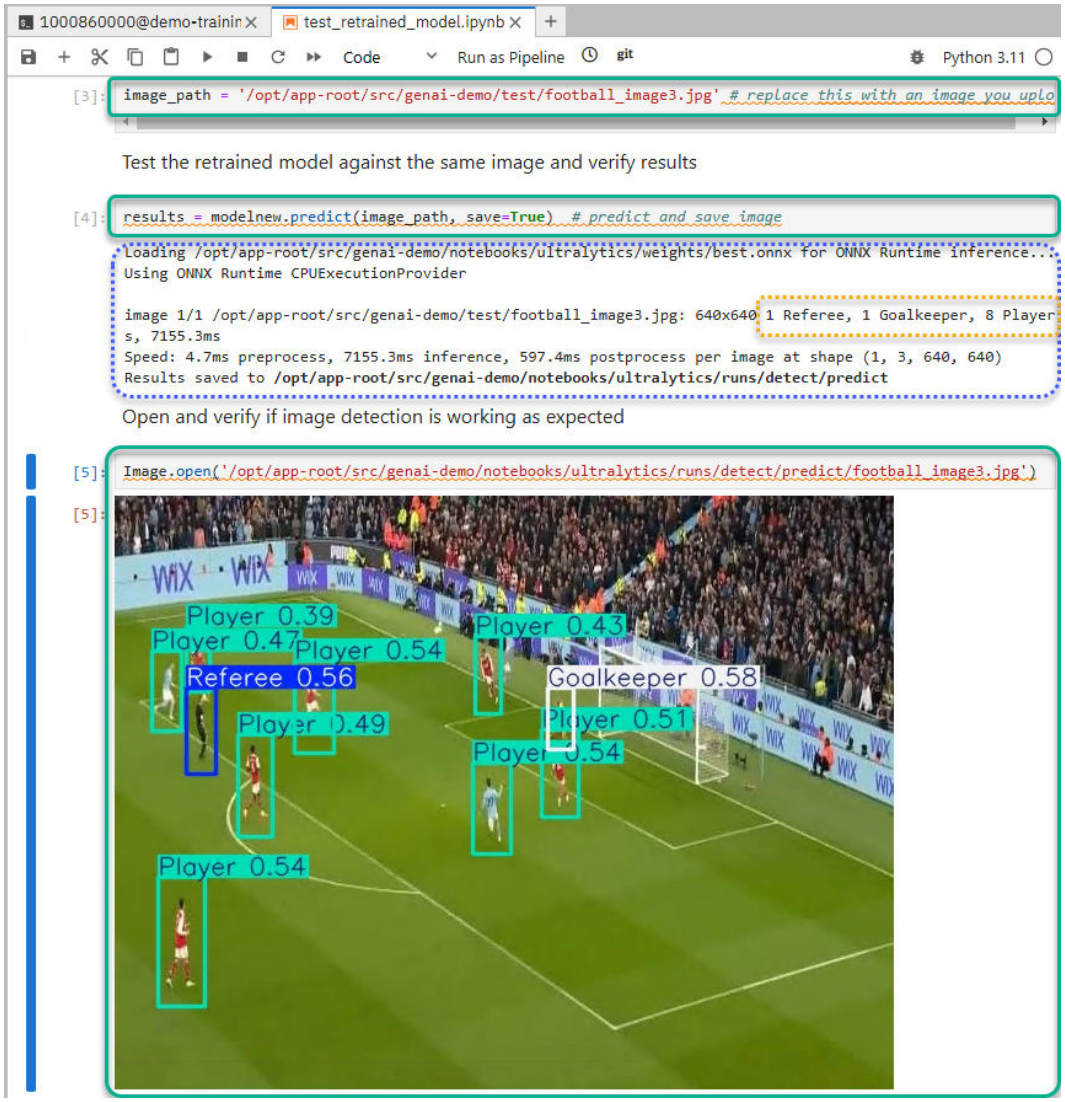


   Next, we get the retrained model, exported in the ONNX format, from the S3 bucket.

   In the output below we can see the ONNX model is being downloaded and stored locally.



Now that we have the model, we are going to test against a different and unseen image.



The result shows that the retrained model obtained from the S3 bucket can do image detection and works as expected, detecting, and differentiating between players, goalkeeper, and referee.

Conclusion

This reference architecture also provides a reference pattern for an AI-enabled platform using the Hitachi Integrated Systems platform with Red Hat AI, OpenShift AI, and NVIDIA GPUs leveraging VSP One Storage Platforms and Hitachi's NVIDIA certified data center HA compute servers. It leverages the latest capabilities and services provided by Red Hat including OpenShift AI Operators, InstructLab, Granite, and NVIDIA NIM microservices

RHEL AI, along with Red Hat OpenShift AI, underpins Red Hat AI, Red Hat's portfolio of solutions that accelerate time to market and reduce the operational cost of delivering AI solutions across the hybrid cloud.

Red Hat OpenShift AI enhances RHEL AI with Kubernetes-based containerized orchestration and scalability. It is a platform that empowers enterprises to develop and deploy AI-driven solutions efficiently across hybrid cloud environments, facilitating the creation and delivery of AI-enabled applications at scale with high reliability and security. Additionally, it supports model serving and hosting, making real-time AI inference and integration seamless.

A key benefit we concluded with this reference pattern is that it helps to accelerate the time to value of AI projects. It significantly eliminates the time to find and deploy the right combination of software packages suitable to run AI/ML workloads and minimizes the expertise required to have an environment up and running to start prototyping/deploying AI/ML applications. The combination of Hitachi automated integrated system with Red Hat OpenShift Software stack will help data scientists and data engineers eliminate spending too much time on the infrastructure/DevOps processes and instead use that time to focus on developing/training, fine-tuning their models and deploying/integrating those models into workloads/applications in production environments.

With this enterprise-grade container orchestration platform, IT and DevOps teams have a set of easy-to-use tools for accelerating the deployment of AI/ML models on-premises and/or the public cloud. Organizations can self-host large language models (LMMs) in their own managed environments allowing developers to quickly build enterprise-grade AI applications with minimum effort.

Some of the key benefits of this platform include:

- Flexible and high-performance environment
- Simplified deployment, reducing the time required to retrain, test, and validate AI models
- Capabilities to manage the entire lifecycle of AI/ML projects

Overall, this solution provides a foundational platform for easy deployment and scalability of AI/ML workloads.

Product descriptions

This section provides information about the hardware and software components used in this solution.

Hitachi Integrated Systems Platform

The Hitachi Integrated Systems platform is a high-performance, low-latency, integrated, converged solution using Hitachi Virtual Storage Platform One Block storage, Hitachi Advanced Server HA820 G3, as well as HA810 G3 with Sapphire Rapids Scalable Processors.

Hitachi Virtual Storage Platform One Block

The Hitachi Virtual Storage Platform One Block series simplifies system setup and management through Hitachi Clear Sight and VSP One Block Administrator. Dynamic Drive Protection reduces RAID complexity, and always-on compression and deduplication enhance simplicity. Virtual Storage Platform One Block with QLC (quad-level cell) delivers the industry's most reliable high density, cost optimized All-Flash Array for read-intensive workloads. It is a compelling infrastructure option for IT organizations that support hybrid cloud environments looking to balance performance, capacity and cost.

Dynamic Carbon Reduction optimizes energy usage by switching CPUs to ECO mode during low activity. Adaptive Data Reduction (ADR) is always on, enhancing efficiency and reducing the overall CO2 footprint.

Thin Image Advanced (TIA) integrates with major snapshot ecosystems, prioritizing security by defending against threats and ensuring data confidentiality. CyberArk Privileged Access Manager plugins enhance block storage system security by prioritizing data confidentiality, ensuring compliance, and actively defending against security threats.

Hitachi Virtual Storage Platform One Block 20 includes 3 dedicated models that support both TLC and QLC NVMe SSD drives, as well as Fibre Channel, iSCSI, and NVMe TCP connectivity. The new capabilities remove complexity: data reduction is always on, Dynamic Drive Protection removes complicated RAID setup, and Dynamic Carbon Reduction delivers real world reduction in power consumption. In addition, the models are FIPS compliant.

- VSP One Block 24 – 256 GB Cache + SW Advanced Data Reduction (ADR) + 24 cores
- VSP One Block 26 – 768GB Cache + 2 × Compression Accelerator Module (CAM) + 24 cores
- VSP One Block 28 – 1TB Cache + 4 × CAM + 64 cores

In short, the Hitachi Virtual Storage Platform One Block series combines simplicity, sustainability, and robust security features to optimize system management, energy efficiency, and data protection.

Hitachi Virtual Storage Platform One Object

VSP One Object offers massive scale, cloud capabilities, and broad protocol support. It supports all-flash configurations and policy-driven tiered storage, allowing performance and capacity to scale independently. It quickly ingests large volumes of small objects and cost-effectively stores very large objects.

Customers benefit from a diverse array of configurations and deployment options. These include traditional archive and backup solutions, as well as modern cloud-native environments, AI/ML applications, and analytics workloads. The platform is optimized for a compact form factor, fitting an entire cluster into a 4U footprint. This enables real-time analytics at the edge, eliminating the need to transfer data to a central location for analysis.

Hitachi Advanced Server

Designed to unlock the full benefits of the hybrid cloud, Hitachi Advanced Server models deliver high performance and enhanced security while reducing operational costs. Global enterprises, cloud service providers, and governments trust Hitachi servers to run bare metal, virtualized, or containerized applications. Powered by industry-leading scalable processors, Hitachi servers are ideal to deliver edge, core, and cloud IT services.

Hitachi Advanced Server supports a variety of GPUs for parallel processing which is used in a wide range of applications, including AI, graphics, and video rendering.

Hitachi servers are designed and optimized to maximize performance for VMware, Red Hat, Oracle, bare metal, virtual desktop infrastructure (VDI), SAP, analytics, and other enterprise workloads.

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA

HitachiVantara.com/contact