



NetApp Verified Architecture

NetApp AI Pod™ with Lenovo and Red Hat OpenShift AI for MLOps

NVA deployment

Abhinav Singh, NetApp

June 2025 | NVA-1179-DEPLOY

In partnership with



Abstract

This verified architecture discusses how the integration of Lenovo ThinkSystem server with NVIDIA L40S GPUs, NetApp storage solutions, and NVIDIA networking infrastructure offers a powerful and scalable platform for machine learning operations (MLOps). This combination ensures high-performance computing, efficient and intelligent data management. Additionally, Red Hat OpenShift AI enhances the orchestration and management of AI applications, streamlining the development, deployment, and monitoring processes in enterprise environments.

TABLE OF CONTENTS

Executive summary	4
Program summary	4
Advantages of Running AI Workloads on NetApp	4
Advantages of Running AI Workloads on Lenovo ThinkSystem Servers	5
Solution overview	5
Target audience.....	5
Technology requirements	6
Hardware requirements	6
Software requirements	6
Concepts and Components	6
Lenovo ThinkSystem SR675 V3 Server	6
Lenovo XClarity Systems Management.....	7
Networking.....	8
NetApp ONTAP 9	8
NetApp ONTAP S3.....	9
NetApp AFF A-Series	10
NetApp Trident	10
Red Hat OpenShift	11
Red Hat OpenShift AI	12
NVIDIA AI Enterprise (NVAIE).....	12
NVIDIA AI Blueprint.....	13
Solution Design.....	13
Single node deployment	13
Scaling within Single Node OpenShift	14
Scaling OpenShift with additional compute node.....	14
Solution Deployment	16
Network configuration	16
Lenovo ThinkSystem server configuration.....	19
NetApp AFF A30 configuration	21
Single Node OpenShift configuration on Bare Metal	33
Install NetApp Trident	47
Installing NVIDIA GPU Operator.....	51
Adding Administrative User to the OpenShift.	55

Installing Red Hat OpenShift AI Self-Managed	56
OpenShift AI – Basic Configuration	63
Solution Validation	66
AI/ML pipeline for Edge	66
RAG with OpenShift AI and Elasticsearch	72
NVIDIA NIM on OpenShift AI	77
Edge to Core to Cloud & NetApp Integration	82
Conclusion	82
Acknowledgments	83
Bill of Materials	83
Version history	85

LIST OF TABLES

Table 1) Hardware requirements	6
Table 2) Software requirements	6
Table 3) VLAN information	16
Table 4) OpenShift parameters	33

LIST OF FIGURES

Figure 1) Lenovo ThinkSystem SR675 V3 configured to support eight double-wide GPUs	7
Figure 2) SN3700 - 32 ports of 200GbE in a compact 1U form factor	8
Figure 3) ONTAP S3 – Logical diagram	10
Figure 4) NetApp Trident deployed on a High Availability Kubernetes Cluster	11
Figure 5) Red Hat OpenShift AI architecture and components	12
Figure 6) NVAIE and Red Hat OpenShift Integration	13
Figure 7) Solution Topology (Single Node)	14
Figure 8) Solution topology with additional compute node	15
Figure 9) Lenovo XClarity Provisioning Manager	20
Figure 10 RAID configuration for M.2 NVMe drives	21
Figure 11) Edge to Core/Cloud Pipeline	82

Executive summary

This document covers a verified architecture optimized for Artificial Intelligence (AI) and ML workloads comprised of Lenovo ThinkSystem servers, NetApp AFF storage, NVIDIA networking, NVIDIA AI Enterprise software stack, Red Hat OpenShift and Red Hat OpenShift AI. The solution covers all aspects of setup, configuration and best practices for NetApp AI Pod with Lenovo and NVIDIA systems to perform Machine Learning Operations (MLOps). It provides orderable PNs and gives scaling guidance to customers and partners to make the correct choice of NetApp storage for the solution.

As organizations look to implement AI/ML as production applications, they face challenges with workload scalability, data availability and the agility required for delivery. The challenge is due to unique requirements of each application based on the specific use case. The deployment of these ML applications in production necessitates extensive collaboration and integration of tools, processes, and workflows between established teams and newer ML and data teams. Post-deployment, these applications, along with their models and data pipelines, will require ongoing improvement and maintenance. Application teams must incorporate model delivery and data pipelines into their existing application delivery framework, which may encompass continuous integration and continuous delivery (CI/CD), automation, and other DevOps best practices. Red Hat OpenShift and Red Hat OpenShift AI will help address these challenges by adopting MLOps.

Combine the proven capabilities of Red Hat OpenShift AI and Red Hat OpenShift in a single enterprise-ready AI application platform that brings teams together. Data scientists, engineers, and app developers can collaborate in a single destination that promotes consistency, security, and scalability. The latest release of OpenShift AI includes a curated collection of optimized, production-ready, third-party models, validated for Red Hat OpenShift AI. Access to this third-party model catalog gives your team more control over model accessibility and visibility to help meet security and policy requirements.

Additionally, OpenShift AI helps manage costs of inferencing with distributed serving through an optimized [Virtual Large Language Model](#) (vLLM) framework. To further reduce operational complexity, it offers advanced tooling to automate deployments and self-service access to models, tools, and resources.

Program summary

Lenovo servers combined with NVIDIA networking and NetApp storage provide key benefits to customers such as:

- Flexible reference architecture for Generative AI workloads.
- Optimized infrastructure for high-performance and scalability of compute and storage
- Solution guidance in terms of Compute, Network and Storage.
- Reliability and security– Data protection, built-in ransomware protection, etc.
- Enhanced automated system management from XClarity for simple, faster deployment and configuration as well as updated security features to detect and protect from unauthorized access

Advantages of Running AI Workloads on NetApp

The objective of this solution is to help enterprises to be successful in their AI journey, with an architecture that provides a powerful data management plane which provides the following benefits:

- Velocity – Handling large datasets at high velocity for versioning
- Disaggregation – Ability to scale performance and capacity independently
- Reliability – Data protection, built-in ransomware protection, and dynamic provisioning of storage

- Cloud Adjacency – Connect with resources across multiple data centers and clouds to load GenAI knowledgebase repositories
- Efficiency – Enterprise storage features such as compression and deduplication on data sets
- Secure Multi-Tenancy – Adaptive QoS to isolate multiple AI workloads along with encryption for data at-rest and in-flight and using multiple SVMs
- Automated Ransomware Protection

Advantages of Running AI Workloads on Lenovo ThinkSystem Servers

Lenovo ThinkSystem Servers are designed to deliver optimal performance and energy efficiency for Artificial Intelligence (AI), High Performance Computing (HPC) and graphical workloads across an array of industries.

- Award winning performance and reliability - Lenovo x86 servers had the best uptime among all x86 platforms for the 10th straight year ([ITIC Global Server Hardware, Server OS Reliability Report](#)).
- Unique thermal and power efficiency - Lenovo Neptune solutions allow customers to optimize performance, density, and energy consumption for challenging extensive data set modeling and simulation or AI workloads.
- Simple Management - XClarity Controller2 (XCC2) provides advanced service-processor control, monitoring, and alerting functions. The XCC2 consolidates the service processor functionality, super I/O, video controller, and remote presence capabilities into a single chip on the server system board.
- Built-in security - The ThinkSystem SR675 V3 includes Platform Firmware Resiliency (PFR) hardware Root of Trust (RoT) which enables the system to be NIST SP800-193 compliant, further enhancing key platform subsystem protections against unauthorized firmware updates and corruption, to restore firmware to an integral state, and to closely monitor firmware for possible compromise from cyber-attacks.
- Lenovo AI-POD is built upon a comprehensive support model. Through solution-level interoperability testing Lenovo and NetApp can give customers confidence in a supported environment based on proven best practices while still tailoring it exactly to the customer's needs. That means that the infrastructure is not just supported on a component break and fix or "box"-level, but with a holistic perspective including software, firmware and even firmware-settings.
- Lenovo Services and NVIDIA Services provide the necessary experience to bring your Generative AI environment to life.

Solution overview

This engineering solution offers the following benefits:

- Run MLOps on a verified platform with NetApp storage, Lenovo servers and Red Hat OpenShift AI.
- Predictable facilitation and effective deployment and management of full-stack solutions for AI by testing and documenting end-to-end infrastructure
- Highly available and scalable platform to create repeatable building blocks
- Single storage platform for both containerized applications and the object store for MLOps pipelines.

Target audience

This document is designed for Data Scientists, Solution Architects, Engineers, IT Operations, and Field Consultants aiming to leverage an infrastructure tailored for AI workloads. Prior knowledge of AI, cloud-

native workloads, networking, storage, and their components will be beneficial during the implementation phase.

Technology requirements

This section covers the hardware and software which are needed for this solution.

Hardware requirements

Table 1 lists all the hardware used for this solution and a detailed bill of materials is available in the last section of this document.

Table 1) Hardware requirements.

Hardware	Quantity	Comment
Lenovo ThinkSystem SR675 V3	1	Single Node OpenShift
NVIDIA SN3700 switch	1	East/West (2 nodes topology) and North/South Traffic – Cumulus Linux 5.9.1
NetApp AFF A30	1	HA Pair
Lenovo ThinkSystem SR675 V3 (OPTIONAL)	1	Additional Compute node for training use case

Software requirements

Table 2 lists all the software stack which was used for this solution.

Table 2) Software requirements

Software	Version	Comment
Red Hat OpenShift	4.17	Single Node OpenShift
Kubernetes	v1.30.10	K8s version
Red Hat OpenShift AI	2.19	For MLOps
NFD Operator	4.17.0	For GPU node discovery
NVIDIA GPU Operator	v25.3.0	Manages GPU resources and drivers in a K8s cluster
NVIDIA Driver Version	570.124.06	NVIDIA GPU driver
NVIDIA CUDA Toolkit	12.8	Software for GPU-accelerated computing applications
Nvidia-Mellanox ConnectX-7	28.43.2026	Firmware for ConnectX adapter
Mellanox ConnectX-6	26.43.1014	OCP Ethernet Adapter
Cumulus	5.9.1	NVIDIA SN3700 switch
NetApp ONTAP®	9.16.1P3	Storage operating system on AFF A30
NetApp Trident™ software	25.02.1	Fully supported open-source storage orchestrator for containers and K8s distributions, including Red Hat OpenShift

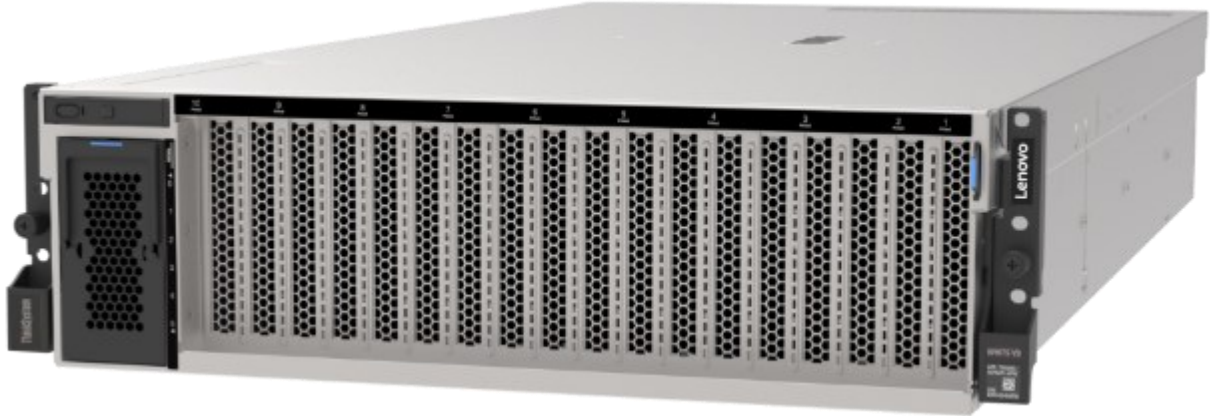
Concepts and Components

Lenovo ThinkSystem SR675 V3 Server

The Lenovo ThinkSystem SR675 V3 is a versatile GPU-rich 3U rack server, designed for the ultimate flexibility. The server can support up to eight double-wide or single-wide GPUs. The GPU options include

the NVIDIA Hopper, Lovelace, and Ampere datacenter portfolio and provide best-in-class cooling for the accelerators positioned in the front and allows both for front and rear IO connectivity for maximum graphic performance and IO throughput. The SR675 V3 is built on one or two 4th Generation AMD EPYC™ Processors, up to 24 TruDDR5 DIMMs and a choice of high-speed NVMe storage and networking. The configuration is built on Lenovo's Think System SR675 V3 which leverages PCIe gen 5 and the latest AMD EPYC processors.

Figure 1) Lenovo ThinkSystem SR675 V3 configured to support eight double-wide GPUs



Lenovo's SR675 V3 server (Figure 1) supports PCIe double-wide or single-wide GPUs that plays a pivotal role in enabling businesses to enhance foundational generative AI models. This server configuration provides enterprises with the capability to seamlessly tailor and deploy generative AI applications, incorporating cutting-edge features such as intuitive chatbots, advanced search systems, and efficient summarization tools.

The AI compute node in AIPod leverages the Lenovo SR675 V3 server, the minimum configuration is shown below:

- 2x AMD EPYC 9634 processors, 84C, 2.25GHz or 2x AMD EPYC 9354 processors, 32C, 3.25GHz or 2x AMD EPYC 9535 processors, 64C, 2.4GHz
- 4x 128GB TruDDR5 3DS RDIMMs
- 1x 10/25GbE dual-port OCP Adapter
- 2x M.2 NVMe Drives
- 2x NVIDIA L40S GPUs
- 1x NVIDIA ConnectX-7 200 GbE dual port adapter

Lenovo XClarity Systems Management

Lenovo XClarity provides fast, flexible, and scalable delivery of Lenovo infrastructure. XClarity integrates easily into Lenovo servers to automate provisioning and operations management, and into Lenovo switches and storage to automate operations management.

By seamlessly integrating with a wide range of external IT applications, you can effectively manage Lenovo infrastructure within your existing software tools' console, ensuring a cohesive and efficient workflow for IT operations.

The **NVIDIA ConnectX-7** family of Remote Direct Memory Access (RDMA) network adapters supports InfiniBand and Ethernet protocols and a range of speeds up to 400Gb/s enabling a wide range of smart,

scalable, and feature-rich networking solutions that address traditional enterprise needs as well as some of the most-demanding AI, scientific computing, and hyperscale cloud data center workloads

The **NVIDIA L40S GPU** is based on the Ada Lovelace architecture, delivers multi-workload acceleration for large language model (LLM) inference and training, graphics, and video applications. As the premier platform for multi-modal generative AI, the L40S GPU provides end-to-end acceleration for inference, training, graphics, and video workflows to power the next generation of AI-enabled audio, speech, 2D, video, and 3D applications. The Lenovo ThinkSystem SR675 V3 is highly compatible with several other powerful GPUs like NVIDIA H200 Tensor Core and NVIDIA H100 Tensor Core GPUs.

Networking

The NVIDIA® Spectrum SN3000 series switches are based on the 2nd generation of Spectrum switches, purpose-built for leaf/spine/super-spine data center applications. Allowing maximum flexibility, the SN3000 series provides port speeds spanning from 1GbE to 400GbE, and a port density that enables full rack connectivity to any server at any speed. In addition, the uplink ports allow a variety of blocking ratios to suit any application requirement.

SN3700 spine/super-spine offers 32 ports of 200GbE in a compact 1U form factor. It enables connectivity to endpoints at different speeds and carries a throughput of 6.4Tb/s, with a landmark 8.33Bpps processing capacity. As an ideal spine solution, the SN3700 allows maximum flexibility, with port speeds spanning from 10GbE to 200GbE per port.

For this solution a SN3700 (Figure 2) switch was used to connect to Lenovo SR675 V3 server and NetApp AFF A30 storage. The switch will carry NFS, S3 and IN-Band Management VLANs traffic. This switch also provides East-West (GPU) communication when an extra compute node with 4 x L40S GPUs will be added to Single Node OpenShift (SNO).

Figure 2) SN3700 - 32 ports of 200GbE in a compact 1U form factor



Management Ethernet switches (Optional)

The NVIDIA® Spectrum™ SN2000 series switches are the 2nd generation of NVIDIA Ethernet switches, purpose-built for leaf, spine, and super-spine datacenter applications. SN2201 is ideal as an out-of-band (OOB) management switch, or as a top of rack (ToR) switch connecting to 48 x 1G Base-T host ports with four non-blocking 100 GbE spine uplinks. These switches are used for monitoring activities through Lenovo XClarity software and virtual operations if choosing a virtual environment. The choice of a management switch could be any compliant switch including existing switching network in the data center.

For this setup, the existing switches in the data center were used as Management Infrastructure.

NetApp ONTAP 9

NetApp ONTAP® 9 is the latest generation of storage management software from NetApp that enables businesses to modernize infrastructure and to transition to a cloud-ready data center. With industry-leading data management capabilities, ONTAP enables you to manage and protect your data with a single set of tools regardless of where that data resides. You can also move data freely to wherever you need it: the edge, the core, or the cloud. ONTAP 9 includes numerous features that simplify data management, accelerate and protect your critical data, and future-proof your infrastructure across hybrid cloud architectures.

Simplify Data Management

Data management is crucial for your enterprise IT operations so that you can use appropriate resources for your applications and datasets. ONTAP includes the following features to streamline and simplify your operations and reduce your total cost of operation:

- **Inline data compaction and expanded de-duplication.** Data compaction reduces wasted space inside storage blocks, and deduplication significantly increases effective capacity.
- **Minimum, maximum, and adaptive quality of service (QoS).** Granular QoS controls help maintain performance levels for critical applications in highly shared environments.
- **ONTAP FabricPool.** This feature provides automatic tiering of cold data to public and private cloud storage options, including Amazon Web Services (AWS), Azure, and NetApp StorageGRID® object-based storage.

Accelerate and Protect Data

ONTAP delivers superior levels of performance and data protection and extends these capabilities with the following features:

- **High performance and low latency.** ONTAP offers the highest possible throughput at the lowest possible latency.
- **NetApp ONTAP FlexGroup technology.** A FlexGroup volume is a high-performance data container that can scale linearly to up to 20PB and 400 billion files, providing a single namespace that simplifies data management.
- **Data protection.** ONTAP provides built-in data protection capabilities with common management across all platforms.
- **NetApp Volume Encryption.** ONTAP offers native volume-level encryption with both onboard and external key management support.

Future-Proof Infrastructure

ONTAP 9 helps to meet your demanding and constantly changing business needs:

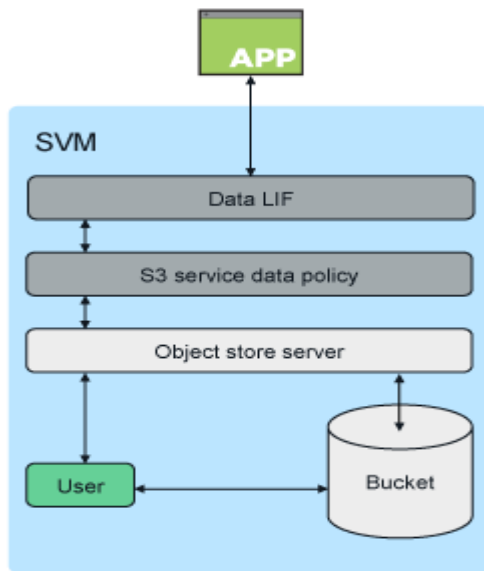
- **Seamless scaling and nondisruptive operations.** ONTAP supports the nondisruptive addition of capacity to existing controllers and to scale-out clusters. You can upgrade to the latest technologies, such as NVMe and end-to-end 100Gbe, without costly data migrations or outages.
- **Cloud connection.** ONTAP is one of the most cloud-connected storage management software, with options for software-defined storage (ONTAP Select) and cloud-native instances (NetApp Cloud Volumes Service) in all public clouds.
- **Integration with emerging applications.** By using the same infrastructure that supports existing enterprise apps, ONTAP offers enterprise-grade data services for next-generation platforms and applications such as OpenStack, Hadoop, and MongoDB.

NetApp ONTAP S3

NetApp ONTAP provides a robust and versatile storage solution for both containers and object storage, making it an ideal choice for modern data management needs. MLOps require an object store for saving models, datasets and pipeline artifacts. For containerized environments, ONTAP offers persistent storage, ensuring data integrity even when containers are restarted or moved. Additionally, ONTAP supports object storage through the S3 protocol, allowing it to manage large amounts of unstructured data efficiently. With scalable architecture, robust data protection features like SnapMirror, and support for multiple storage protocols, ONTAP delivers a unified and efficient storage infrastructure that meets the demands of both containerized and object storage environments.

You can enable an ONTAP Simple Storage Service (S3) object storage server in an ONTAP cluster, using familiar manageability tools such as ONTAP System Manager to rapidly provision high-performance object storage for development and operations in ONTAP and taking advantage of ONTAP's storage efficiencies and security.

Figure 3) ONTAP S3 – Logical diagram



NetApp AFF A-Series

The AFF A-Series storage family, powered by NetApp ONTAP® data management software, delivers the same NetApp simplicity and reliability that tens of thousands of organizations of every size, in every industry around the globe, have trusted for years. It's the same technology that the top three public clouds rely on to drive all your apps and data across hybrid clouds. No more silos, no more storage complexity. Just powerful, intelligent, secure storage to seamlessly accelerate your business.

All AFF A-Series systems offer advanced reliability, availability, and serviceability to keep your critical data available. They also provide comprehensive data management and data protection capabilities for your enterprise applications with industry-leading ONTAP software. Leverage unmatched consolidation and scale

- Consolidate all your workloads on AFF A-Series systems, which can:
- Deliver up to 2x better performance compared to previous generations of systems, with latency as low as 100µs
- Support any data type, any app workload, across hybrid cloud
- Provide consistent performance, adaptive quality of service (AQoS), and proven 99.9999% data availability to safeguard SLAs even in multi-workload and multitenant environments
- Scale non-disruptively to 185PB in a single namespace
- Improve the speed and productivity of collaborative teams across multiple locations and increase data throughput for read-intensive applications with NetApp FlexCache® software.

For more information on NetApp AFF A-Series, refer to

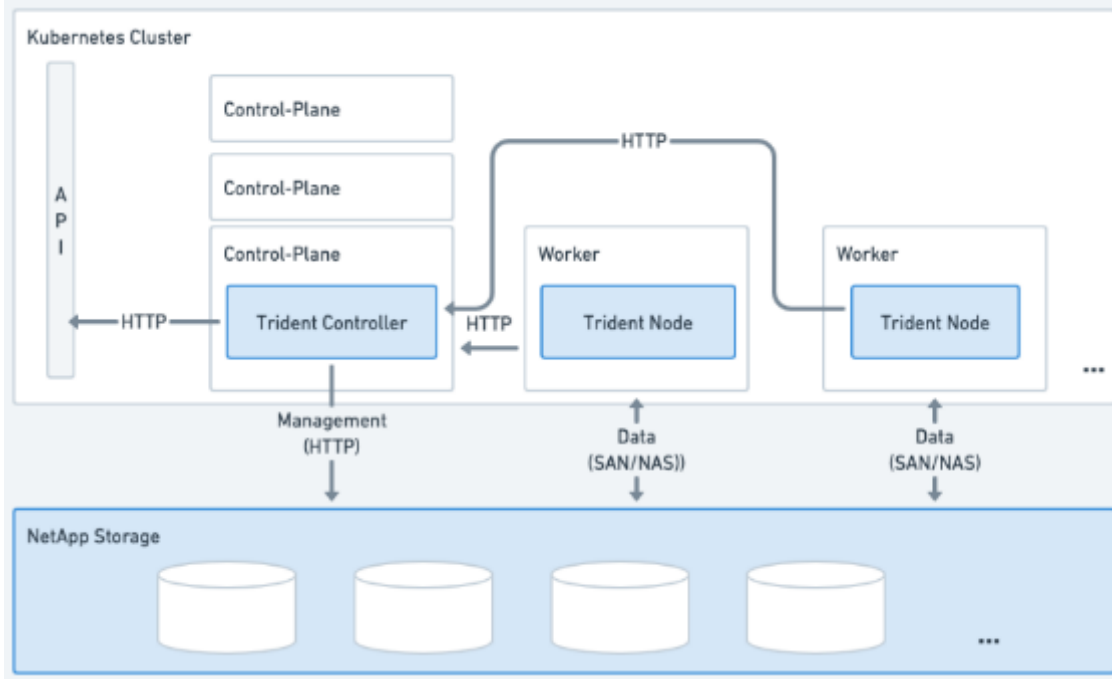
<https://www.netapp.com/pdf.html?item=/media/7828-ds-3582-aff-a-series-ai-era.pdf>.

NetApp Trident

Trident is an open-source storage orchestrator (Figure 4) developed and maintained by NetApp that greatly simplifies the creation, management, and consumption of persistent storage for Kubernetes workloads. Trident, itself a Kubernetes-native application, runs directly within a Kubernetes cluster. With Trident, Kubernetes users (developers, data scientists, Kubernetes administrators, and so on) can create,

manage, and interact with persistent storage volumes in the standard Kubernetes format that they are already familiar with. At the same time, they can take advantage of NetApp advanced data management capabilities and a data fabric that is powered by NetApp technology. Trident abstracts away the complexities of persistent storage and makes it simple to consume. For more information, refer [here](#).

Figure 4) NetApp Trident deployed on a High Availability Kubernetes Cluster



Red Hat OpenShift

Red Hat OpenShift is a robust enterprise application platform designed to streamline the entire application lifecycle, from development to deployment and maintenance. It leverages Kubernetes to provide a scalable and flexible environment for containerized applications.

Key Features:

- **OpenShift Serverless:** Enables developers to build and deploy serverless, event-driven applications without managing servers. It uses Kubernetes-native building blocks to simplify the development process
- **OpenShift Pipelines:** A cloud-native CI/CD solution based on Tekton, which automates deployments across multiple platforms. It abstracts the underlying implementation details, allowing for seamless integration and continuous delivery
- **OpenShift GitOps:** Utilizes Argo CD for continuous integration and deployment of cloud-native applications. It continuously monitors application definitions and configurations stored in Git repositories
- **OpenShift Virtualization:** Allows the management of virtual machine workloads alongside container workloads, providing a unified platform for diverse application

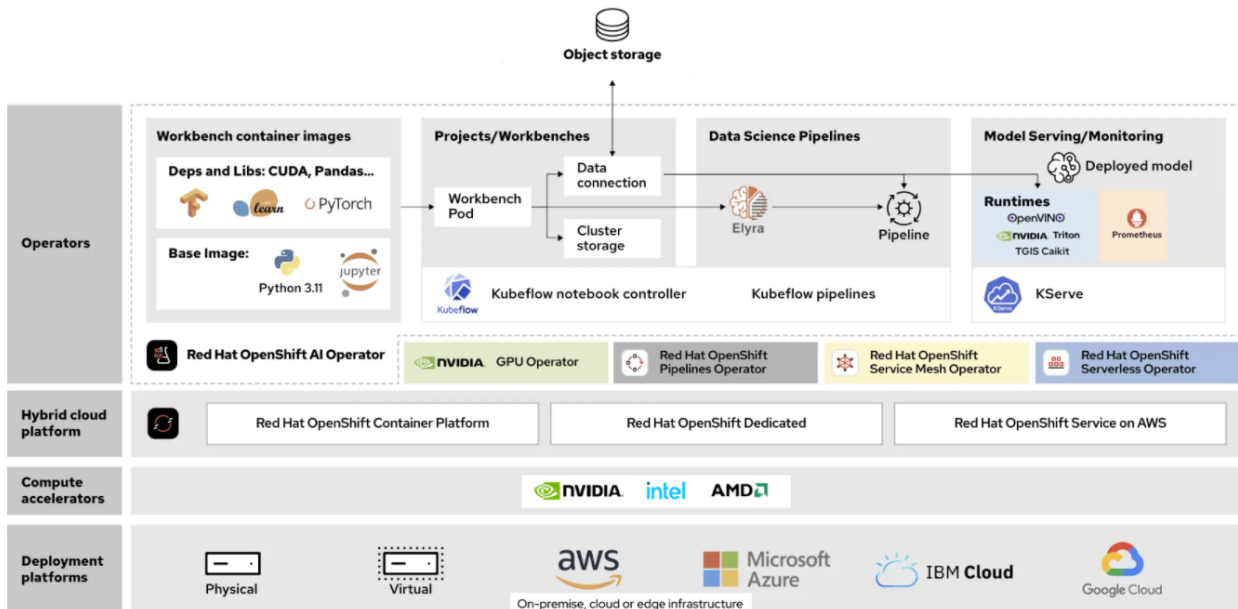
Red Hat OpenShift uses Red Hat Enterprise Linux CoreOS (RHCOS), a container-oriented operating system that is specifically designed for running containerized applications and provides several tools for fast installation, Operator-based management, and simplified upgrades.

Red Hat OpenShift AI

Red Hat OpenShift AI is a platform built on Red Hat OpenShift, providing a comprehensive suite of tools and components for developing, deploying, and managing AI models and applications at scale. It integrates various open-source technologies and provides a user-friendly interface for data scientists and developers. Red Hat OpenShift AI running on OpenShift provides a single enterprise-grade application platform for ML models and applications that use them. Data scientists, engineers, and app developers can collaborate in a single destination that promotes consistency, security, and scalability. OpenShift administrators that manage existing application environments can continue to do the same for OpenShift AI and ML workloads. This also allows application, ML, and data science teams to focus on their areas of work and spend less time managing the infrastructure.

OpenShift AI is compatible with leading AI tools and frameworks such as TensorFlow, PyTorch, and can work seamlessly with NVIDIA GPUs, to accelerate AI workloads. It provides pre-configured Jupyter notebook images with popular data science libraries. Red Hat tracks, integrates, tests, and supports common AI/ML tooling and model serving on Red Hat OpenShift. The latest release of Red Hat OpenShift AI delivers enhanced support for predictive and generative AI model serving and improves efficiency of data processing and model training

Figure 5) Red Hat OpenShift AI architecture and components



NVIDIA AI Enterprise (NVAIE)

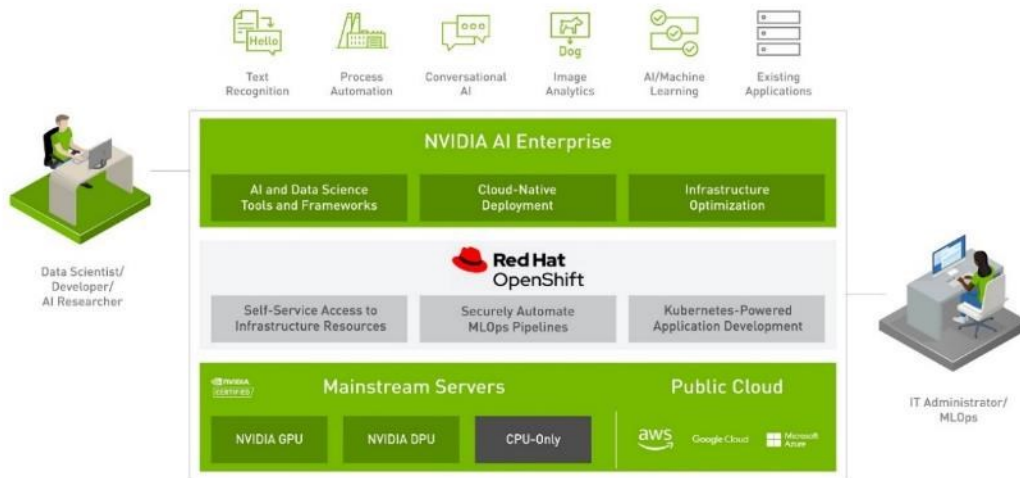
NVIDIA AI Enterprise is an end-to-end, cloud-native software platform that accelerates data science pipelines and streamlines development and deployment of production-grade co-pilots and other generative AI applications. Easy-to-use microservices provide optimized model performance with enterprise-grade security, support, and stability to ensure a smooth transition from prototype to production for enterprises that run their businesses on AI.

Red Hat and NVIDIA have partnered to unlock the power of AI for every business by delivering an end-to-end enterprise platform optimized for AI workloads. This integrated platform delivers best-in-class AI software, the NVIDIA AI Enterprise suite, optimized and certified for Red Hat OpenShift, the industry's leading Containers and Kubernetes platform. Running on NVIDIA-Certified Systems™, industry-leading accelerated servers, this platform accelerates the speed at which developers can build AI and high-

performance data analytics, enabling organizations to scale modern workloads on the same infrastructure they have already invested in, and delivers enterprise-class manageability, security, and availability.

For more information on NVAIE, refer to <https://docs.nvidia.com/ai-enterprise/deployment/openshift-on-bare-metal/latest/introduction.html>.

Figure 6) NVAIE and Red Hat OpenShift Integration



NVIDIA AI Blueprint

NVIDIA AI blueprint outlines a comprehensive strategy for building and scaling artificial intelligence across industries. It combines powerful GPU hardware, cutting-edge software frameworks like CUDA and TensorRT, and AI platforms. Central to the blueprint is the NVIDIA AI Enterprise suite, which simplifies deployment for businesses. NVIDIA also emphasizes partnerships and ecosystems, supporting generative AI, autonomous systems, and edge AI through platforms like Omniverse, Isaac, and Jetson. Overall, NVIDIA's AI blueprint aims to democratize access to AI and accelerate innovation across sectors.

[NVIDIA AI Blueprints](https://github.com/NVIDIA-AI-Blueprints) are reference architecture that illustrate how NVIDIA NIM can be leveraged to build innovative solutions. Example includes customizable generative AI reference architectures designed to equip enterprise developers with essential assets such as NIM microservices, reference code, detailed documentation, and Helm charts for deployment. These blueprints serve as a foundation for building advanced AI virtual assistants, either as standalone applications or as enhancements to existing systems. Their focus is on enabling personalization, summarization, and sentiment analysis, particularly using generative AI for data that's often inaccessible. For more information about NVIDIA AI Blueprints reference architecture, refer to <https://github.com/NVIDIA-AI-Blueprints>.

Solution Design

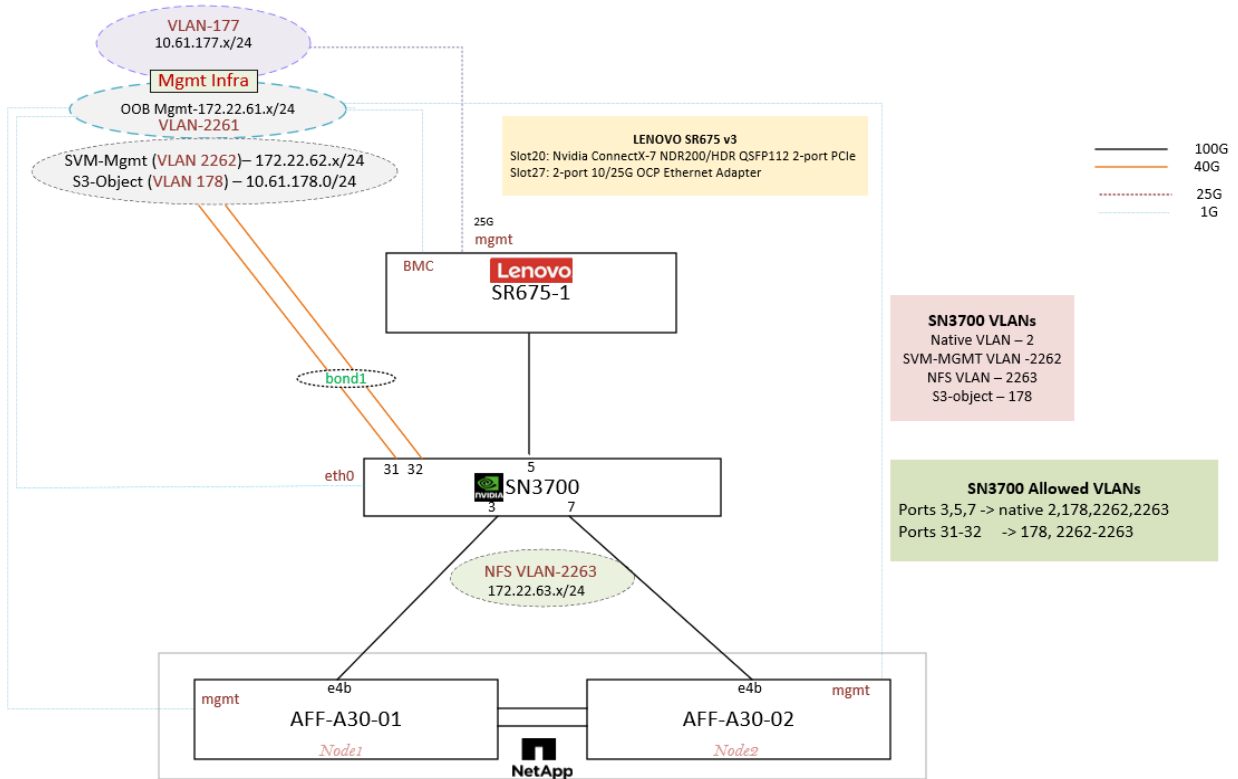
Single node deployment

The solution topology (Figure 7) consists of single Lenovo Think System SR675 V3 server with 2x L40S GPUs, one NVIDIA SN3700 switch, and NetApp's latest AFF A30 storage. The server has the option of 1G/10G/25G/100G 4-port OCP adapter and the first port connects to the management infrastructure. For this configuration, a 25G OCP adapter was utilized. However, customers can choose and use an adapter

that meets their specific needs. Note that only one OCP card can be installed on the server. For additional details on supported OCP adapters, please refer [here](#). Server also has one ConnectX-7 connect adapter for storage connectivity.

This solution was tested with NVIDIA L40S GPU, but it can technically utilize any NVIDIA DW GPU with the same capacity, including the H200 NVL, RTX Pro 6000, or the H100 NVL.

Figure 7) Solution Topology (Single Node)



The SN3700 switch has a single 100G link connecting to the CX-7 adapter on SR675 server. This link carries the NFS and S3 traffic from the SR675 server. The SN3700 switch has a bond interface with two 40G links connecting to the core infrastructure and it also carries the In-Band management VLANs. The bond interfaces are configured as trunks with native VLAN (2). It also has interfaces with 100G ports connecting to each NetApp AFF A30 controller carrying the SVM management, NFS and S3 VLANs.

Each NetApp controller has a 100G port (e4b) carrying the NFS, S3 and SVM management traffic. The second 100G port (e2b) is reserved to connect to a redundant switch, when available.

Scaling within Single Node OpenShift

To accommodate increasing workloads for inferencing, the server solution can be scaled by integrating six additional GPUs, resulting in a total of eight GPUs on a single server. This expansion enhances computational power, ensuring efficient processing and improved performance. By scaling up, the system can handle more demanding tasks seamlessly. This approach provides a robust solution for growing computational needs.

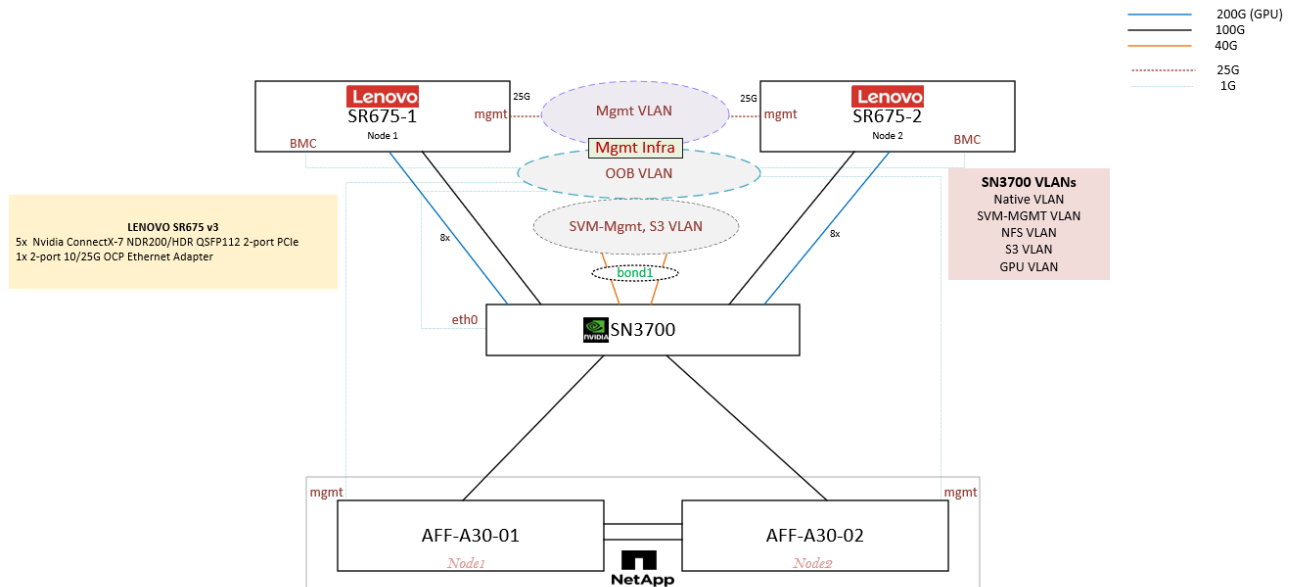
Scaling OpenShift with additional compute node

For AI training use cases that demand more than 8 GPUs, this solution can be scaled by adding another server equipped with GPUs. Proper alignment of ConnectX-7 adapters is necessary when integrating an

additional server into the solution. Both servers must be equipped with enough ConnectX-7 cards to establish East-West networking for the GPUs.

It is recommended to maintain a 2x GPU to 1x ConnectX-7 card ratio to ensure 200G connectivity to each GPU. To set up connectivity for East-West GPU traffic, you'll need to configure the network (GPU VLAN) and ports on the SN3700 and the interfaces on OpenShift node. The [Kubernetes NMState](#) operator can be utilized to do the interface configurations within OpenShift.

Figure 8) Solution topology with additional compute node



The above topology includes two nodes for the training use case, each equipped with 8 GPUs. Each Lenovo SR675 V3 server with the following configuration:

- 2x AMD EPYC 9634 processors, 84C, 2.25GHz or 2x AMD EPYC 9354 processors, 32C, 3.25GHz or 2x AMD EPYC 9535 processors, 64C, 2.4GHz
- 4x 128GB TruDDR5 3DS RDIMMs
- 1x 25GbE 4-port OCP Adapter – Management Connector
- 2x M.2 NVMe SSD drives
- 8x NVIDIA L40S GPUs
- 5x NVIDIA ConnectX-7 200 GbE dual port
 - 1x ConnectX-7 card – Single link connected to SN3700 switch for storage (NFS/S3) traffic
 - 4x ConnectX-7 cards – All ports are connected to the same SN3700 switch for East-West GPU traffic

This setup results in a total of eight links per server connected to the SN3700 switch to provide East-West connectivity. This configuration ensures sufficient bandwidth for East-West GPU traffic, providing 200Gbps per GPU.

Note: Adding a worker node to single node OpenShift does not expand the cluster control plane, and it does not provide high availability to your cluster.

Note: When adding worker nodes to single-node OpenShift clusters, a tested maximum of two worker nodes is recommended.

Solution Deployment

Network configuration

In this section, we will discuss the configuration of NVIDIA switches. For security and traffic isolation we configured VLANs on the switch. Below are the VLANs used for the solution validation.

Table 3) VLAN information

VLAN Name	ID	Remarks
Native VLAN	2	Native VLAN
OOB-Mgmt VLAN	2261	Out of Band Management (Existing Infra)
IB-Mgmt VLAN	2262	In-band Management
NFS VLAN	2263	NFS VLAN
S3 VLAN	178	ONTAP S3 VLAN
GPU VLAN (OPTIONAL)	100	Only in dual node topology

In reference to the single node topology, the NVIDIA Cumulus Linux is used to configure the spectrum switches. These switches are configured as Layer 2 devices, so the Layer 1 & Layer 2 configuration commands are discussed here.

The following section provides a list of commands used to configure the switches. After each step, you may apply the configuration using `nv config apply` command.

1. Configure the system hostname.

```
nv set system hostname <hostname>
```

2. Configure the management interface.

```
nv set interface eth0 ip address <ip address/mask>
```

3. Set link speed and link state.

```
nv set interface <interface_name> link speed <speed>
nv set interface <interface_name> link state up
```

4. Configure a bridge domain and set various parameters such as VLAN, untagged VLAN, STP priority (optional).

```
nv set bridge domain <bridge_domain> vlan <VLAN>
nv set bridge domain <bridge_domain> untagged <VLAN>
nv set bridge domain <bridge_domain> stp priority <priority>
```

5. Configure a bond interface, add bond members and add the interface as a trunk to the bridge domain

```
nv set interface <bond_interface> bond member <member ports>
nv set interface <bond_interface> bond mode lacp
nv set interface <bond_interface> bridge domain <bridge_domain> vlan <VLAN>
```

6. Configure an access port and add it to the bridge domain

```
nv set interface <interface_name> bridge domain <bridge_domain> access <VLAN>
```

7. Configure switchport breakout.


```
nv set interface <interface_name> link breakout <breakout_option>
```

8. Enable RoCE protocol.

```
nv set qos roce
```

9. Save the configuration across reboots.

```
nv config save
```

10. Display the configuration

```
nv config show
```

NVIDIA SN3700 - Running configuration

The SN3700 switch is configured as a VLAN-aware bridge (bridge1) that carries SVM management VLAN (2262), NFS VLAN (2263) and S3 VLAN (178). Switch ports swp31-32 form a bond interface bond1 and carry traffic of VLAN 178,2262-2263. Switch port swp3, swp5-6, swp7 are also configured as individual interface and carry tagged VLAN 178, 2262-2263. Interface eth0 is configured as a switch management interface, and it connects to the OOB management network in VLAN 2261.

For more information on configuring VLAN-aware bridge, refer to the following URL.

[VLAN-aware Bridge Mode | Cumulus Linux v5.9 \(nvidia.com\)](#)

The SN3700 switch configuration is shown below.

```
cumulus@aipod-sn3700-1:mgmt:~$ nv config show
- header:
  model: MSN3700C
  nvue-api-version: nvue_v1
  rev-id: 1.0
  version: Cumulus Linux 5.9.1
- set:
  bridge:
    domain:
      bridge1:
        stp:
          priority: 8192
          untagged: 2
          vlan:
            100,178,2262-2263: {}
  interface:
    bond1:
      bond:
        member:
          swp31: {}
          swp32: {}
        mode: lacp
      bridge:
        domain:
          bridge1:
            vlan:
              178,2262-2263: {}
        type: bond
    eth0:
      ip:
        address:
          172.22.61.19/24: {}
        gateway:
          172.22.61.1: {}
        vrf: mgmt
    eth0,swp11-12:
      type: eth
    swp5-6:
      link:
        auto-negotiate: on
```

```

swp1-3,5-7,9-10,31-32:
  type: swp
swp1-3,5-7,31-32:
  link:
    state:
      up: {}
swp3,5-7:
  link:
    speed: 100G
swp3,5-7,9-10:
  bridge:
    domain:
      bridge1:
        vlan:
          178,2262-2263: {}
swp3,7:
  link:
    mtu: 9000
swp9-12:
  link:
    state:
      down: {}
swp31-32:
  link:
    speed: 40G
qos:
  roce:
    enable: on
service:
  ntp:
    mgmt:
      server:
        0.10.61.176.251: {}
        1.10.61.176.252: {}
        2.cumulusnetworks.pool.ntp.org: {}
        3.cumulusnetworks.pool.ntp.org: {}
system:
  aaa:
    class:
      nvapply:
        action: allow
        command-path:
          /:
            permission: all
      nvshow:
        action: allow
        command-path:
          /:
            permission: ro
      sudo:
        action: allow
        command-path:
          /:
            permission: all
    role:
      nvue-admin:
        class:
          nvapply: {}
      nvue-monitor:
        class:
          nvshow: {}
      system-admin:
        class:
          nvapply: {}
          sudo: {}
  user:
    aipod1:
      full-name: AIPOD USER1
      hashed-password: '*'
      role: nvue-monitor
      cumulus:

```

```

    full-name: cumulus,,,
    hashed-password: '*'
    role: system-admin
api:
  state: enabled
config:
  auto-save:
    enable: on
control-plane:
  acl:
    acl-default-dos:
      inbound: {}
    acl-default-whitelist:
      inbound: {}
hostname: aipod-sn3700-1
reboot:
  mode: cold
ssh-server:
  state: enabled
wjh:
  channel:
    forwarding:
      trigger:
        12: {}
        13: {}
        tunnel: {}
  enable: on

```

The CLI command displays the port-vlan configuration on this switch.

```

cumulus@aipod-sn3700-1:mgmt:/var/home/cumulus$ nv show bridge port-vlan
domain      port      vlan      tag-state
-----
bridge1     bond1     2         untagged
            178       tagged
            2262-2263 tagged
            swp3     2         untagged
            178       tagged
            2262-2263 tagged
            swp5     2         untagged
            178       tagged
            2262-2263 tagged
            swp6     2         untagged
            178       tagged
            2262-2263 tagged
            swp7     2         untagged
            178       tagged
            2262-2263 tagged

```

Lenovo ThinkSystem server configuration

This section discusses about the Lenovo ThinkSystem server's initial setup, before an operating system can be installed. In this section, we will talk about the BMC and Local Boot configuration. The major steps are listed below.

- Configure BMC using Lenovo XClarity Provisioning Manager
- Configure RAID for local boot from M.2 NVMe drives

Configure BMC using Lenovo XClarity Provisioning Manager

The Lenovo XClarity Provisioning Manager was used in the lab to connect the Lenovo XClarity Controller to the management network. Refer to the following steps more details.

1. Start the server.

2. Press the key specified in the on-screen instructions to display the Lenovo XClarity Provisioning Manager interface.
3. Go to **LXPM** → **UEFI Setup** → **BMC Settings** to specify how the Lenovo XClarity Controller will connect to the network.
 - If you choose a static IP connection, make sure that you specify an IPv4 or IPv6 address that is available on the network.
 - If you choose a DHCP connection, make sure that the MAC address for the server has been configured in the DHCP server.
4. Click OK to apply the setting and wait for two to three minutes.
5. Use an IPv4 or IPv6 address to connect Lenovo XClarity Controller.

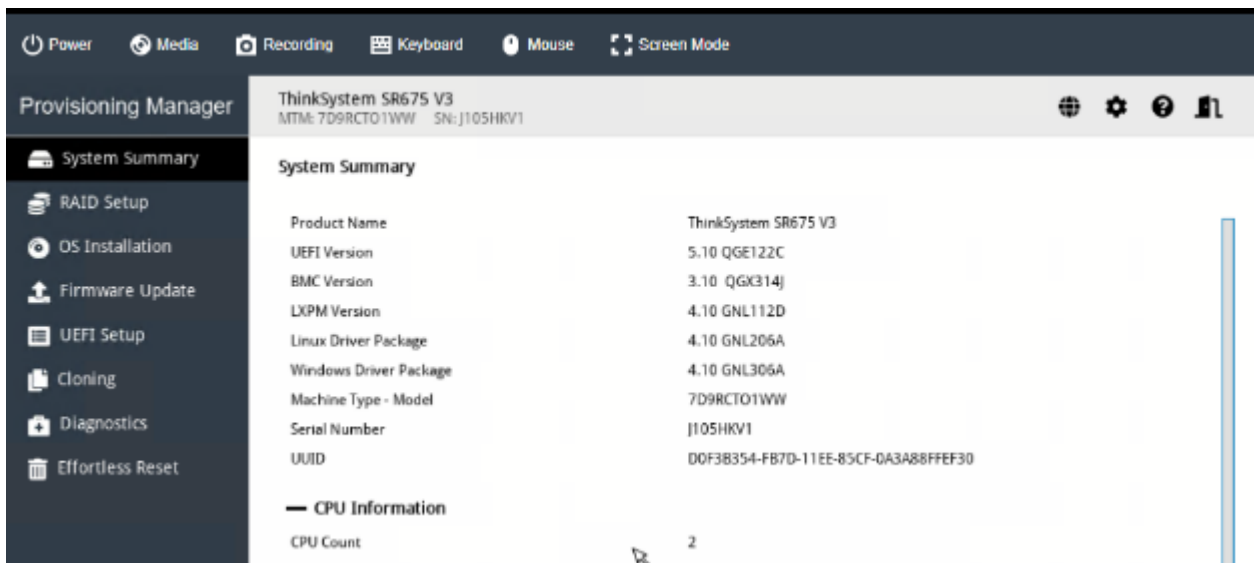
For more information, refer to <https://pubs.lenovo.com/lxpm-overview/>.

Note: The Lenovo XClarity Controller is set initially with a username of USERID and password of PASSW0RD (with a zero, not the letter O). This default user setting has Supervisor access. It is required to change this username and password during your initial configuration for enhanced security.

Note: Once you have configured BMC, you can access the XClarity Controller interface using the web interface.

Check (Figure 13) for the Lenovo XClarity Provisioning Manager screen.

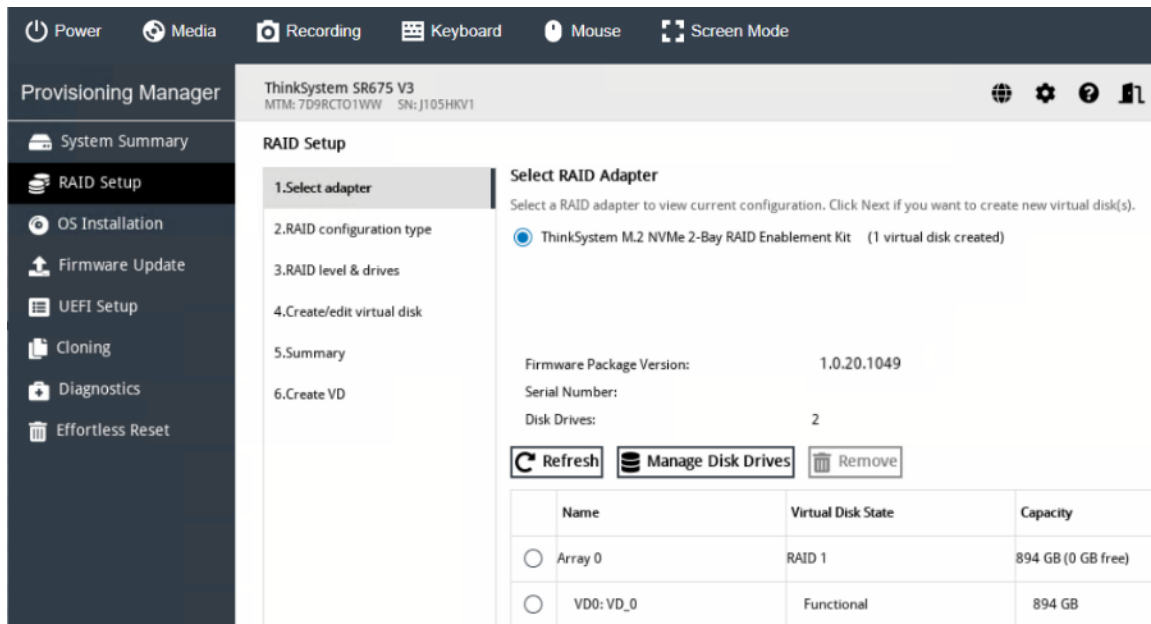
Figure 9) Lenovo XClarity Provisioning Manager



Configure RAID

1. In the Provisioning Manager, click on “RAID Setup” to select the M.2 NVMe RAID adapter and configure the RAID settings.
2. Select the RAID type as RAID1.
3. Select both disks and move it to the right.
4. Create virtual disk
5. Check the summary and click Create.

Figure 10 RAID configuration for M.2 NVMe drives



Note: The server is ready for Operating System installation.

NetApp AFF A30 configuration

See the following section ([NetApp Hardware Universe](#)) for planning the physical location of the storage systems:

- Site Preparation
- System Connectivity Requirements
- Circuit Breaker, Power Outlet Balancing, System Cabinet Power Cord Plugs, and Console Pinout Requirements
- AFF Series Systems

NetApp Hardware Universe

The NetApp Hardware Universe (HWU) application provides supported hardware and software components for any specific ONTAP version. It also provides configuration information for all the NetApp storage appliances currently supported by ONTAP software and a table of component compatibilities.

To confirm that the hardware and software components that you would like to use are supported with the version of ONTAP that you plan to install, follow these steps at the [NetApp Support](#) site:

1. Access the [HWU application](#) to view the System Configuration guides.
2. Click the Products tab to select Platforms menu to view the compatibility between different versions of the ONTAP software and the NetApp storage appliances with your desired specifications.
3. Alternatively, to compare components by storage appliance, click Utilities and select compare Storage Systems.

Controllers

Follow the physical installation procedures for the controllers found here: <https://docs.netapp.com/us-on/ontap-systems/index.html>.

Disk Shelves

NetApp storage systems support a wide variety of disk shelves and disk drives. The complete list of disk shelves that is supported by the AFF A30 is available at the [HWU](#) site.

When using NVMe drive shelves with NetApp storage controllers, refer to: <https://docs.netapp.com/us-en/ontap-systems/ns224/hot-add-shelf.html> for installation and servicing guidelines.

To setup a new ONTAP cluster, follow the instruction [here](#):

To apply ONTAP license, follow the instructions [here](#):

ONTAP Configuration

1. Once the cluster is setup, log into the Cluster and verify Storage Failover

```
storage failover show
```

Node	Partner	Takeover Possible	State	Description
aipod-a30-01	aipod-a30-02	true		Connected to aipod-a30-02
aipod-a30-02	aipod-a30-01	true		Connected to aipod-a30-01

2 entries were displayed.

2 entries were displayed.

Note: Both <st-node01> and <st-node02> must be capable of performing a takeover. Continue with Step 2 if the nodes can perform a takeover.

2. Enable failover on one of the two nodes if it was not completed during the installation:

```
storage failover modify -node <st-node01> -enabled true
```

Note: Enabling failover on one node enables it for both nodes.

3. Verify the HA status for a two-node cluster.

Note: This step is not applicable for clusters with more than two nodes.

```
cluster ha show
High-Availability Configured: true
```

4. If HA is not configured use the below commands. Only enable HA mode for two-node clusters.

Note: Do not run this command for clusters with more than two nodes because it causes problems with failover.

```
cluster ha modify -configured true
Do you want to continue? {y|n}: y
```

5. Verify that hardware assist is correctly configured:

```
storage failover hwassist show -node *
```

6. Set Auto-Revert on Cluster Management Interface

To set the auto-revert parameter on the cluster management interface, follow this step:

```
network interface modify -vserver <clustername> -lif cluster_mgmt_lif_1 -auto-revert true
```

Note: A storage virtual machine (SVM) is referred to as a Vserver or vservers in the GUI and CLI.

7. To zero all spare disks in the cluster, run the following command:

```
disk zerospares
```

Note: Advanced Data Partitioning creates a root partition and two data partitions on each SSD drive in an AFF configuration. Disk auto assign should have assigned one data partition to each node in an HA pair. If a different disk assignment is re-quired, disk auto assignment must be disabled on both nodes in the HA pair by running the `disk option modify` command. Spare partitions can then be moved from one node to another by running the `disk removeowner` and `disk assign` commands.

Create Aggregate

An aggregate containing the root volume is created during the ONTAP setup process. To manually create additional aggregates, determine the aggregate name, the node on which to create it, and the number of disks it should contain.

1. To create new aggregates, run the following commands:

```
storage aggregate create -aggregate <aggr1_node01> -node <st-node01> -diskcount <num-disks> -
diskclass solid-state

storage aggregate create -aggregate <aggr1_node02> -node <st-node02> -diskcount <num-disks> -
diskclass solid-state
```

Note: You should have the minimum number of hot spare disks for the recommended hot spare disk partitions for their aggregate.

Note: In an AFF configuration with a small number of SSDs, you might want to create an aggregate with all, but one remaining disk (spare) assigned to the controller.

Note: The aggregate cannot be created until disk zeroing completes. Run the `storage aggregate show` command to display the aggregate creation status. Do not proceed until both `aggr1_node01` and `aggr1_node02` are online.

Configure FIPS

1. Enable FIPS Mode on the NetApp ONTAP Cluster (Optional)

NetApp ONTAP is compliant in the Federal Information Processing Standards (FIPS) 140-2 for all SSL connections. When SSL FIPS mode is enabled, SSL communication from NetApp ONTAP to external client or server components outside of NetApp ONTAP will use FIPS compliant crypto for SSL.

```
set -privilege advanced
security config modify -interface SSL -is-fips-enabled true
```

Configure Time zone

1. Set the timezone for the cluster

```
timezone -timezone <timezone>
```

2. Verify the timezone setting.

```
date
```

Configure Simple Network Management Protocol

Note: If you have enabled FIPS then please look at the following points while configuring SNMP.

- The SNMP users or SNMP traphosts that are non-compliant with FIPS will be deleted automatically. "Configure SNMP traphosts" configuration will be non-compliant with FIPS.
- The SNMPv1 user, SNMPv2c user (After configuring SNMP community) or SNMPv3 user (with none or MD5 as authentication protocol or none or DES as encryption protocol or both) is non-compliant with FIPS.

1. Configure basic SNMP information, such as the location and contact. When polled, this information is visible as the sysLocation and sysContact variables in SNMP.

```
snmp contact <snmp-contact>
snmp location "<snmp-location>"
snmp init 1
options snmp.enable on
```

2. Configure SNMP traps to send to remote hosts, such as an Active IQ Unified Manager server or another fault management system.

Note: This step works when FIPS is disabled. An SNMPv1 traphost or SNMPv3 traphost (configured with an SNMPv3 user non-compliant to FIPS) is non-compliant to FIPS.

```
snmp traphost add <oncommand-um-server-fqdn>
```

3. Configure SNMP community.

Note: This step works when FIPS is disabled. SNMPv1 and SNMPv2c are not supported when cluster FIPS mode is enabled.

```
system snmp community add -type ro -community-name <snmp-community> -vserver <clustername>
```

4. Configure SNMPv3 Access

SNMPv3 offers advanced security by using encryption and passphrases. The SNMPv3 user can run SNMP utilities from the traphost using the authentication and privacy settings that you specify.

Note: When FIPS is enabled, the following are the supported/compliant options for authentication and privacy protocol:

- Authentication Protocol: sha, sha2-256
- Privacy protocol: aes128

```
security login create -user-or-group-name <<snmp-v3-usr>> -application snmp -authentication-
method usm

Enter the authoritative entity's EngineID [local EngineID]:
Which authentication protocol do you want to choose (none, md5, sha, sha2-256) [none]: <<snmp-v3-
auth-proto>>

Enter the authentication protocol password (minimum 8 characters long):
Enter the authentication protocol password again:
Which privacy protocol do you want to choose (none, des, aes128) [none]: <<snmpv3-priv-proto>>
Enter privacy protocol password (minimum 8 characters long):
Enter privacy protocol password again:
```

Note: Refer to the [SNMP Configuration Express Guide](#) for additional information when configuring SNMPv3 security users.

Configure Service Processor

1. Set Up Service Processor Network Interface

```
system service-processor network modify -node <st-node01> -address-family IPv4 -enable true -dhcp
none -ip-address <node01-sp-ip> -netmask <node01-sp-mask> -gateway <node01-sp-gateway>

system service-processor network modify -node <st-node02> -address-family IPv4 -enable true -dhcp
none -ip-address <node02-sp-ip> -netmask <node02-sp-mask> -gateway <node02-sp-gateway>
```

Note: The Service Processor IP addresses should be in the same subnet as the node management IP addresses.

2. Verify the service processor configuration.

```
system service-processor show
```

Remove Default Broadcast Domains

By default, all network ports are included in separate default broadcast domain. Network ports used for data services (for example e0e, e0f, and so on) should be removed from their default broadcast domain and that broadcast domain should be deleted.

```
network port broadcast-domain delete -broadcast-domain <Default-N> -ip-space Default
network port broadcast-domain show
```

Note: Delete the Default broadcast domains with Network ports.

Disable flow control on data ports

1. Disable Flow Control on 25/100GbE Data Ports and verify.

```
network port modify -node <st-node01> -port e4a,e4b -flowcontrol-admin none
network port modify -node <st-node02> -port e4a,e4b -flowcontrol-admin none
network port show -node * -port e4a,e4b -fields speed-admin,duplex-admin,flowcontrol-admin
```

Enable network discovery protocols

1. Enable Link-layer Discovery Protocol (LLDP).

```
node run * options lldp.enable on
```

Create Broadcast Domain

Note: If the management interfaces are required to be on a separate VLAN, create a new broadcast domain for those interfaces by running the following command:

1. Create NFS Broadcast Domain with recommended MTU

```
network port broadcast-domain create -broadcast-domain OCP-MGMT -mtu 1500
```

2. Create NFS Broadcast Domain with recommended MTU

```
network port broadcast-domain create -broadcast-domain OCP-NFS -mtu 9000
```

3. Create S3 Broadcast Domain with recommended MTU

```
network port broadcast-domain create -broadcast-domain OCP-S3 -mtu 9000
```

Create VLAN

1. Create the MGMT VLAN ports and add them to the OCP-MGMT broadcast domain:

```
network port vlan create -node aipod-a30-01 -vlan-name e4b-2262
network port vlan create -node aipod-a30-02 -vlan-name e4b-2262
network port broadcast-domain add-ports -broadcast-domain OCP-MGMT -ports aipod-a30-01:e4b-2262,aipod-a30-02:e4b-2262
```

2. Create the NFS VLAN ports and add them to the OCP-NFS broadcast domain:

```
network port vlan create -node aipod-a30-01 -vlan-name e4b-2263
```

```
network port vlan create -node aipod-a30-02 -vlan-name e4b-2263

network port broadcast-domain add-ports -broadcast-domain OCP-NFS -ports aipod-a30-01:e4b-2263,aipod-a30-02:e4b-2263
```

3. Create S3 VLAN ports and add then to the OCP-S3 broadcast domain:

```
network port vlan create -node aipod-a30-01 -vlan-name e4b-178
network port vlan create -node aipod-a30-02 -vlan-name e4b-178

network port broadcast-domain add-ports -broadcast-domain OCP-S3 -ports aipod-a30-01:e4b-178,aipod-a30-02:e4b-178
```

4. Create SVM (Storage Virtual Machine)

Note: The SVM is used to configure NFS services for storage access by NetApp Astra Trident.

```
vserver create -vserver OCP-SVM
```

Add protocol and create service policy

1. Add NFS protocol to OCP-SVM.

```
vserver add-protocols -protocols nfs -vserver OCP-SVM
```

2. Remove the unused data protocols from the SVM:

```
vserver remove-protocols -vserver OCP-SVM -protocols iscsi, fcp, nvme, ndmp, cifs
```

3. Enable and run the NFS protocol

```
vserver nfs create -vserver OCP-SVM -udp disabled -v3 enabled -v4.1 enabled
```

Note: If the NFS license was not installed during the cluster configuration, make sure to install the license before starting the NFS service.

4. Vserver Protocol Verification

```
vserver show-protocols -vserver OCP-SVM
```

Add Aggregate

1. Add the two data aggregates to the OCP-SVM vserver.

```
vserver modify -vserver OCP-SVM -aggr-list aipod_a30_01_NVME_SSD_1,aipod_a30_02_NVME_SSD_1
```

Configure LS mirror

1. Create a load-sharing mirror volume of the “OCP-SVM” SVM root volume on the node that does not have the Root Volume:

```
volume show -vserver OCP-SVM # Note down the aggregate and node for the root volume

volume create -vserver OCP-SVM -volume OCP_SVM_root_lsm01 -aggregate fpsa_aipod_a30_02_NVME_SSD_1 -size 1GB -type DP
```

2. Create a job schedule to update the root volume mirror relationships every 15 minutes:

```
job schedule interval create -name lsm-15min -minutes 15
```

3. Create mirroring relationships:

```
snapmirror create -source-path OCP-SVM:OCP_SVM_root -destination-path OCP-SVM:OCP_SVM_root_lsm01
-type LS -schedule lsm-15min
```

4. Initialize the mirroring relationship:

```
snapmirror initialize-ls-set -source-path OCP-SVM:OCP_SVM_root
[Job 79] Job is queued: "snapmirror initialize-ls-set" for source "fpsa-aipod-a30://OCP-
SVM/OCP_SVM_root".
```

To verify:

```
snapmirror show
```

Source Path	Type	Destination Path	Mirror State	Relationship Status	Total Progress	Progress Healthy	Last Updated
fpsa-aipod-a30://OCP-SVM/OCP_SVM_root	LS	fpsa-aipod-a30://OCP-SVM/OCP_SVM_root_lsm01	Snapmirrored	Idle	-	true	-

Configure HTTPS Access, vServer Admin and Banner

1. To configure secure access to the storage controller, follow these steps:

```
set -privilege diag
Do you want to continue? {y|n}: y
```

2. A self-signed certificate is already in place. Verify the certificate and obtain parameters (for example, the <serial-number>) by running the following command:

```
security certificate show
```

3. For each SVM shown, the certificate common name should match the DNS fully qualified domain name (FQDN) of the SVM. Delete the two default certificates and replace them with either self-signed certificates or certificates from a certificate authority (CA). To delete the default certificates, run the following commands:

```
security certificate delete -vserver OCP-SVM -common-name OCP-SVM -ca OCP-SVM -type server -
serial <serial-number>
```

Note: Deleting expired certificates before creating new certificates is best practice. Run the **security certificate delete** command to delete the expired certificates. In the following command, use TAB completion to select and delete each default certificate.

4. To generate and install self-signed certificates, run the following commands as one-time commands. Generate a server certificate for the OCP-SVM and the cluster SVM. Use TAB completion to aid in the completion of these commands.

```
security certificate create -common-name <cert-common-name> -type server -size 2048 -country
<cert-country> -state <cert-state> -locality <cert-locality> -organization <cert-org> -unit
<cert-unit> -email-addr <cert-email> -expire-days <cert-days> -protocol SSL -hash-function SHA256
-vserver OCP-SVM
```

5. Obtain the values for the parameters required in step 5 (<cert-ca> and <cert-serial>)

```
security certificate show
```

6. Enable each certificate that was just created by using the `-server-enabled true` and `-client-enabled false` parameters. Use TAB completion to aid in the completion of these commands.

```
security ssl modify -vserver <clustername> -server-enabled true -client-enabled false -ca <cert-
ca> -serial <cert-serial> -common-name <cert-common-name>
```

7. Disable HTTP cluster management access.

```
network interface service-policy remove-service -vserver <clustername> -policy default-management  
-service management-http
```

Note: It is normal for some of these commands to return an error message stating that the entry does not exist.

8. Change back to the normal admin privilege level and verify that the system logs are available in a web browser.

```
set -privilege admin  
  
https://<node01-mgmt-ip>/spi  
https://<node02-mgmt-ip>/spi
```

9. Set password for SVM vsadmin user and unlock the user

```
security login password -username vsadmin -vserver OCP-SVM  
Enter a new password: <password>  
Enter it again: <password>  
  
security login unlock -username vsadmin -vserver OCP-SVM
```

10. Configure login banner for the SVM

```
security login banner modify -vserver OCP-SVM -message "This OCP-SVM is reserved for authorized  
users only!"
```

Configure NFS export policy

1. Create a new rule for the infrastructure NFS subnet in the default export policy:

```
vserver export-policy rule create -vserver OCP-SVM -policyname default -ruleindex 1 -protocol nfs  
-clientmatch 172.22.63.0/24 -rorule sys -rwrule sys -superuser sys -allow-suid true
```

Note: For more information on configuring NFS Export Policy for Trident, go to: <https://docs.netapp.com/us-en/trident/trident-use/ontap-nas-prep.html#requirements>.

Note: This step is crucial when using NFS storage driver in NetApp Trident since this SVM will be added as a Trident Backend.

2. Assign the export policy to the SVM (OCP-SVM) root volume:

```
volume modify -volume OCP_SVM_root -policy default
```

Create FlexVol® Volumes

The following information is required to create a NetApp FlexVol® volume:

- The volume name
- The volume size
- The aggregate on which the volume exists

1. Run the following command to create a volume for storing SVM audit log configuration:

```
volume create -vserver OCP-SVM -volume audit_log -aggregate aipod_a30_01_NVME_SSD_1 -size 50GB -  
state online -policy default -junction-path /audit_log -space-guarantee none -percent-snapshot-  
space 0  
  
snapmirror update-ls-set -source-path OCP-SVM:OCP_SVM_root # Update set of load-sharing mirrors  
  
vserver audit create -vserver OCP-SVM -destination /audit_log
```

```
vserver audit enable -vserver OCP-SVM
```

Create NFS LIFs

1. To create NFS LIFs run the below commands.

```
network interface create -vserver OCP-SVM -lif nfs-lif-01 -service-policy default-data-files -
home-node aipod-a30-01 -home-port e4b-2263 -address 172.22.63.51 -netmask 255.255.255.0 -status-
admin up -failover-policy broadcast-domain-wide -auto-revert true

network interface create -vserver OCP-SVM -lif nfs-lif-02 -service-policy default-data-files -
home-node aipod-a30-02 -home-port e4b-2263 -address 172.22.63.52 -netmask 255.255.255.0 -status-
admin up -failover-policy broadcast-domain-wide -auto-revert true
```

Create S3 LIFs and service policy

1. To create ONTAP S3 LIFs run the below commands.

```
network interface create -vserver OCP-SVM -lif s3-lif-01 -service-policy oai-data-s3 -home-node
aipod-a30-01 -home-port e4b-178 -address 10.61.178.51 -netmask 255.255.255.0 -status-admin up -
failover-policy broadcast-domain-wide -auto-revert true

network interface create -vserver OCP-SVM -lif s3-lif-02 -service-policy oai-data-s3 -home-node
aipod-a30-02 -home-port e4b-178 -address 10.61.178.52 -netmask 255.255.255.0 -status-admin up -
failover-policy broadcast-domain-wide -auto-revert true
```

2. Create service policy for S3.

```
set -privilege advanced

network interface service-policy create -vserver OCP-SVM -policy oai-data-s3 -services data-s3-
server, data-core -allowed-addresses 10.61.0.0/16

set -privilege admin
```

Note: The service policy is an advanced level setting in ONTAP.

Verify service policy

1. To verify the service policy applied on relevant LIFs, run the following commands:

For NFS

```
network interface show -vserver OCP-SVM -service-policy default-data-files
(network interface show)
```

Vserver	Logical Interface	Status Admin/Oper	Network Address/Mask	Current Node	Current Port	Is Home
OCP-SVM	nfs-lif-01	up/up	172.22.63.51/24	fpsa-aipod-a30-01	e4b-2263	true
	nfs-lif-02	up/up	172.22.63.52/24	fpsa-aipod-a30-02	e4b-2263	true

For S3

```
network interface show -vserver OCP-SVM -service-policy oai-data-s3
(network interface show)
```

Vserver	Logical Interface	Status Admin/Oper	Network Address/Mask	Current Node	Current Port	Is Home
OCP-SVM						

s3-lif-01	up/up	10.61.178.51/24	fpsa-aipod-a30-01 e4b-178 true
s3-lif-02	up/up	10.61.178.52/24	fpsa-aipod-a30-02 e4b-178 true

Create SVM Management LIF (Add Infrastructure SVM Administrator)

1. To add the SVM administrator and SVM administration LIF in the in-band management network, follow these steps:

```
network interface create -vserver OCP-SVM -lif svm-mgmt -service-policy default-management -home-  
node aipod-a30-01 -home-port e4b-2262 -address 172.22.62.50 -netmask 255.255.255.0 -status-admin  
up -failover-policy broadcast-domain-wide -auto-revert true
```

2. Create a default route that enables the SVM management interface to reach the outside world.

```
network route create -vserver OCP-SVM -destination 0.0.0.0/0 -gateway <svm-mgmt-gateway>
```

3. To verify, run the following commands:

```
network route show -vserver OCP-SVM
```

Vserver	Destination	Gateway	Metric
OCP-SVM	0.0.0.0/0	172.22.62.1	20

4. (Optional) Add route to reach the corresponding control/app network:

```
net route show -vserver OCP-SVM  
(network route show)
```

Vserver	Destination	Gateway	Metric
OCP-SVM	0.0.0.0/0	172.22.62.1	20

Note: A cluster serves data through at least one and possibly several SVMs. With these steps, you've created a single data SVM. You can create additional SVMs depending on your requirement.

Configure DNS

1. To configure DNS for the OCP-SVM run the following command:

```
dns create -vserver <vserver-name> -domains <dns-domain> -nameserver <dns-servers>
```

Example:

```
dns create -vserver OCP-SVM -domains FPMC.SA -name-servers 10.61.176.251,10.61.176.252
```

Configure AutoSupport™

1. To configure auto support, run the following command:

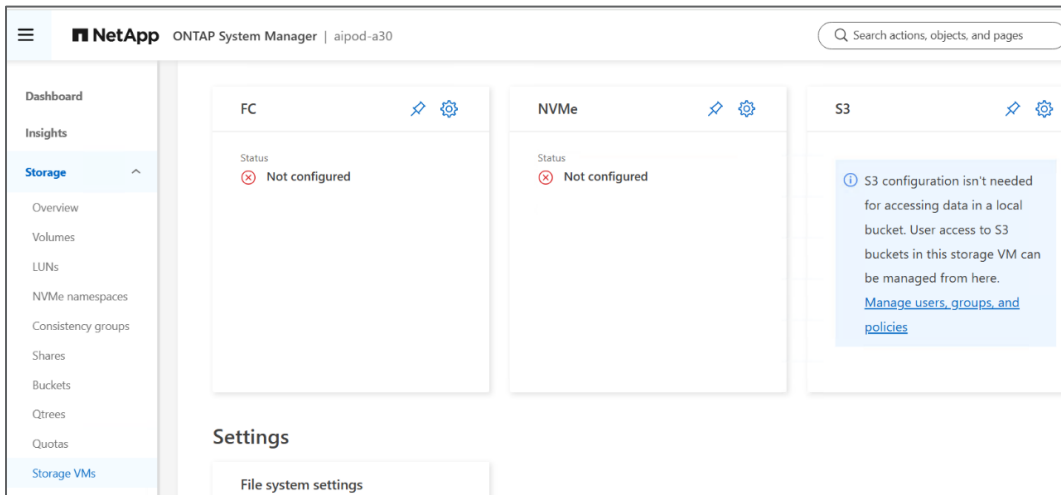
```
system node autosupport modify -state enable -mail-hosts <mailhost> -from <from-address> -  
transport https -support enable -to <storage-admin-email>
```

2. To test the Auto Support configuration by sending a message from all nodes of the cluster, run the following command:

```
autosupport invoke -node * -type all -message "ONTAP storage configuration for AIPOD is  
completed"
```

Configure ONTAP S3 and Bucket

1. To configure ONTAP S3 Bucket, Go to **Storage > Storage VMs** and click the SVM and go to **Settings**. Then Go to **Protocol** and click the setting icon to create the S3 server.



2. Enter the S3 server name and other details and click **Save**.

Configure S3

Server configuration

S3 server name

☒ Enable TLS

Port

Certificate

☒ Use system-generated certificate ?

Expiration Period

 Days ▼

☐ Use external-CA signed certificate

☒ Use HTTP (non-secure)

Port

Note: CA signed certificates can also be uploaded for the S3 bucket.

3. Go to **Storage > Buckets** and click **Add**. Enter the bucket name and click **Save**.

Add bucket

Name

model-repo

Capacity

500

GiB

☒ Enable ListBucket access for all users on the storage VM "OCP-SVM".
Enabling this will allow users to access the bucket.

↶ ↗ More options

Cancel

Save

- To configure S3 user and group, Go to **Cluster > Storage VMs**. Choose the OCP-SVM and go to Settings. Click on the edit buttons under S3 and add an User.

Add user

Name

rhoai

Key validity

365

 days

0

 hours

0

 minutes

0

 seconds

Note: A key will never expire if the validity value is set to 0.

Cancel

Save

- After creating User, go to **Groups** and create a group and add the newly created user and select the access policy from the list and Click **Save**.

Add group

Name

OAI

Users

rhoai ×

Policies

FullAccess ×

Cancel

Save

Single Node OpenShift configuration on Bare Metal

The section provides detailed procedures for the installation and configuration of Single Node OpenShift on Bare Metal. Single Node OpenShift will be used for hosting AI/ML workloads managed using Red Hat OpenShift AI serving as the MLOps platform for this solution. The OpenShift cluster is deployed from the cloud using Red Hat Hybrid Cloud Console using the recommended Red Hat Assisted Installer.

Prerequisites

- Network and Storage setup
- OpenShift requires the following components to be in place before the installation:
 - Installer workstation or machine for OpenShift cluster management. This installer provides CLI access to the cluster. More importantly, it provides secure SSH access to nodes in the cluster for post-deployment
 - To enable SSH Access to the OpenShift cluster, public keys must be provided to the OpenShift installer
- DHCP is used to provide IP address for OpenShift Bare Metal node on all interfaces for OpenShift Management, NFS, S3 and GPU (East-West only, when additional compute is added)
- DNS configuration – DNS entries for OpenShift.
 - Base Domain – This is the primary domain for your cluster
 - OpenShift Cluster Name
 - API Virtual
 - Ingress Load Balancer

Table 4) OpenShift parameters

Parameter	Value	Remark
IP subnet for OpenShift	10.61.177.0/24	
Gateway	10.61.177.1	
NTP	10.61.176.251 10.61.176.252	
DNS Server	10.61.176.251 10.61.176.252	
DHCP Server	10.61.176.251	

	10.61.176.252	
Red Hat OpenShift Cluster		
Base Domain	fpmc.sa	DNS Domain
Cluster Name	aipod-ocp	OpenShift Cluster Name
API VIP	api.aipod-ocp.fpmc.sa	10.61.177.124
Ingress VIP	*.apps.aipod-ocp.fpmc.sa	10.61.177.124
control node	control-1.aipod-ocp.fpmc.sa	10.61.177.124

Deployment Steps - Install Single Node OpenShift

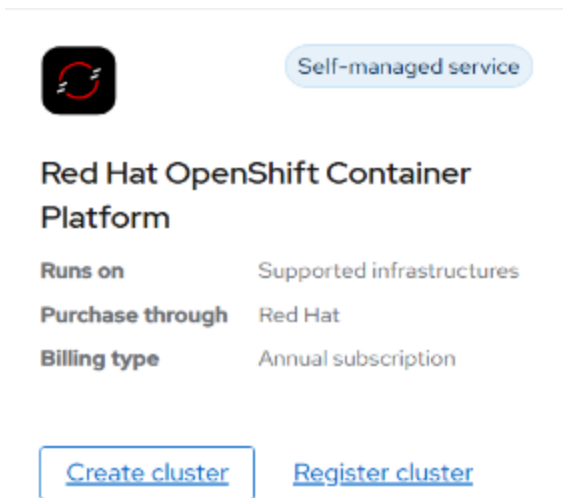
Note: For managing OpenShift cluster using CLI, a Linux workstation is deployed.

1. Login to workstation VM and create a new directory with the name aipod-ocp.
2. Generate RSA key which will be used during OpenShift installation:

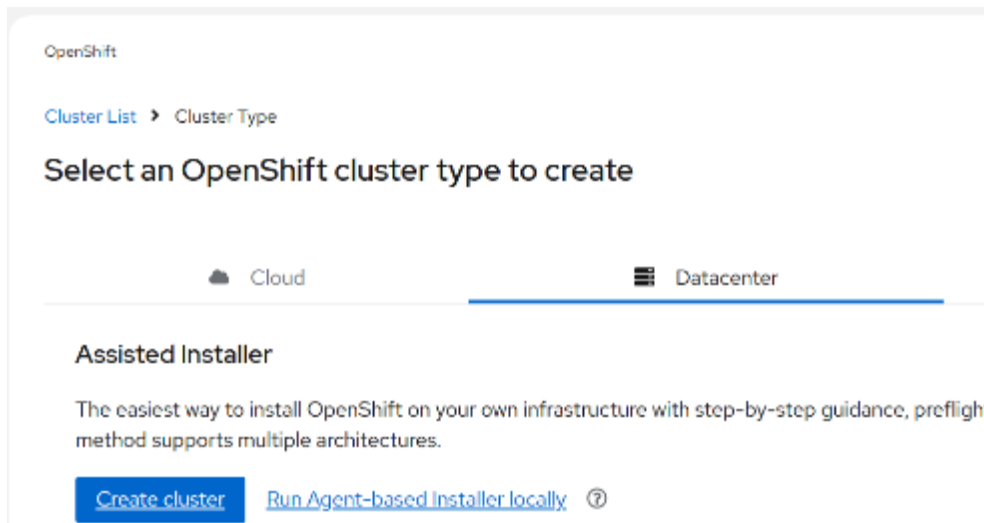
```
ssh-keygen -t rsa -N '' -f ~/.ssh/id_rsa
eval "$(ssh-agent -s)"
```

Install Single Node OpenShift

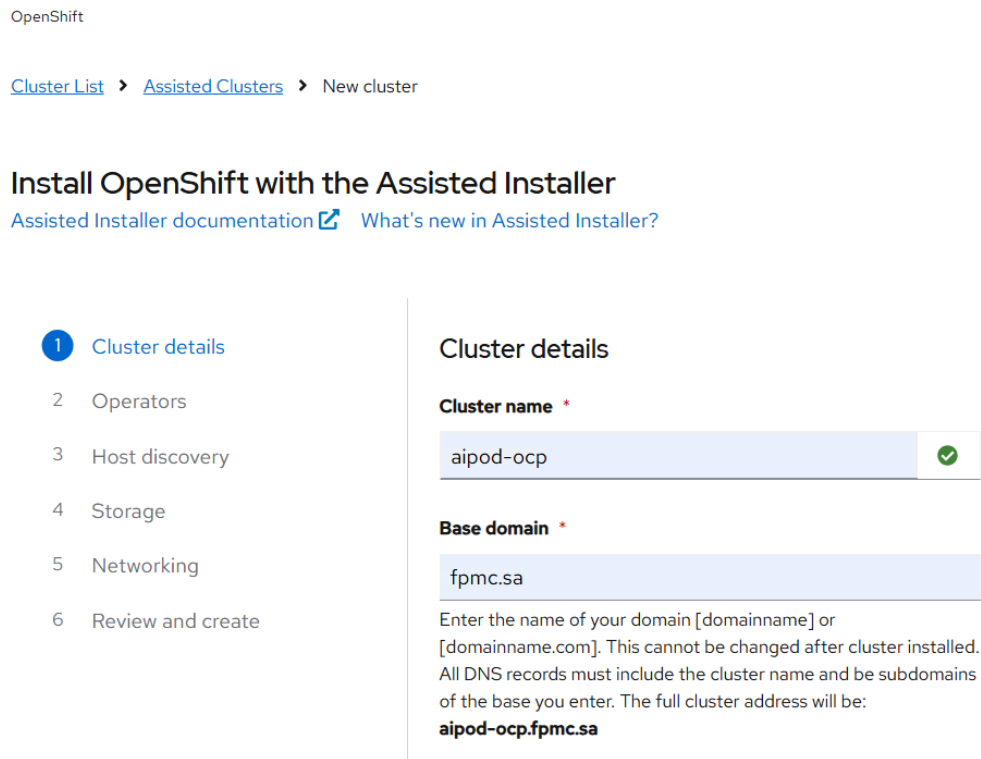
1. Open a web browser and go to <https://console.redhat.com> and login using your account.
2. Under **Services**, Go to **Container** and click on **Clusters**.
3. On the **Red Hat OpenShift Container Platform**, click on Create cluster.



4. Under **Datacenter** tab, choose the **Assisted Installer** and click on **Create cluster**.



5. Enter the Cluster details, which includes cluster name, base domain, OpenShift version and other details.



6. Select Single Node OpenShift and click **Next**.

No platform integration ▼

Number of control plane nodes ⓘ

1 (Single Node OpenShift) ▼

Limitations for using Single Node OpenShift

- Installing SNO will result in an OpenShift deployment that is not highly available.

☐ Include custom manifests ⓘ
Additional manifests will be applied at the install time for advanced configuration of the cluster.

Hosts' network configuration

☒ DHCP only ☐ Static IP, bridges, and bonds

Encryption of installation disks

☐ Control plane node, worker

Next Cancel

- Click **Add Hosts** under Host Discovery.

Install OpenShift with the Assisted Installer

[Assisted Installer documentation](#) [What's new in Assisted Installer?](#)

1

Cluster details

2

Operators

3

Host discovery

4

Storage

5

Networking

6

Review and create

Host discovery

Add host

☒ Run workloads on control plane nodes ⓘ

Information & Troubleshooting

[Minimum hardware requirements](#)
[Host not showing up?](#)

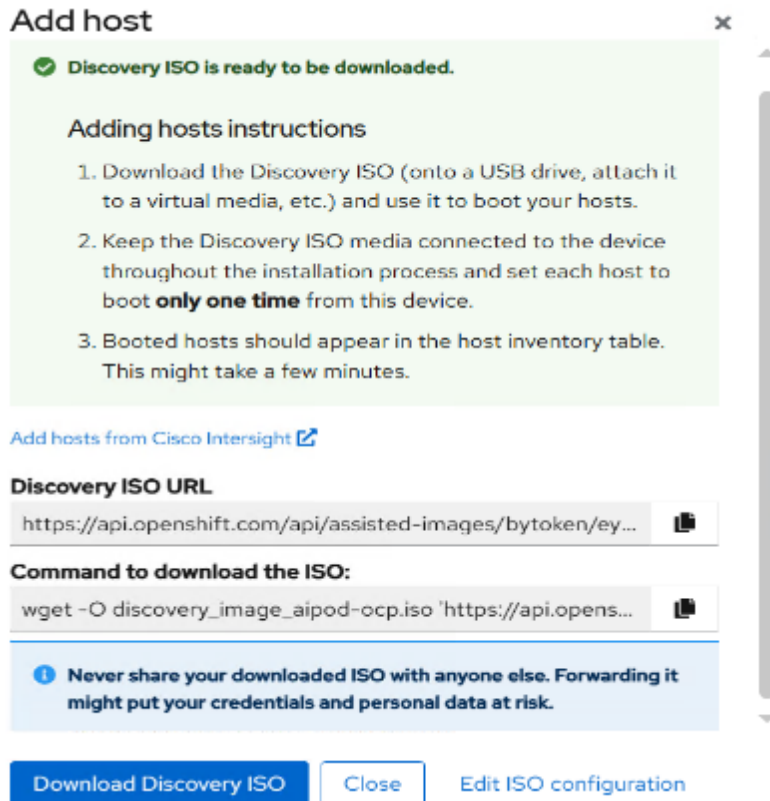
Host Inventory

Hostname	Role	Status	Discover	CPU	Memory
<div> <div></div> <div>Waiting for host...</div> <div> Hosts might take a few minutes to appear here after booting. Host not showing up? </div> </div>					

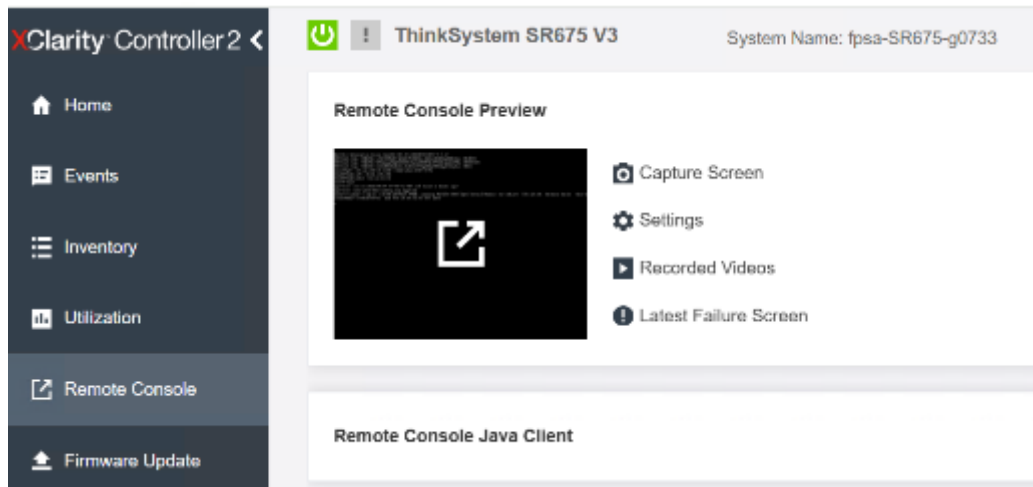
- For the provisioning type, choose “Minimal image file”

9. Copy and paste the ssh public key from the workstation which we created earlier, then click on **Generate Discovery ISO**.

10. Download the Discovery ISO to start the OpenShift Installation process.

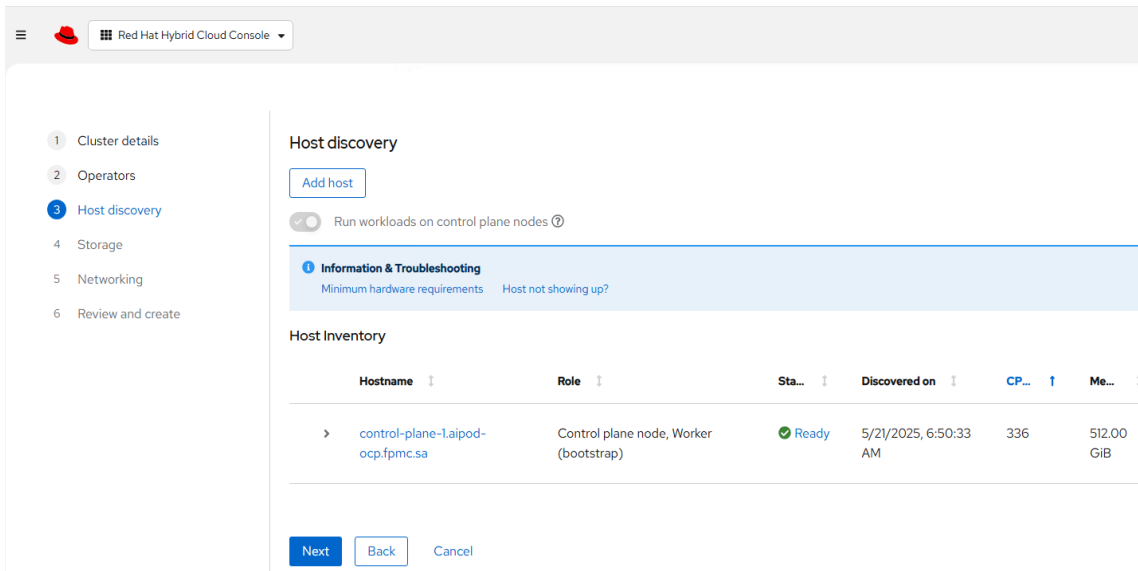


11. Login to Lenovo XClarity using BMC IP address of the server.
12. Under Remote Console, click on the black screen to Launch Remote Console.

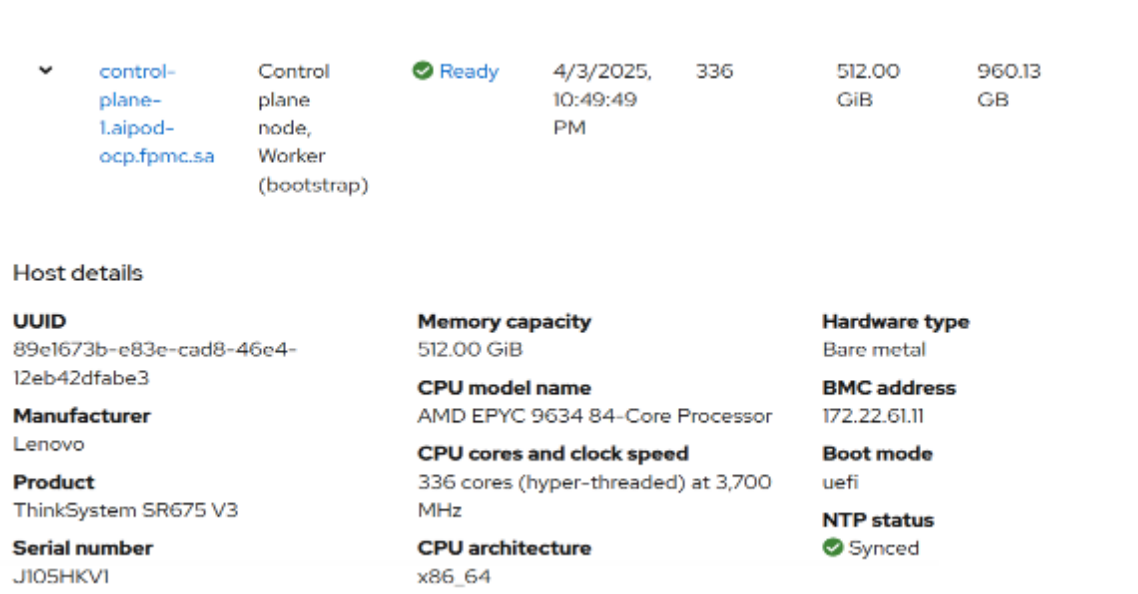


13. Under media, click **Activate > Browse**, select the iso and mount it. Scroll to the bottom and select the **Restart server immediately** from the drop-down list and click **Apply**.
14. During the startup, the server will try to boot using discovery iso image.

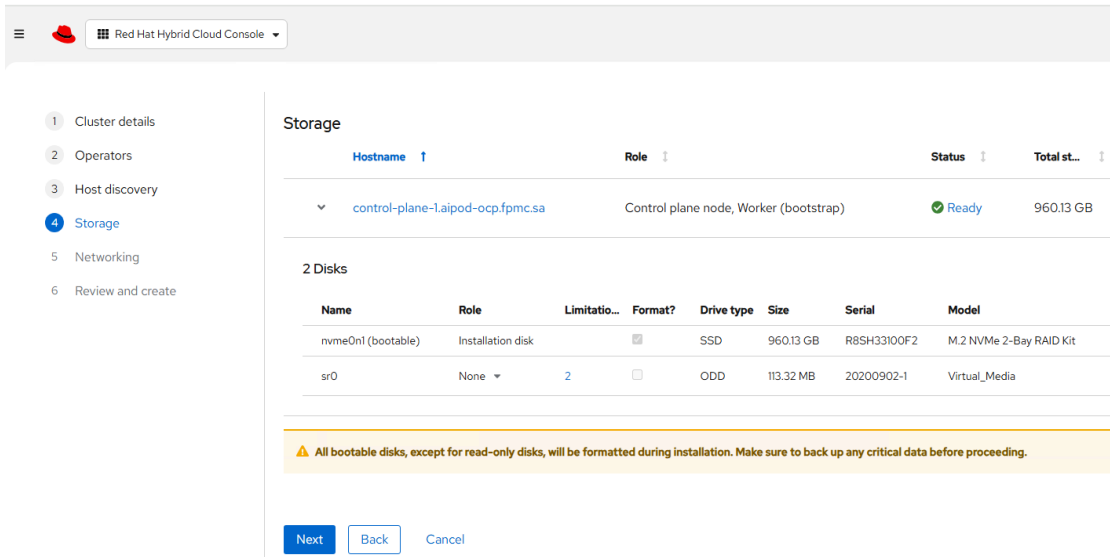
15. Once the server gets the IP address from DHCP, it will be shown under Red Hat hybrid console.



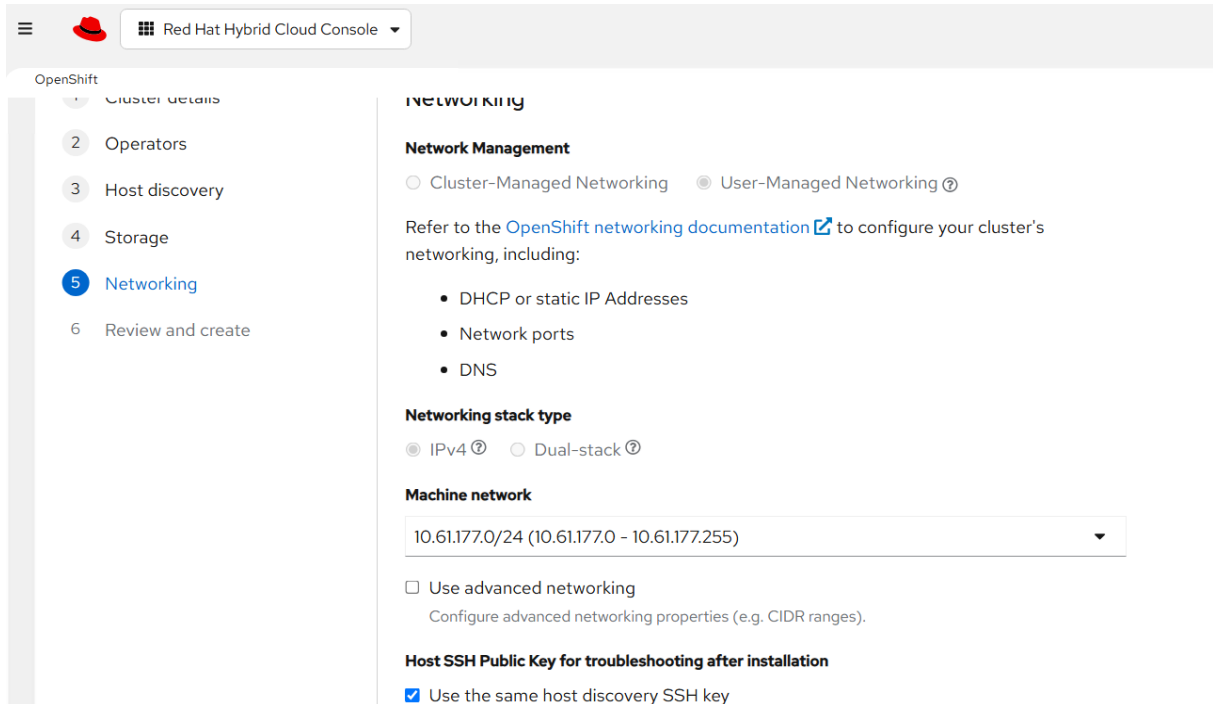
16. Under Host Inventory, expand the server to verify the NTP status.



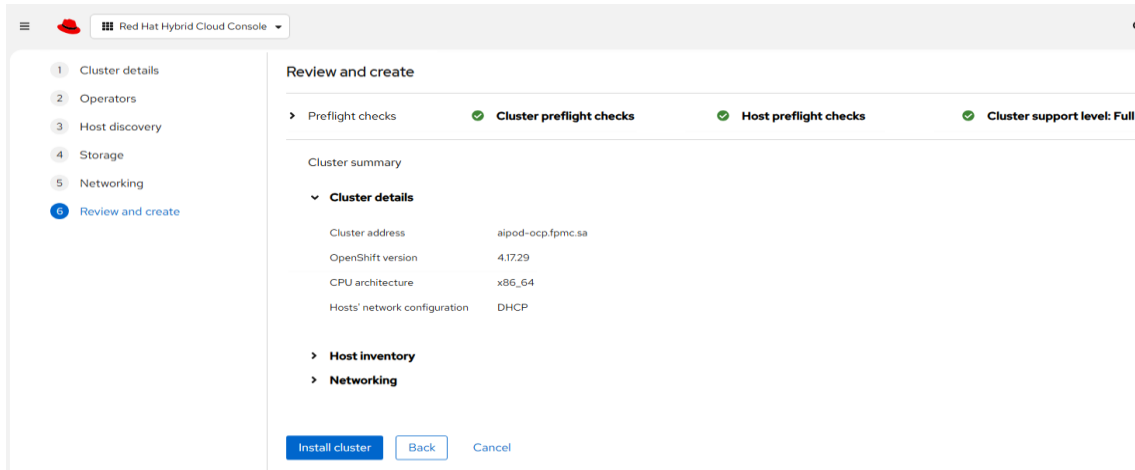
17. M.2 drives will be visible under **Storage**. Click **Next**.



18. The Network Management will be grayed out for SNO. Scroll down and click **Next**.



19. Review and click on **Install cluster**.



20. The installation process will start immediately, and progress can be seen.

Installation progress

Started on
5/21/2025, 6:57:08 AM

Preparing for installation 0%

Control Plane
Installing 1 control plane node

Initialization
Pending

[Abort installation](#)
[Download kubeconfig](#)
[View cluster events](#)

[Download Installation Logs](#)

Host inventory (1)

Hostname ↑	Role ↓	Status ↓	Discovered on ↓	CPU ... ↓	Memo... ↓	Tot... ↓
control-plane-1.aipod-ocp.fpmc.sa	Control plane node, Worker (bootstrap)	Preparing for installation	5/21/2025, 6:50:33 AM	336	512.00 GiB	960.13 GB

Deployment Steps – Post Installation

In this section we will discuss post installation steps.

1. After the installation completed successfully, download the kubeconfig and save it on the workstation.

OpenShift
aipod-ocp

Installation progress

Started on
5/21/2025, 6:57:08 AM

Installed on 5/21/2025, 7:54:30 AM

Control Plane
1 control plane node installed

Initialization
Completed

Installation completed successfully

[Launch OpenShift Console](#)
[Download kubeconfig](#)
[View cluster events](#)

[Download Installation Logs](#)

2. Launch OpenShift Web console and use the Username and Password to login.

Web Console URL
<https://console-openshift-console.apps.aipod-ocp.fpmc.sa>

❗ Not able to access the Web Console?

Username
kubeadmin

Password
.....

Download and save your kubeconfig file in a safe place. This file will be automatically deleted from Assisted Installer's service in 20 days.

Add new hosts by generating a new Discovery ISO under your cluster's "Add hosts" tab on console.redhat.com/openshift.

3. After logging in, click the ? icon and choose **Command Line Tools** from the drop-down list.

4. Click **Download oc for Linux for x86_64**. Copy the file to the Linux workstation.

Command Line Tools

[Copy login command](#)

oc - OpenShift Command Line Interface (CLI)

With the OpenShift command line interface, you can create applications and manage OpenShift projects from a terminal.

The oc binary offers the same capabilities as the kubectl binary, but it is further extended to natively support OpenShift Container Platform features.

- [Download oc for Linux for x86_64](#)
- [Download oc for Mac for x86_64](#)
- [Download oc for Windows for x86_64](#)
- [Download oc for Linux for ARM 64](#)
- [Download oc for Mac for ARM 64](#)
- [Download oc for Linux for IBM Power, little endian](#)
- [Download oc for Linux for IBM Z](#)
- [LICENSE](#)

5. Run the commands below to install oc client on the Linux workstation.

```
[root@workstation-rhel tool~]# tar xvf oc.tar
oc
[root@workstation-rhel tool~]# ls
oc oc.tar
[root@workstation-rhel tool~]# mv oc /usr/local/bin

[root@workstation-rhel tool~]# oc get node
NAME                                STATUS    ROLES                                AGE      VERSION
control-plane-1.aipod-ocp.fpmc.sa  Ready    control-plane,master,worker         2d22h    v1.30.10
```

6. Configure NTP on the node. On the workstation, create a machine-configs directory and download butane.

```
mkdir machine-configs
cd machine-configs
curl https://mirror.openshift.com/pub/openshift-v4/clients/butane/latest/butane --output butane
chmod +x butane
```

7. Create the file for the OpenShift Node.

```
cat 99-control-plane-chrony-conf-override.bu

variant: openshift

version: 4.17.0

metadata:
  name: 99-control-plane-chrony-conf-override
  labels:
    machineconfiguration.openshift.io/role: master

storage:
  files:
    - path: /etc/chrony.conf
      mode: 0644
      overwrite: true
      contents:
        inline: |
          driftfile /var/lib/chrony/drift
          makestep 1.0 3
          rtcsync
          logdir /var/log/chrony
          server 10.61.176.251 iburst
          server 10.61.176.252 iburst
```

8. Convert into yaml and apply to the OpenShift cluster.

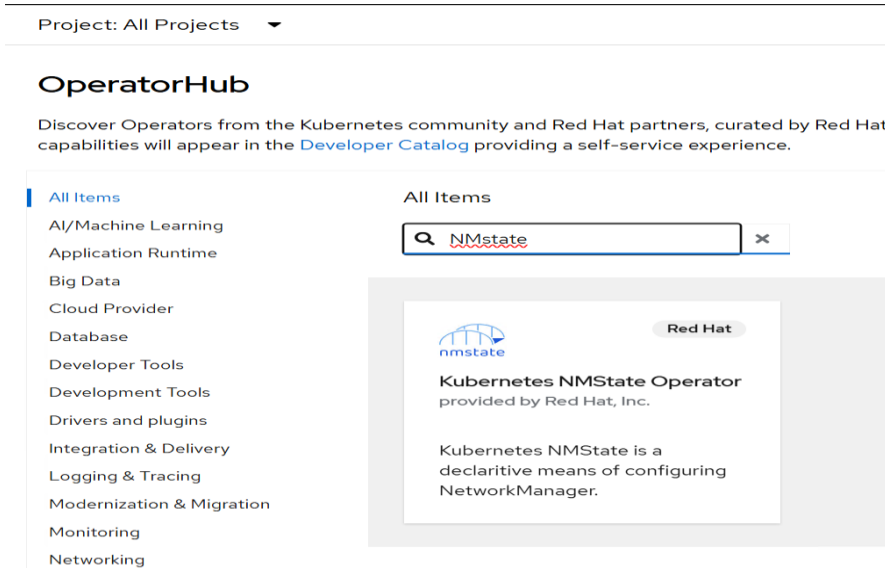
```
./butane 99-control-plane-chrony-conf-override.bu -o ./99-control-plane-chrony-conf-override.yaml

oc create -f 99-control-plane-chrony-conf-override.yaml
```

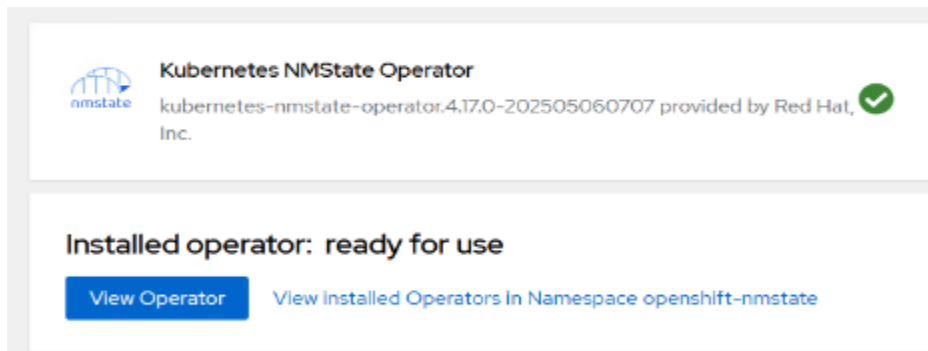
Deployment Steps – Configure Storage interfaces on the Node

To configure NFS and S3 storage interfaces on the node, Kubernetes NMState Operator will be used.

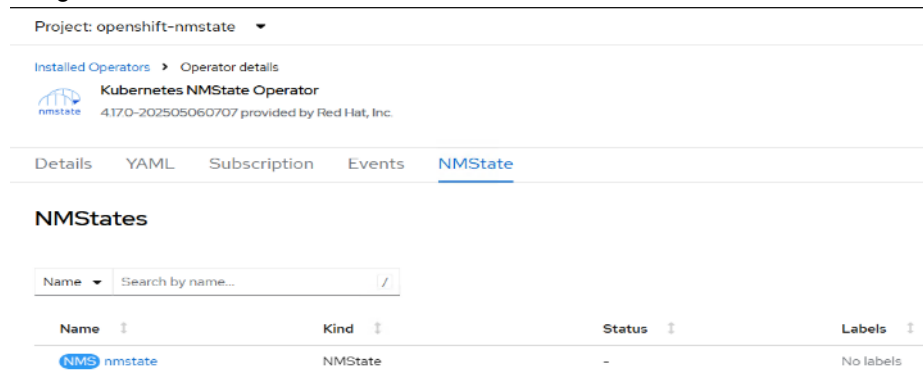
1. In the OpenShift console, go to **Operators > OperatorHub** and search with the name NMstate. Click Kubernetes NMState Operator.



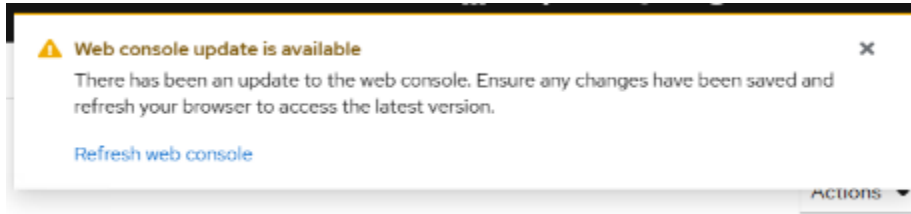
2. Click **Install**, keep everything default and click **Install** again. The operator will be installed in a new namespace.



3. Click **View Operator**, select **NMState** tab and on the right, click **Create NMState**. Leave the default settings and click **Create**.



4. Refresh the web console.



5. On the Linux workstation, create the following files.

Note: NFS and S3 VLAN interfaces were created on a single physical interface with MTU 9000.

```
##For MTU

cat ens20f0np0.yaml
apiVersion: nmstate.io/v1

kind: NodeNetworkConfigurationPolicy

metadata:
  name: ens20-mtu

spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ''

  desiredState:
    interfaces:
      - name: ens20f0np0
        description: Configuring ens20f0np0 on node
        type: ethernet
        state: up
        mtu: 9000

##For NFS

cat ens20-nfs.yaml
apiVersion: nmstate.io/v1

kind: NodeNetworkConfigurationPolicy

metadata:
  name: ocp-nfs-policy

spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""

  desiredState:
    interfaces:
      - name: ens20f0np0.2263
        description: Configuring ens20f0np0 on node
        type: vlan
        state: up
```

```

    ipv4:
      dhcp: true
      enabled: true
    ipv6:
      enabled: false
  vlan:
    base-iface: ens20f0np0
    id: 2263

##For S3
cat ens20-s3.yaml
apiVersion: nmstate.io/v1

kind: NodeNetworkConfigurationPolicy
metadata:
  name: ocp-s3-policy
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""
  desiredState:
    interfaces:
    - name: ens20f0np0.178
      description: Configuring ens20f0np0 on node
      type: vlan
      state: up
      ipv4:
        dhcp: true
        enabled: true
      ipv6:
        enabled: false
      vlan:
        base-iface: ens20f0np0
        id: 178

```

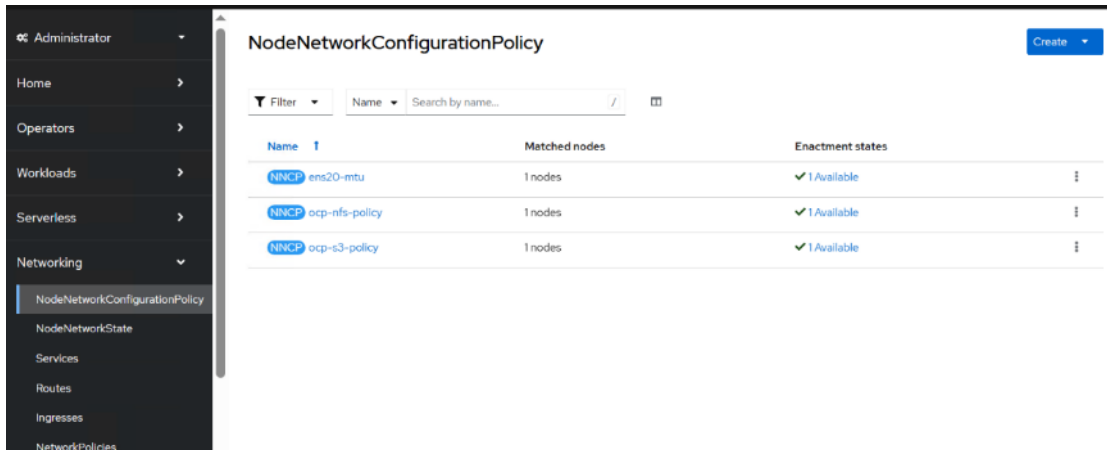
6. To add the Node Network Configuration Policies, apply the yaml files created.

```

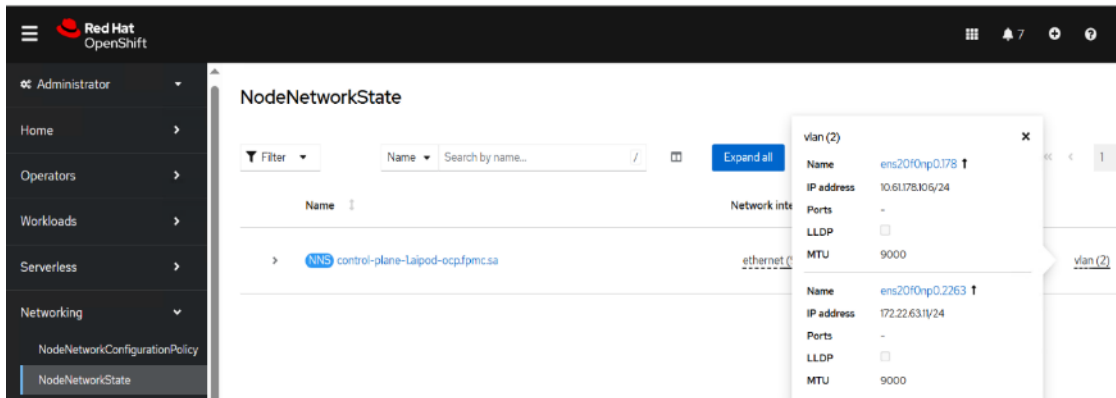
oc create -f ens20f0np0.yaml
oc create -f ens20-nfs.yaml
oc create -f ens20-s3.yaml

```

- The policy applied should be visible under **Networking > NodeNetworkConfigurationPolicy**.



- Both NFS and S3 VLAN interfaces will get IP addresses from DHCP.

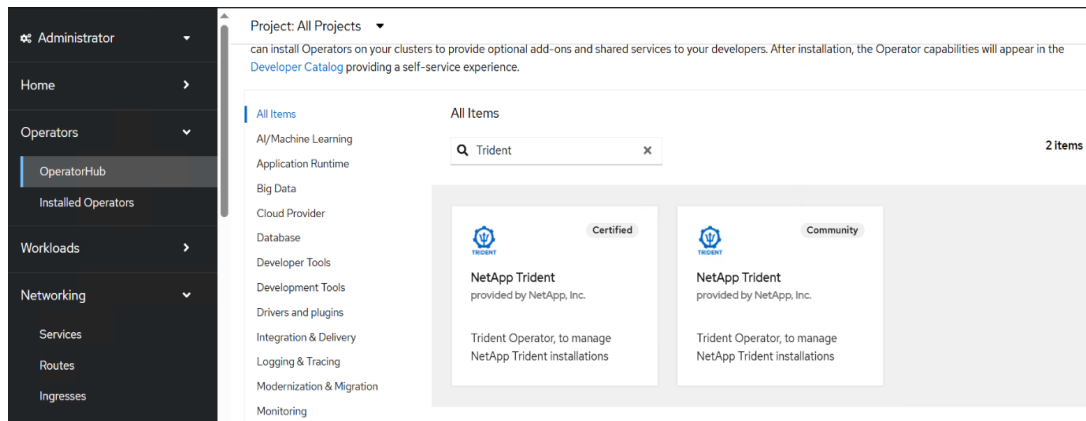


Install NetApp Trident

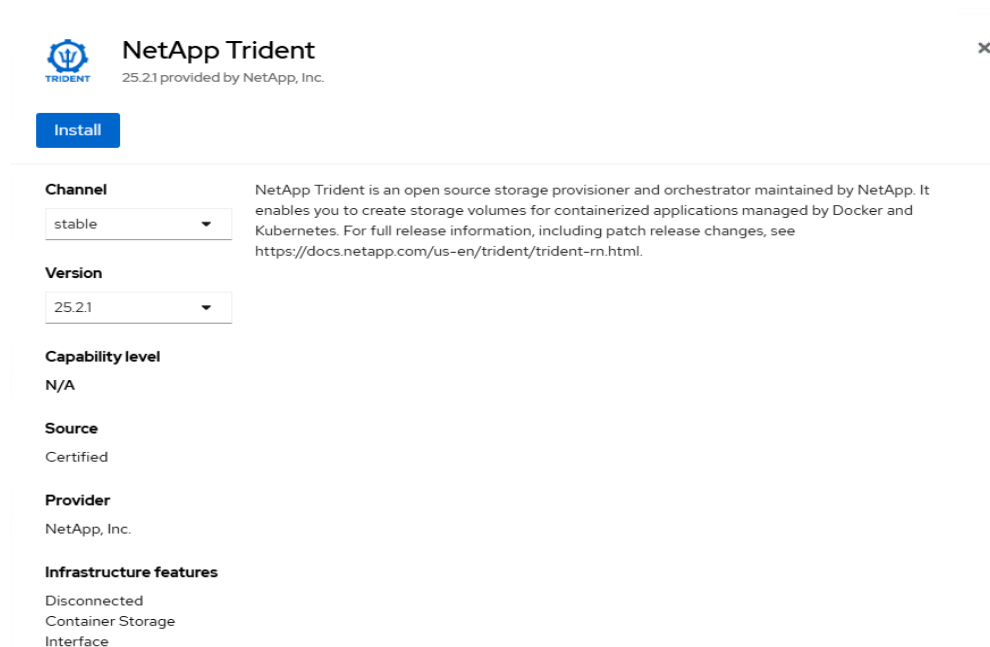
NetApp Trident is an open-source, fully supported storage orchestrator for containers and Kubernetes distributions. It was designed to help meet the containerized applications' persistence demands using industry-standard interfaces, such as the Container Storage Interface (CSI). With Trident, microservices and containerized applications can take advantage of enterprise-class storage services provided by the NetApp portfolio of storage systems. More information about Trident can be found here: [NetApp Trident Documentation](#). There are various methods to install NetApp Trident. In this solution, we will cover the installation of NetApp Trident version 25.2.1 using the Trident Operator, which is installed using OperatorHub.

- In the OCP console and create a namespace with the name **trident**.

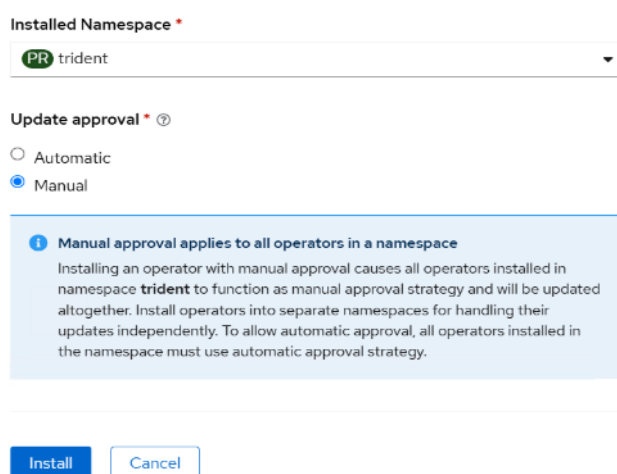
2. Go to **Operators > OperatorHub** and type Trident in the filter box. Click on Certified NetApp Trident.



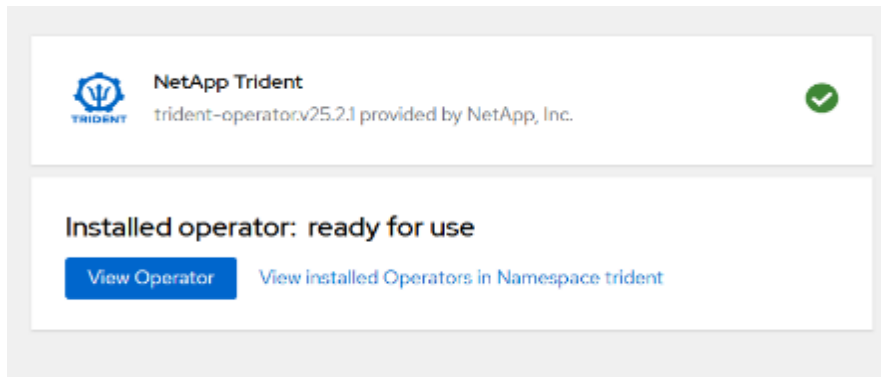
3. Click **Install**.



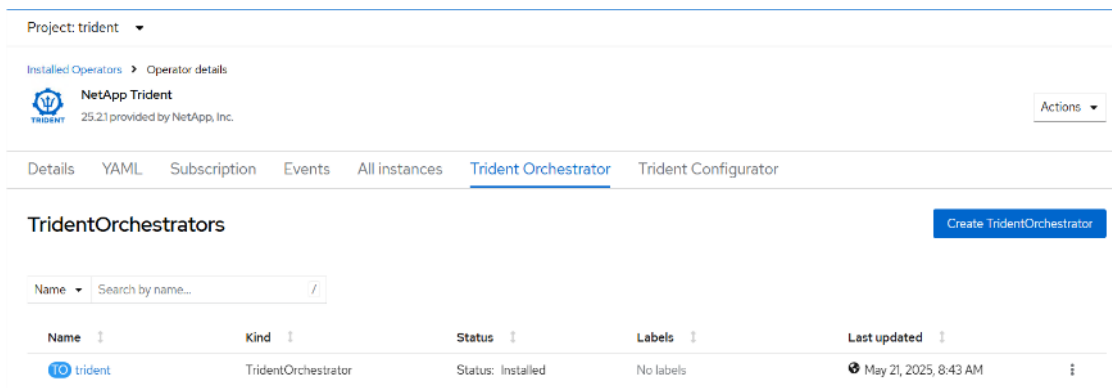
4. Select Trident namespace and click **Install**.



- Once the installation is completed, click **View Operator**.



- Under Trident Orchestrator tab, on the right side click Create **TridentOrchestrator**. Leave everything default and click Create.



- On the Linux workstation, check the pod status.

```
oc get pod -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-controller-84cb9bff89-tljcc	6/6	Running	0	65s
trident-node-linux-tnjsp	2/2	Running	0	65s
trident-operator-78b985dbd9-dmg5b	1/1	Running	0	112s

- Create the backend yaml files.

Note: In this solution we created two backend types, Standard NFS and FlexGroup NFS.

```
cat backend_nfs.yaml
---
version: 1
storageDriverName: ontap-nas
backendName: ocp-nfs-backend
managementLIF: 172.22.61.40
dataLIF: 172.22.63.51
svm: OCP-SVM
username: admin
password: *****
userREST: true
defaults:
  spaceReserve: none
  exportPolicy: default
  snapshotPolicy: default
  snapshotReserve: '10'

cat backend_nfs_flexgrp.yaml
---
version: 1
storageDriverName: ontap-nas-flexgroup
```

```

backendName: ocp-nfs-flexgroup
managementLIF: 172.22.61.40
dataLIF: 172.22.63.52
svm: OCP-SVM
username: admin
password: *****
useREST: true
defaults:
  spaceReserve: none
  exportPolicy: default
  snapshotPolicy: default
  snapshotReserve: '10'

```

9. Create the standard NFS backend.

```

[root@abhinav-rhel ~]# tridentctl create backend -f backend_nfs.yaml -n trident
+-----+-----+-----+-----+-----+-----+
+-----+
|      NAME      | STORAGE DRIVER |          UUID          | STATE | USER-STATE |
+-----+-----+-----+-----+-----+-----+
| ocp-nfs-backend | ontap-nas      | a41299de-3479-458f-a5a7-60548185a00f | online | normal      |
+-----+-----+-----+-----+-----+-----+
0 |
+-----+-----+-----+-----+-----+-----+
+-----+

```

10. Create the FlexGroup NFS backend.

```

[root@abhinav-rhel ~]# tridentctl create backend -f backend_nfs_flexgrp.yaml -n trident
+-----+-----+-----+-----+-----+-----+
+-----+
|      NAME      | STORAGE DRIVER |          UUID          | STATE | USER-STATE |
+-----+-----+-----+-----+-----+-----+
| ocp-nfs-flexgroup | ontap-nas-flexgroup | a4637e95-5fba-4b2f-b1e1-7f9644da01af | online | normal      |
+-----+-----+-----+-----+-----+-----+
0 |
+-----+-----+-----+-----+-----+-----+
+-----+

```

11. Create storage class for both backend types.

```

cat sc-ontap-nfs.yaml
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nfs
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"

```

```
allowVolumeExpansion: true
```

```
cat sc-ontap-nfs-flexgroup.yaml
```

```
---
```

```
apiVersion: storage.k8s.io/v1
```

```
kind: StorageClass
```

```
metadata:
```

```
  name: ontap-nfs-flexgroup
```

```
  annotations:
```

```
    storageclass.kubernetes.io/is-default-class: "false"
```

```
provisioner: csi.trident.netapp.io
```

```
parameters:
```

```
  backendType: "ontap-nas-flexgroup"
```

```
  provisioningType: "thin"
```

```
  snapshots: "true"
```

```
allowVolumeExpansion: true
```

12. Create the following storage classes.

```
oc create -f sc-ontap-nfs.yaml
```

```
storageclass.storage.k8s.io/ontap-nfs created
```

```
oc create -f sc-ontap-nfs-flexgroup.yaml
```

```
storageclass.storage.k8s.io/ontap-nfs-flexgroup created
```

```
oc get sc
```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION	AGE		
ontap-nfs (default)	csi.trident.netapp.io	Delete	Immediate
8s			true
ontap-nfs-flexgroup	csi.trident.netapp.io	Delete	Immediate
2s			true

13. Enable auto support for Trident.

```
[root@abhinav-rhel ~]# tridentctl send autosupport -n trident
```

```
Please see NetApp's privacy policy at https://www.netapp.com/company/legal/privacy-policy/
```

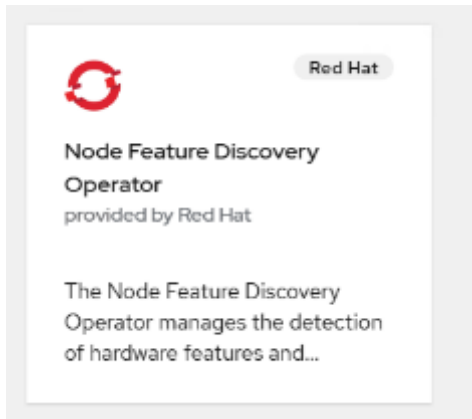
```
Do you authorize NetApp to collect personal information for the exclusive purpose of providing support services? [y/n]: y
```

```
Autosupport sent.
```

Installing NVIDIA GPU Operator

Note: To deploy NVIDIA's GPU Operator in Red Hat OpenShift, Red Hat's Node Feature Discovery (NFD) Operator must first be deployed.

1. Log into OpenShift web console, go to **Operator > OperatorHub** and search for "Node Feature Discovery".



2. Click Node Feature Discovery Operator provided by Red Hat, (Keep everything default) click **Install**.

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

stable

Version *

4.17.0-202505061137

Installation mode *

☐ All namespaces on the cluster (default)
This mode is not supported by this Operator

☒ A specific namespace on the cluster
Operator will be available in a single Namespace only.

Installed Namespace *

☒ Operator recommended Namespace: **PR** openshift-nfd

☐ Select a Namespace

Namespace creation
Namespace **openshift-nfd** does not exist and will be created.

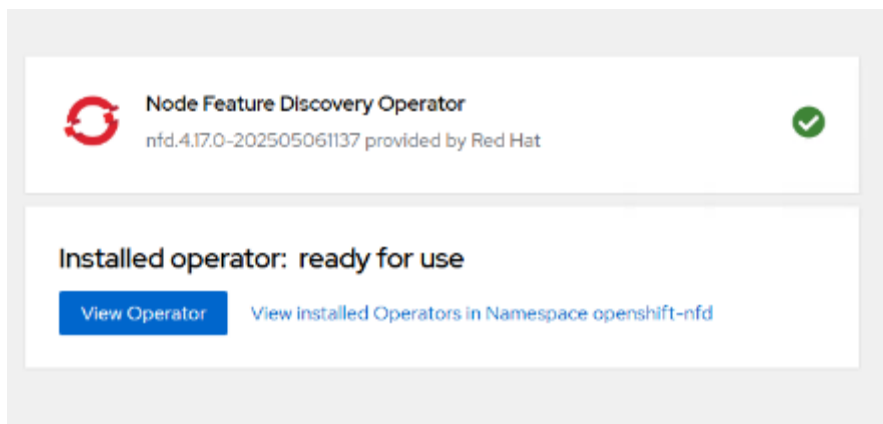
Node Feature Discovery Operator
provided by Red Hat

Provided APIs

NFD NodeFeatureDiscovery
The NodeFeatureDiscovery instance is the CustomResource being watched by the NFD-Operator, and holds all the needed information to setup the behaviour of the master and worker pods

NFR NodeFeatureRule
NodeFeatureRule resource specifies a configuration for feature-based customization of node objects, such as node labeling.


3. Click **View Operator**.



4. From the tab, go to **NodeFeatureDiscovery** and click **Create NodeFeatureDiscovery** (keep everything default).

Project: openshift-nfd

Installed Operators > Operator details


 **Node Feature Discovery Operator**
4.17.0-202505061137 provided by Red Hat

Actions

Details | YAML | Subscription | Events | All instances | NodeFeatureDiscovery | NodeFeatureGroup | NodeFeatureRule | NodeFeature

NodeFeatureDiscoveries Create NodeFeatureDiscovery

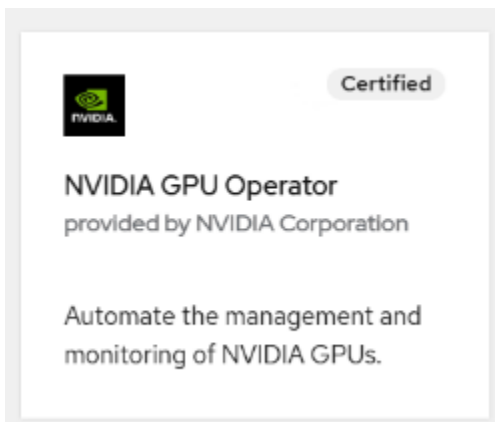
Name Search by name...


Name	Kind	Status	Labels	Last updated
 nfd-instance	NodeFeatureDiscovery	Conditions: Available, Upgradeable	No labels	May 21, 2025, 8:58 AM

- From the Linux workstation run the below command to confirm the label.

```
oc get nodes -l feature.node.kubernetes.io/pci-10de.present
```

- Again, go to **Operator > OperatorHub** and enter NVIDIA GPU Operator in the search bar.

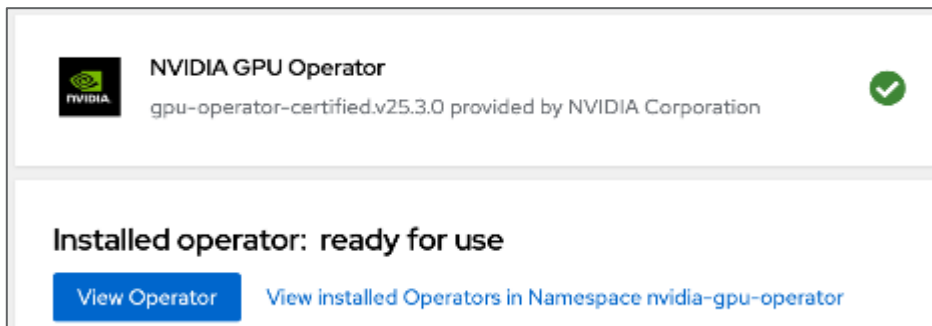




 **Certified**

NVIDIA GPU Operator
provided by NVIDIA Corporation

Automate the management and monitoring of NVIDIA GPUs.

- Select NVIDIA GPU Operator and click **Install**.
- Leave everything default and click **Install**.



 **NVIDIA GPU Operator**
gpu-operator-certified.v25.3.0 provided by NVIDIA Corporation 

Installed operator: ready for use

[View Operator](#) [View installed Operators in Namespace nvidia-gpu-operator](#)

- Click View Operator and create cluster policy under **ClusterPolicy**. Leave the default values and click **Create**.

Project: nvidia-gpu-operator ▼

Create ClusterPolicy

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: ☒ Form view ☐ YAML view

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *

gpu-cluster-policy

Labels

app=frontend

10. The policy status will be ready

Project: nvidia-gpu-operator ▼

[Installed Operators](#) > Operator details

NVIDIA GPU Operator

25.3.0 provided by NVIDIA Corporation

[Details](#)
[YAML](#)
[Subscription](#)
[Events](#)
[All instances](#)
[ClusterPolicy](#)
[NVIDIADriver](#)

All Instances

Filter ▼

Name ▼ Search by name... /

Name	Kind	Status	Labels
gpu-cluster-policy	ClusterPolicy	State: ready	No labels

11. Verify the pods status.

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

NAME                                1/1    Running    0          7m35s
gpu-feature-discovery-87tpq         1/1    Running    0          12m
gpu-operator-855f574c7d-bd58w       1/1    Running    0          7m35s
nvidia-container-toolkit-daemonset-x77s4 1/1    Running    0          7m35s
nvidia-cuda-validator-lctmb         0/1    Completed  0          91s
nvidia-dcgm-exporter-lrnlf          1/1    Running    0          7m35s
nvidia-dcgm-njwvx                   1/1    Running    0          7m35s
nvidia-device-plugin-daemonset-7slwv  1/1    Running    0          7m35s
nvidia-driver-daemonset-417.94.202505062152-0-7ctt8 2/2    Running    0          8m24s
nvidia-node-status-exporter-tqhvz    1/1    Running    0          8m22s
nvidia-operator-validator-fbzbfb     1/1    Running    0          7m35s
```

12. Connect to “nvidia-driver-daemonset” container and verify the GPU.

```
#nvidia-driver-daemonset-417.94.202505062152-0-7ctt8

#nvidia-smi
```

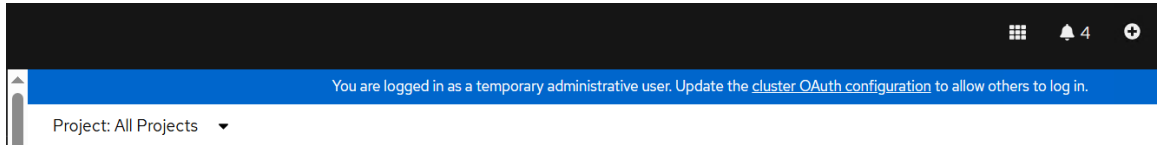
Adding Administrative User to the OpenShift.

Note: An administrative user needs to be added to OpenShift cluster, default **kubeadmin** does not have administrative privileges on OpenShift AI.

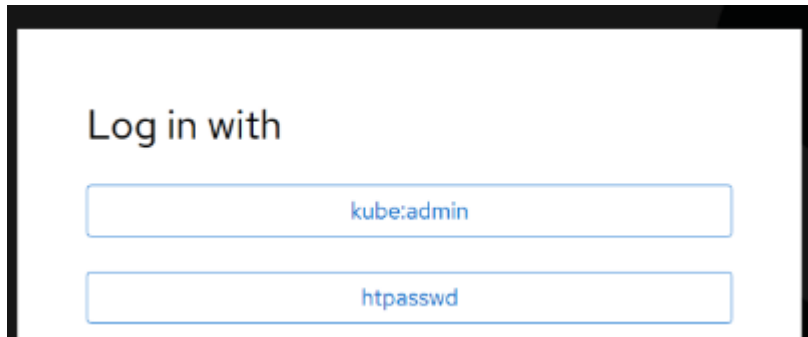
1. Login to Linux workstation and go to ocp directory and run the command below.

```
htpasswd -c -B -b ./admin.htpasswd admin <your_password>
cat admin.htpasswd
```

2. On the OpenShift web console, **cluster OAuth configuration** message will appear. Click on the link.



3. In the next window, for **IDP**, choose **HTPasswd** from the drop-down menu.
4. In the **Add Identify Provider:HTPasswd** screen, paste the contents of the **admin.htpasswd** file
And click **Add**.
5. Wait for some time and try to login to OpenShift web console. Click **htpasswd** and enter the user and password.



6. You will see basic access to the OpenShift cluster, logout now and again login using **kubeadmin** credential.
7. Go to User **Management** > **Users** and Choose the user that was previously created using htpasswd. Click the username.
8. In the **User** > **User Details** window, go to the **RoleBindings** tab, Click **Create binding**.
9. In the Create **Rolebinding** window, click **Cluster-wide** role binding (ClusterRoleBinding).

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#)

Create RoleBinding

Associate a user/group to the selected role to define the type of access and resources that are allowed.

Binding type

☐ Namespace role binding (RoleBinding)
Grant the permissions to a user or set of users within the selected namespace.

☒ Cluster-wide role binding (ClusterRoleBinding)
Grant the permissions to a user or set of users at the cluster level and in all namespaces.


RoleBinding

Name *

ocp-admin

Role

Role name *

 cluster-admin

- Specify a name, such as mlops-admin and for Role Name, choose **cluster-admin** from the drop-down list and click **Create**.
- Logout and login into the OpenShift web console. You will now have an administrator access, the same as kubectl.

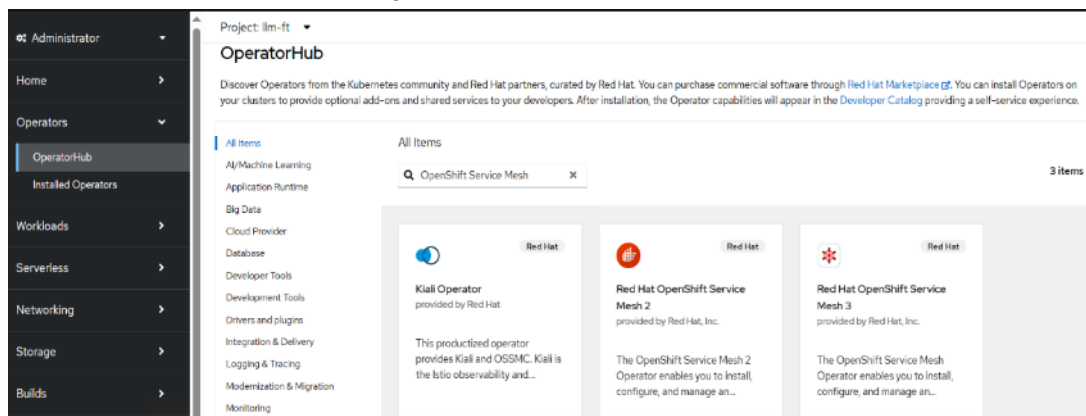
Installing Red Hat OpenShift AI Self-Managed

Red Hat® OpenShift® AI is a platform for managing the lifecycle of predictive and generative AI (gen AI) models, at scale, across hybrid cloud environments. Refer to [requirements](#) for OpenShift AI Self-Managed.

To support KServe for single model with Red Hat OpenShift AI, we need to deploy **OpenShift Service Mesh Operator**, **OpenShift Serverless Operator** and **Red Hat Authorino Operator**.

Deploy Red Hat OpenShift Service Mesh Operator

- Log into Red Hat OpenShift web console, go to **Operators > OperatorHub** and search for **OpenShift Service Mesh**. Click **Red Hat OpenShift Service Mesh 2**.



Note: OpenShift Service Mesh 2 is only supported at the time of documentation.

- Click **Install**, leave everything default and click **Install**.

3. Click **View Operator**.

Deploy Red Hat OpenShift Serverless Operator

1. Go to **Operators > OperatorHub** and search for **OpenShift Serverless**.
2. Click Red Hat OpenShift Serverless.

3. Keep the settings default and click **Install**. The operator will be deployed in a new **openshift-serverless** namespace.

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel * ⓘ

stable

Version *

1.35.1

Installation mode *

☒ All namespaces on the cluster (default)
Operator will be available in all Namespaces.

☐ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *

☒ Operator recommended Namespace: **PR** openshift-serverless

☐ Select a Namespace

Namespace creation
Namespace **openshift-serverless** does not exist and will be created.

Update approval * ⓘ

☒ Automatic

☐ Manual

Install **Cancel**

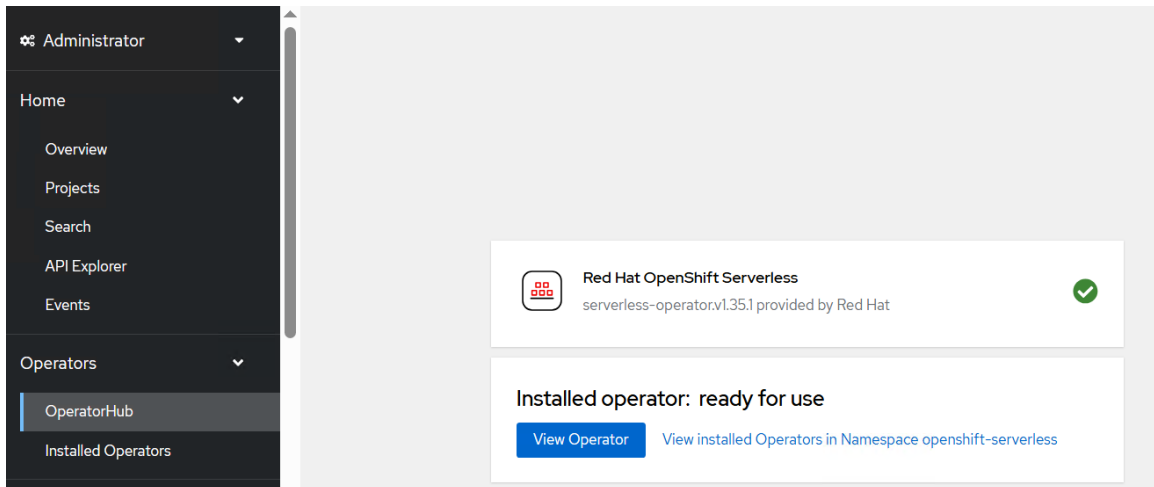
Red Hat OpenShift Serverless
provided by Red Hat

Provided APIs

KS Knative Serving
A platform for streamlined application deployment, traffic-based auto-scaling from zero to N, and traffic-split rollouts

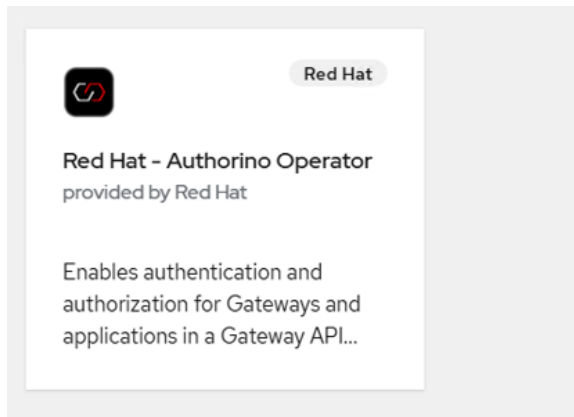
KK Knative Kafka
An extension to Knative Eventing, merging HTTP accessibility with Apache Kafka's proven efficiency and reliability

4. The operator will be deployed successfully.



Deploy Red Hat Authorino

1. Go to **Operators > OperatorHub** and search for **Authorino**.



2. Keep the settings default and click **Install**.

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel * ⓘ

stable

Version *

1.2.1

Installation mode *

☒ All namespaces on the cluster (default)
Operator will be available in all Namespaces.

☐ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *

openshift-operators

Update approval * ⓘ

☒ Automatic

☐ Manual

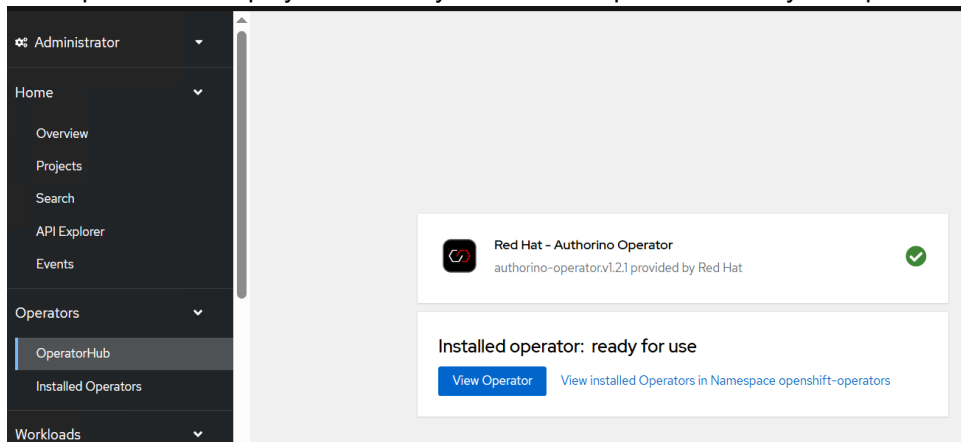
Red Hat - Authorino Operator
provided by Red Hat

Provided APIs

- AuthConfig**
API to describe the desired protection for a service
- Authorino**
API to create instances of authorino

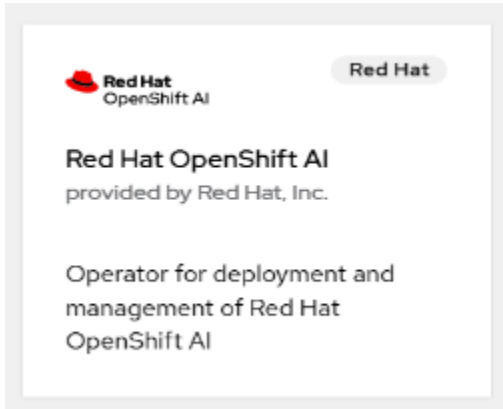
Install **Cancel**

3. The operator will deploy successfully. Click View Operator to verify the operator.

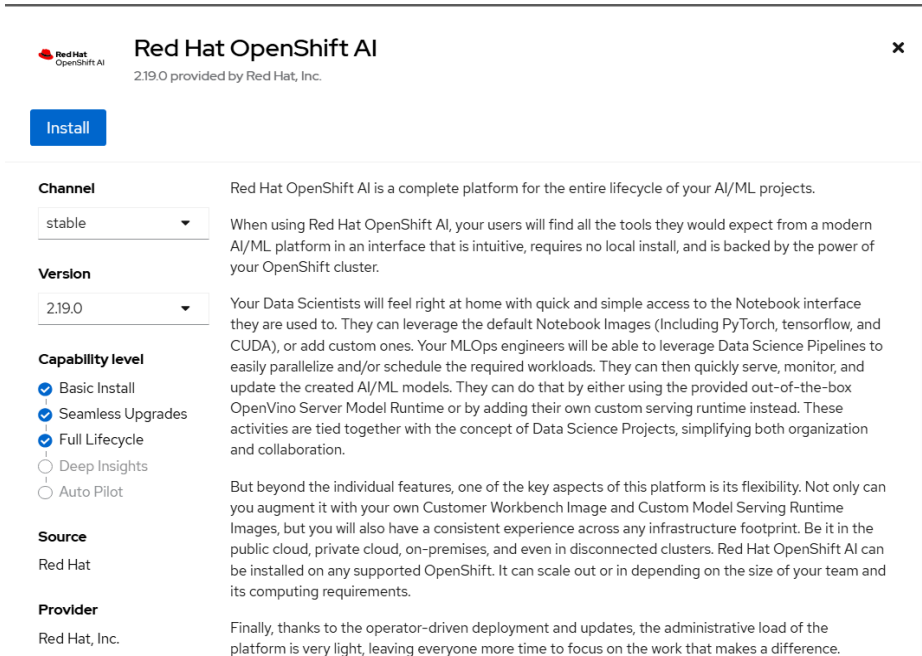


Deploy Red Hat OpenShift AI Operator

1. Go to **Operators > OperatorHub** and search for **Red Hat OpenShift AI**. Choose **Red Hat OpenShift AI**.



2. Select Version 2.19.0 and click **Install**.



3. Keep the default settings and click **Install**.

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel * ⓘ
stable

Version *
2.19.0


Installation mode *
☒ All namespaces on the cluster (default)
Operator will be available in all Namespaces.
☐ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *
☒ Operator recommended Namespace: **PR** redhat-ods-operator
☐ Select a Namespace

i Namespace creation
Namespace **redhat-ods-operator** does not exist and will be created.

Update approval * ⓘ
☒ Automatic
☐ Manual

Install Cancel


Red Hat OpenShift AI
provided by Red Hat, Inc.


Provided APIs

DSC Data Science Cluster
Required

DataScienceCluster is the Schema for the datascienceclusters API.

Auth
Auth is the Schema for the auths API

- DataScienceCluster custom resource is required for OpenShift AI. Click **Create DataScienceCluster**


Red Hat OpenShift AI
rhods-operator.2.19.0 provided by Red Hat, Inc.

Installed operator: custom resource required

The Operator has installed successfully. Create the required custom resource to be able to use this Operator.

DSC DataScienceCluster
Required

Operator for deployment and management of Red Hat OpenShift AI

Create DataScienceCluster

View Installed Operators in Namespace redhat-ods-operator

- Keep the default settings and click **Create**.

Create DataScienceCluster

Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.

Configure via: ☐ Form view ☒ YAML view

Alt + F1 Accessibility help | View shortcuts | Show tooltips

```

11 spec:
12   components:
13     codeflare:
14       managementState: Managed
15     kserve:
16       nim:
17         managementState: Managed
18         rawDeploymentServiceConfig: Headless
19       serving:
20         ingressGateway:
21           certificate:
22             type: OpenshiftDefaultIngress
23         managementState: Managed
24         name: knative-serving
25         managementState: Managed
26     modelregistry:
27       registriesNamespace: rhoai-model-registries
28       managementState: Removed
  
```

Create Cancel Download

Note: Under spec > components, kserve should be “Managed”

6. After few minutes the status of **All Instances** should be Ready.

Project: redhat-ods-operator ▾

Installed Operators > Operator details

Red Hat OpenShift AI
 2.19.0 provided by Red Hat, Inc.

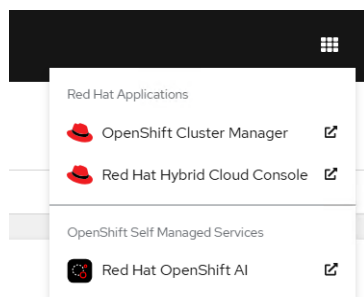
Details YAML Subscription Events All instances Data Science Cluster DSC Initialization Auth

All Instances

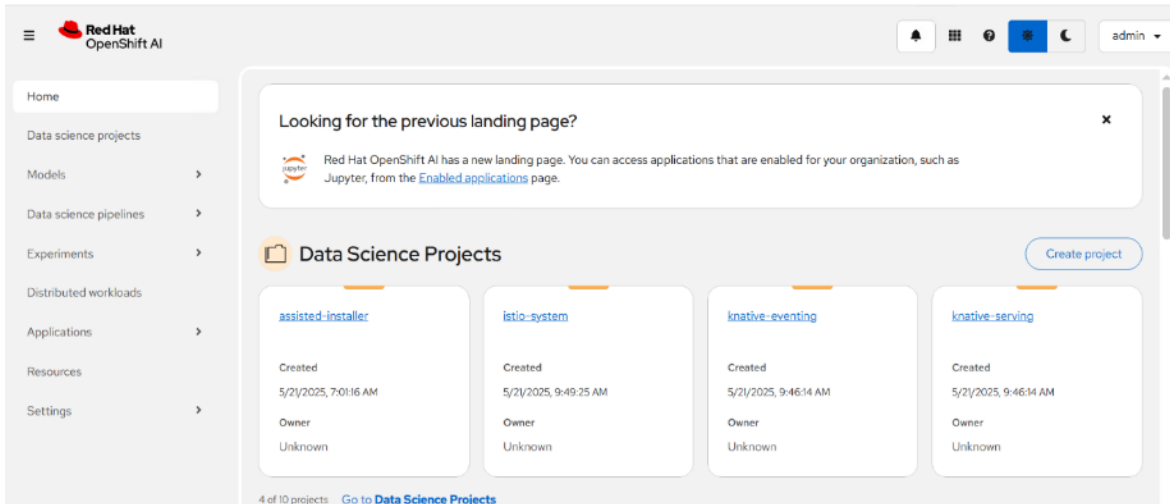
Filter ▾ Name ▾ Search by name... /

Name	Kind	Status	Labels
auth	Auth	Phase: Ready	No labels
default-dsc	DataScienceCluster	Phase: Ready	app.kubernetes.io/created-by=rhods-operator app.kubernetes.io/instance=default-dsc app.kubernetes.io/managed-by=kustomize app.kubernetes.io/name=datasciencecluster app.kubernetes.io/part-of=rhods-operator
default-dsci	DSCInitialization	Phase: Ready	No labels

7. Log into OpenShift AI by clicking the square dotted tile.



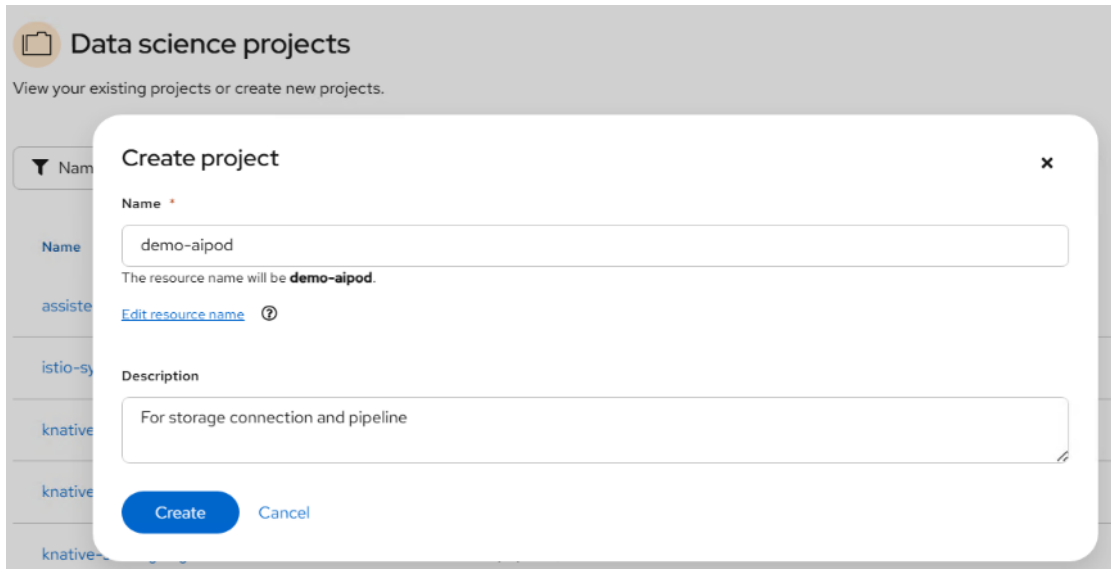
8. Log in with the admin account created earlier with htpasswd.



OpenShift AI – Basic Configuration

This section describes basic configuration of S3 object store and pipeline configuration in Red Hat OpenShift AI.

1. Log into OpenShift AI web console and create project under **Data science projects**.
2. Enter the name of the project.



3. Go to Connections tab and click **Create connection**. From the drop-down list, choose **S3 compatible object storage -v1**.
4. Enter the details related to S3 storage and click **Create**.

Create connection

S3 compatible object storage - v1

View details

Connection details

Connection name *

s3

The resource name will be **s3**.

[Edit resource name](#)
?

Connection description

Access key * ?

WOD5132IN3O1VFR6M7GG

Secret key * ?

.....

Endpoint * ?

http://10.61.178.51

Region ?

us-east-1

Bucket ?

pipeline

Create

Cancel

5. Object storage connection was created successfully.

Overview

Workbenches

Pipelines

Models

Cluster storage

Connections

Permissions

Settings

Connections

?

Create connection

Name

↑

Type

↓

Model serving compatibility

Connected resources

s3

?

S3 compatible object storage - v1

S3 compatible object storage

-

- Go to Pipelines and click **Configure pipeline server**.
- Populate the S3 credentials and other details which were used to create storage connections.

Configure pipeline server

Configuring a pipeline server enables you to create and manage pipelines.

Object storage connection
To store pipeline artifacts. Must be S3 compatible

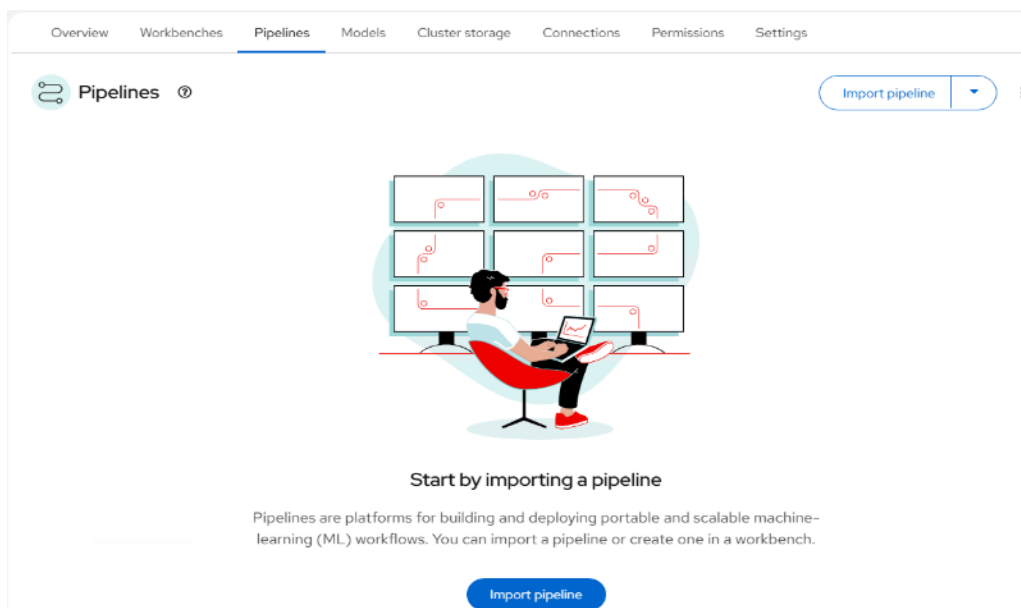
Access key *

Secret key *

Populate the form with credentials from your selected data connection

s3

- Click **Configure pipeline server**. In less than a minute the pipeline server will be configured.



- The pipeline server configuration will deploy below pods to the same namespace.

```
Every 2.0s: oc get pod -n demo-aipod
```

NAME	READY	STATUS	RESTARTS	AGE
ds-pipeline-dspa-8679bccd9d-t52qc	2/2	Running	0	21s
ds-pipeline-metadata-envoy-dspa-5999b9b986-2hgrz	2/2	Running	0	20s
ds-pipeline-metadata-grpc-dspa-599fd98bd-bwxzfz	1/1	Running	0	20s
ds-pipeline-persistenceagent-dspa-86cb969bbf-p7zbw	1/1	Running	0	21s
ds-pipeline-scheduledworkflow-dspa-5644547987-7bkcd	1/1	Running	0	21s
ds-pipeline-workflow-controller-dspa-7b9f6fb78c-lhbbp	1/1	Running	0	21s
mariadb-dspa-9bc764fdf-9pwwk	1/1	Running	0	46s

Note: Once pipeline server is configured, we can create a workbench with standard data science image.

10. Go to workbenches and click **Create workbench**.

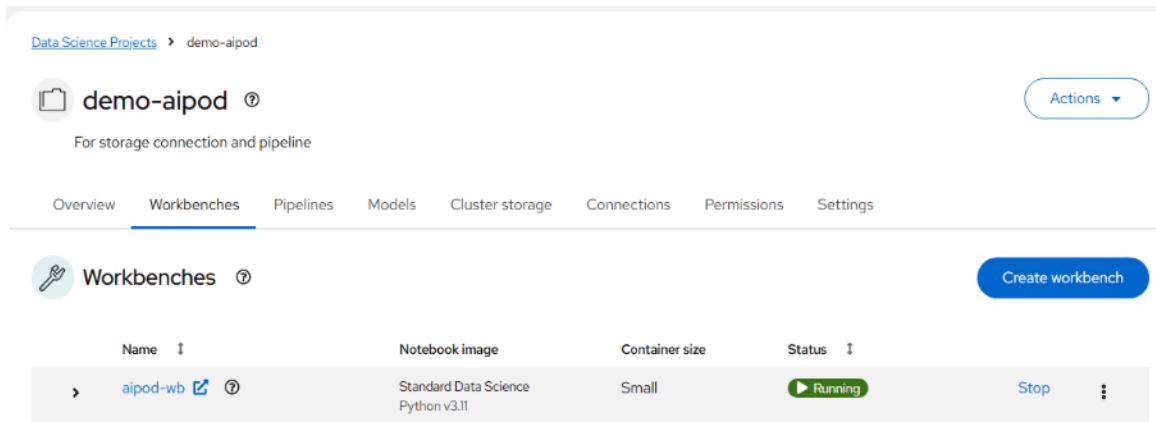
11. Enter the details like -

- Name and description
- Notebook image
- Deployment size
- Environment variables
- Cluster storage
- Connections

12. After providing details, click **Create workbench**.

Note: Connections enable you to store and retrieve information that typically should not be stored in code

13. A workbench with the notebook image mentioned will be created.



Solution Validation

The solution validation was done for the following use-cases.

1. [AI/ML pipeline for Edge](#)
2. [RAG with OpenShift AI and Elasticsearch](#)
3. [NVIDIA NIM with OpenShift AI](#)

AI/ML pipeline for Edge

In this section we will discuss how Red Hat OpenShift AI running on a single-node OpenShift instance can help in building pipelines fully automate the machine learning (ML) and data science workflow at the edge. In this validation OpenShift AI implements Kubeflow Pipelines 2.0. A pipeline is a workflow definition consisting of steps, inputs, and output artifacts stored in S3-compatible storage. These pipelines allow organizations to standardize and automate machine learning workflows that might include the following steps:

- **Data extraction:** Retrieving and collecting data automatically from sensors or databases is a vital step before training models.

- **Data processing:** This consists of manipulating and transforming the data into a format that can be easily interpreted by the model.
- **Model training:** With pipelines we can inject the data into the model and trigger the training process automatically.
- **Model validation:** Comparing the performance of the new model to ensure improvements.
- **Model saving:** When the model is trained and validated, the last step will be automating the process of storing it in an ONTAP S3 bucket.

Note: For this case, we created 2 buckets on the AFF-A30 system using ONTAP S3 feature.

1. We will create a Data science project with name “edge”.



Create project [X]

Name *

edge

The resource name will be **edge**.

[Edit resource name](#) ?

Description

Pipeline for automating AI/ML at the edge







Create Cancel

2. In the edge project page, create a storage connection using ONTAP S3 configuration details, a pipeline server and a workbench.
3. Launch Jupyter notebook and enter OpenShift admin credentials.
4. Clone the repo <https://github.com/dialvare/ai-edge-blog>.
5. Currently the ONTAP S3 bucket is empty.

```
[root@workstation-rhel ~]# aws s3 ls
2025-05-26 01:30:45 pipeline
2025-05-26 01:28:26 model-repo

[root@workstation-rhel ~]# aws s3 ls model-repo
```

6. Go to **ai-edge-blog > pipelines**. There are a total of 4 notebooks.

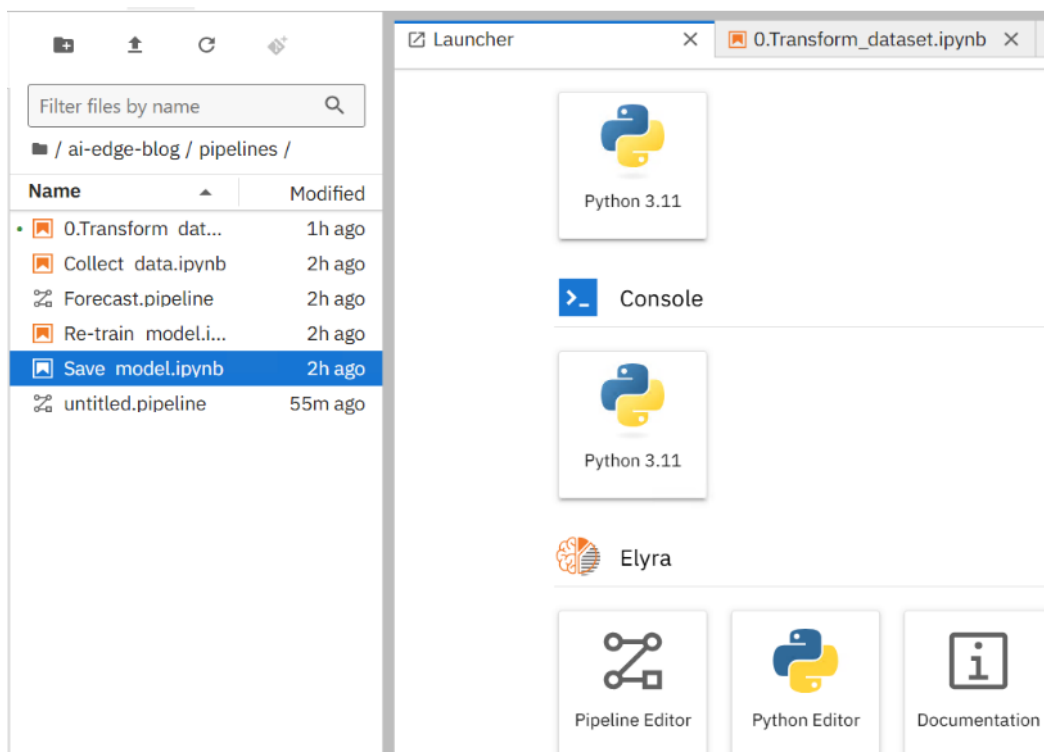
Filter files by name 	
/ ai-edge-blog / pipelines /	
Name	Modified
•  0.Transform dat...	5m ago
 Collect data.ipynb	13m ago
 Forecast.pipeline	13m ago
 Re-train model.i...	13m ago
 Save model.ipynb	13m ago

7. Open **0.Transform_dataset.ipynb** notebook and run it to store the sample data into the data ONTAP S3 bucket.
8. The sample data is stored in the ONTAP S3 bucket now.







```
[root@workstation-rhel ~]# aws s3 ls model-repo
PRE data/

[root@workstation-rhel ~]# aws s3 ls model-repo/data/
2025-05-26 02:41:41          9068 historical.csv
```

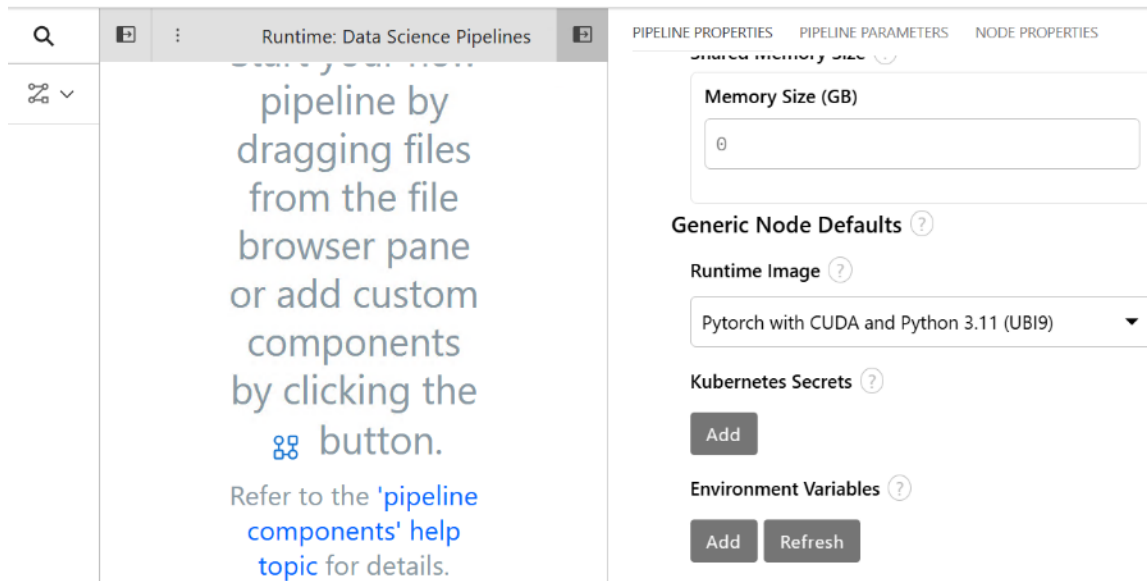
9. Now to collect the data, retrain the model and save the model we will use Pipeline Editor “Elyra” available in the notebook.



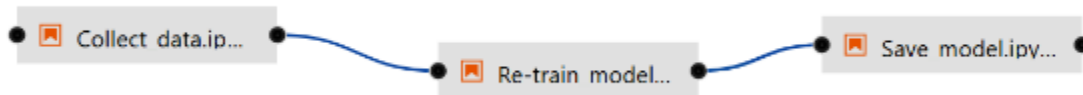
The screenshot displays the Elyra notebook interface. On the left, a file explorer shows the directory structure: / ai-edge-blog / pipelines /. A table lists files with their names and modification times. The file 'Save model.ipynb' is highlighted. On the right, the launcher area shows options for Python 3.11, a Console, and the Elyra logo. Below these are buttons for 'Pipeline Editor', 'Python Editor', and 'Documentation'.

Name	Modified
•  0.Transform dat...	1h ago
 Collect data.ipynb	2h ago
 Forecast.pipeline	2h ago
 Re-train model.i...	2h ago
 Save model.ipynb	2h ago
 untitled.pipeline	55m ago

- Open panel icon and scroll down in the panel to set the Runtime Image and close the panel.



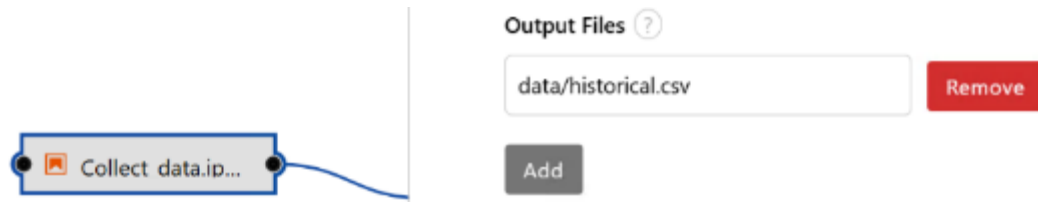
- Drag the **Collect_data.ipynb**, then the **Re-train_model.ipynb** and the **Save_model.ipynb** notebook to the pipeline editor.
- Click on the **Output** of the first box and connect it with the **Input Port** of the second box. Repeat the steps with the second and last notebook.



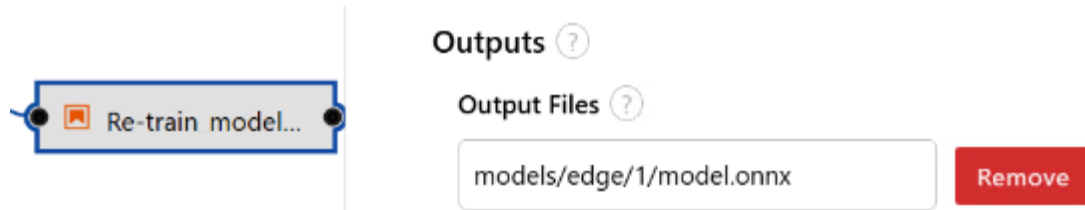
- On the first node, click **NODE PROPERTIES** tab and remove existing environment variables. Under Kubernetes Secrets add below variables.

Environment Variable	Secret Name	Secret Key
AWS_ACCESS_KEY_ID	storage	AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY	storage	AWS_SECRET_ACCESS_KEY
AWS_S3_ENDPOINT	storage	AWS_S3_ENDPOINT
AWS_DEFAULT_REGION	storage	AWS_DEFAULT_REGION
AWS_S3_BUCKET	storage	AWS_S3_BUCKET

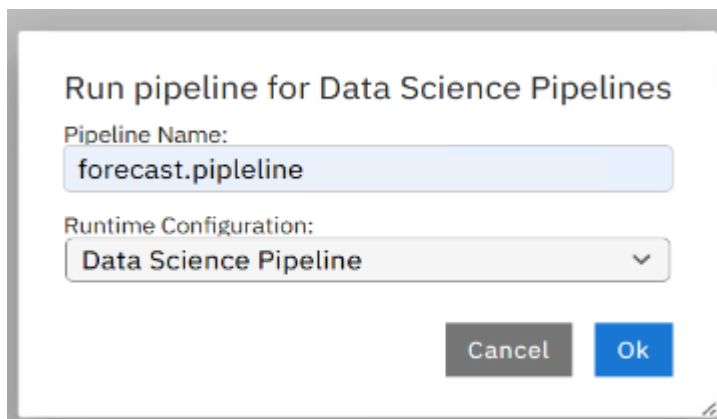
14. Also, under **Output Files**, click Add and enter the Output File. Save the Pipeline.



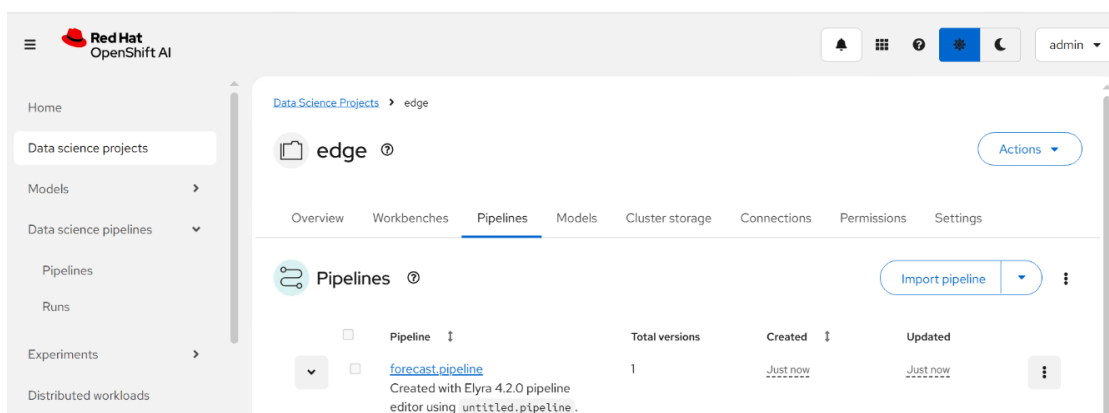
15. To configure the **Re_train_model**, right click second pipeline and select Open Properties. Under Output Files field enter the output path.



16. Configure the **Save_model** node, right click third pipeline and select Open Properties, delete the Environment Variables and add Kubernetes Secrets same as **Collect_data** node.
17. Save and run the pipeline.



18. On the OpenShift AI page, check the pipelines.



19. Click on the Job details and verify the pipeline succeeded.

The screenshot shows the MLflow interface for a pipeline named 'forecast.pipeline-0526092830'. The pipeline is in a 'Running' state. The graph shows three steps: 'Collect_data', 'Re-train_model', and 'Save_model', all of which are completed successfully, indicated by green checkmarks.

20. The entire pipeline artifacts are saved in pipeline ONTAP S3 bucket.

```
aws s3 ls pipeline/forecast.pipeline-0526104118/
PRE data/
PRE models/
2025-05-26 06:41:07 57760 Collect_data-f0436d7d-ef97-48b6-b8f3-0621efdddfa4.tar.gz
2025-05-26 06:41:43 367396 Collect_data.html
2025-05-26 06:41:43 92027 Collect_data.ipynb
2025-05-26 06:41:07 96191 Re-train_model-6678b6a4-8d2e-4bd5-a4d1-b3c9bf1f1c6d.tar.gz
2025-05-26 06:42:34 448536 Re-train_model.html
2025-05-26 06:42:34 162495 Re-train_model.ipynb
2025-05-26 06:41:07 1088 Save_model-b48389fd-328a-4ad8-b771-2b6ab8dc834a.tar.gz
2025-05-26 06:43:06 274574 Save_model.html
2025-05-26 06:43:06 4891 Save_model.ipynb
```

21. The model is saved in the model-repo bucket specified in the storage connection.

```
aws s3 ls model-repo/models/edge/1/
2025-05-26 06:43:05 47321 model.onnx
```

Note: We have configured and executed the pipeline but still we had to manually trigger it every time. We can automate the task by creating a schedule.

22. Navigate back to the **Data Science Pipelines** tab.

23. Click on the three vertical dots on the right side.

The screenshot shows the MLflow Pipelines interface. A table lists the pipelines, with 'forecast.pipeline' selected. The context menu is open, showing options like 'Upload new version', 'Create run', 'Create schedule', and 'Delete pipeline'.

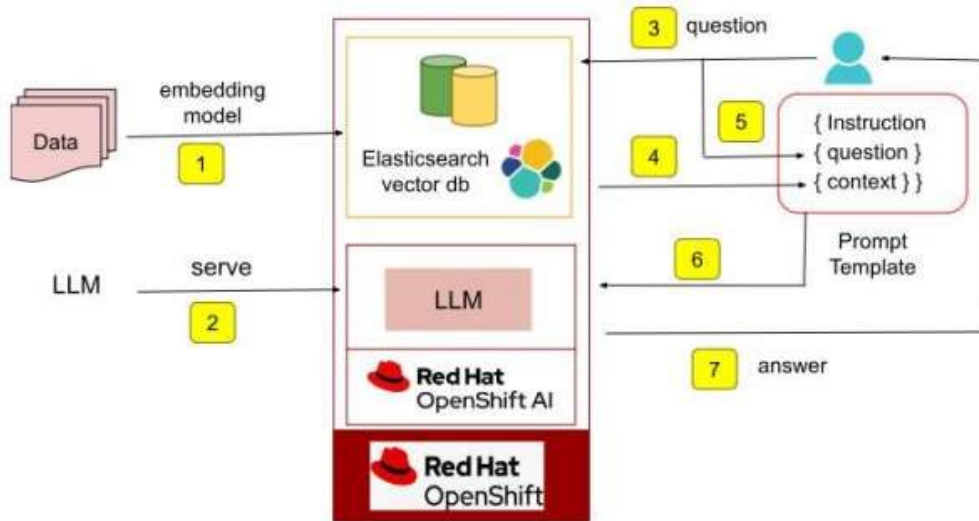
24. Complete the setup by adding all required parameters and click **Create schedule**.

Name	Daily run
Project	edge
Pipeline version	forecast.pipeline
Pipeline	forecast.pipeline
Run ID	3edead0f-80ab-46a9-8d4d-a0f834262a40
Workflow name	Daily run
Created	Monday, May 26, 2025 at 11:01:58 AM UTC
Run trigger enabled	Yes
Trigger	Every 1 day

In this validation, we investigated the utilization of pipelines within Red Hat OpenShift AI to automate the complete AI/ML lifecycle—from data extraction to model training and storage—on a single-node OpenShift instance. Automating workflows at the edge is essential for minimizing latency, enhancing security, and boosting overall efficiency. NetApp AFF A30 provided an intelligent data platform to run the containerized AI/ML workloads along with ONTAP S3 for saving pipeline artifacts and models.

RAG with OpenShift AI and Elasticsearch

Below is the RAG workflow which was validated in this solution. Notebooks are available at <https://github.com/rh-aiservices-bu/rag-with-elasticsearch.git>.



1. Log into OpenShift AI web console and create a data science project title RAG.
2. Create storage connection using ONTAP S3 object store, which is needed to store the LLM.

Note: To serve LLMs with OpenShift AI, models must be stored in object stores. Model servers in OpenShift AI then reference these stored models when serving them

3. Download the model from hugging face.

```
[2]: import os
git_repo = f"https://[REDACTED]huggingface.co/meta-llama/Llama-3.2-3B-Instruct"
!git clone $git_repo

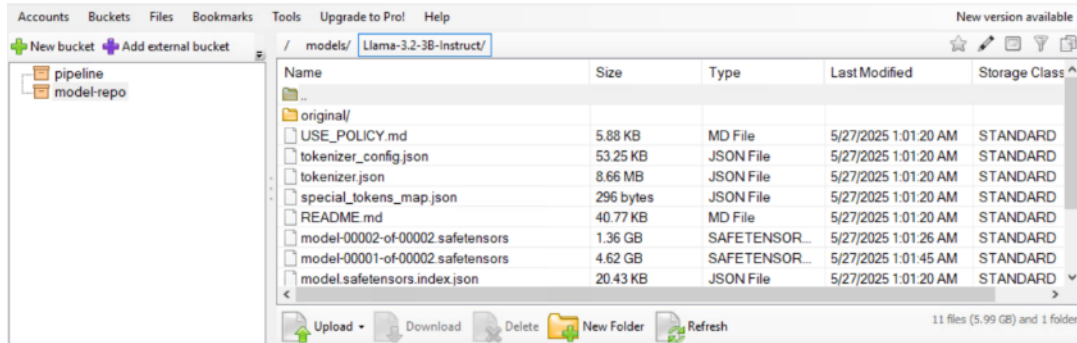
Cloning into 'Llama-3.2-3B-Instruct'...
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 52 (delta 0), reused 0 (delta 0), pack-reused 49 (from 1)
Unpacking objects: 100% (52/52), 2.27 MiB | 2.41 MiB/s, done.
Filtering content: 100% (4/4), 3.97 GiB | 13.47 MiB/s, done.
```

4. Upload the model to the ONTAP S3 bucket.

```
[6]: upload_directory_to_s3(model_name, f"models/{model_name}")

skipping .gitattributes
Llama-3.2-3B-Instruct/LICENSE.txt -> models/Llama-3.2-3B-Instruct/LICENSE.txt
Llama-3.2-3B-Instruct/README.md -> models/Llama-3.2-3B-Instruct/README.md
Llama-3.2-3B-Instruct/USE_POLICY.md -> models/Llama-3.2-3B-Instruct/USE_POLICY.md
Llama-3.2-3B-Instruct/config.json -> models/Llama-3.2-3B-Instruct/config.json
Llama-3.2-3B-Instruct/generation_config.json -> models/Llama-3.2-3B-Instruct/generation_config.json
Llama-3.2-3B-Instruct/model.safetensors.index.json -> models/Llama-3.2-3B-Instruct/model.safetensors.index.json
Llama-3.2-3B-Instruct/special_tokens_map.json -> models/Llama-3.2-3B-Instruct/special_tokens_map.json
Llama-3.2-3B-Instruct/tokenizer.json -> models/Llama-3.2-3B-Instruct/tokenizer.json
Llama-3.2-3B-Instruct/tokenizer_config.json -> models/Llama-3.2-3B-Instruct/tokenizer_config.json
Llama-3.2-3B-Instruct/model-00002-of-00002.safetensors -> models/Llama-3.2-3B-Instruct/model-00002-of-00002.safetensors
Llama-3.2-3B-Instruct/model-00001-of-00002.safetensors -> models/Llama-3.2-3B-Instruct/model-00001-of-00002.safetensors
```

5. Verify on the ONTAP S3 bucket.



6. Install Elasticsearch (ECK) Operator and deploy Elasticsearch instance with a route as described [here](#).

7. Get the routes detail.

Routes

Name	Status	Location	Service
elasticsearch-sample	Accepted	https://elasticsearch-sample-rag.apps.aipod-ocp.fpmc.sa	elasticsearch-sample-es-http

8. Retrieve the password of the elastic user.

```
oc get secret elasticsearch-sample-es-elastic-user -n rag -o go-template='{{.data.elastic | base64decode}}'
```

- Go to Models and click **Deploy model**. Enter the details mentioned in the screen below and scroll down to enter other details.

Deploy model

Configure properties for deploying your model

This is the name of the inference service created when the model is deployed
The resource name will be **llm**.

[Edit resource name](#) ?

Serving runtime *

vLLM NVIDIA GPU ServingRuntime for KServe

You can optimize model performance by [configuring the parameters](#) of the selected serving runtime.

Model framework (name - version) *

vLLM

Deployment mode * ?

Advanced

Number of model server replicas to deploy * ?

- 1 +

Model server size * ?

Large

Limits: 10 CPU, 20GiB Memory Requests: 6 CPU, 16GiB Memory

Accelerator ?

NVIDIA GPU

Deploy Cancel

- Enable the model for external route and enter the path of the ONTAP S3 bucket where the model is stored. Click **Deploy**.

Deploy model

Configure properties for deploying your model

Model route

☒ Make deployed models available through an external route

Token authentication

☒ Require token authentication

The actual tokens will be created and displayed when the model server is configured.

Service account name

Enter the service account name for which the token will be generated

[+ Add a service account](#)

Source model location

☒ Existing connection

Connection

storage

View details

Path

Enter a path to a model or folder. This path cannot point to a root folder.

Deploy

Cancel

11. The inference pod will get deployed and vllm will be loaded in a few minutes.

Project: rag

Pods > Pod details

llm-predictor-00002-deployment-547747785-7rr4p

Running

Actions

Details

Metrics

YAML

Environment

Logs

Events

Terminal

Some lines have been abridged because they are exceptionally long.

To view unabridged log content, you can either [open the raw file in another window](#) or [download it](#).

Log stream paused.

kserve-container

Current log

Search

Show full log

Wrap lines

Raw

Download

Expand

106 lines

22

INFO 05-27 08:25:18 worker.py:267] the current vLLM instance can use total_gpu_memory (44.40GiB) x gpu_memory_util

23

INFO 05-27 08:25:18 worker.py:267] model weights take 6.02GiB; non_torch_memory takes 0.00GiB; PyTorch activation

24

INFO 05-27 08:25:18 executor_base.py:111] # cuda blocks: 19123, # CPU blocks: 2340

25

INFO 05-27 08:25:18 executor_base.py:116] Maximum concurrency for 131072 tokens per request: 2.33x

26

INFO 05-27 08:25:27 model_runner.py:1434] Capturing cudagraphs for decoding. This may lead to unexpected consequen

27

Capturing CUDA graph shapes: 0% | 0/35 [00:00<?, ?it/s]Capturing CUDA graph shapes: 3% | 1/35 [00:00<00:24, 1.

28

INFO 05-27 08:25:47 model_runner.py:1562] Graph capturing finished in 20 secs, took 0.21 GiB

29

INFO 05-27 08:25:47 llm_engine.py:436] init engine (profile, create kv cache, warmup model) took 29.94 seconds

30

INFO 05-27 08:25:48 api_server.py:958] Starting vLLM API server on http://0.0.0.0:8080

75

NetApp AIPod with Lenovo and Red Hat
OpenShift AI for MLOps

© 2025 NetApp, Inc. All rights reserved. NetApp Verified Architecture

12. Get the external url of the Inference endpoint and the token.

The screenshot shows the OpenShift AI console interface. At the top, there are tabs for Overview, Workbenches, Pipelines, Models, and Cluster status. The 'Models' tab is selected. Below the tabs, there's a section titled 'Models and model servers'. A modal window titled 'Inference endpoints' is open, showing two URLs: 'https://llm.rag.svc.cluster.local' and 'https://llm-rag.apps.aipod-ocp.fpmc.sa'. The main page displays a table with details for the 'llm' model deployment, including framework (vLLM), replicas (1), size (Large), and token authentication details.

Model deployment name	Serving runtime	API protocol	Status
llm	vLLM NVIDIA GPU ServingRuntime for KServe	REST	Single-model serving enabled

Internal and external endpoint details

Framework	Model server replicas	Model server size	Accelerator	Number of accelerators	Token authentication
vLLM	1	Large 6 CPUs, 16GiB Memory requested 10 CPUs, 20GiB Memory limit	NVIDIA GPU	1	Token name: default-name Token secret: eyJhbGciOiJSUzI1NiIsImtpZCI6InJYR0JaNFVPE9tWGpxMHM0encxZnZOYXRPUHhOWUFIMmxCLXJ...fwXeVaQ

13. Use the notebook 3_RAG_withElastic.ipynb to define a prompt template, which has all the instructions.

```
template = """
### [INST]
Instruction: Answer the question based on your
OpenShift AI knowledge. Here is context to help:

{context}

### QUESTION:
{question}

[/INST]
"""

os.environ["TOKENIZERS_PARALLELISM"] = "false"

QA_CHAIN_PROMPT = PromptTemplate.from_template(template)

llm = VLLMOpenAI(
    openai_api_key="<token secret of the Inference Endpoint URL>",
    openai_api_base=INFERENCE_SERVER_URL,
    model_name=MODEL_NAME,
    top_p=TOP_P,
    temperature=TEMPERATURE,
    max_tokens=MAX_TOKENS,
    presence_penalty=PRESENCE_PENALTY,
    streaming=True,
    verbose=False,
    callbacks=[StreamingStdOutCallbackHandler()]
)

qa_chain = RetrievalQA.from_chain_type(
    llm,
    retriever=store.as_retriever(
        search_type="similarity",
        search_kwargs={"k": 4}
    ),
    chain_type_kwargs={"prompt": QA_CHAIN_PROMPT},
    return_source_documents=True
)
```

```
os.environ["TOKENIZERS_PARALLELISM"] = "false"
```

14. The model answers the question based on the json data available as knowledge source.

```
[9]: question = "What is Distributed workloads in OpenShift AI, explain in detail?"
result = qa_chain.invoke({"query": question})

### [STEP 1: INTRODUCTION]
Distributed workloads in OpenShift AI is a feature that allows users to queue, scale, and manage the resources required to run data science workloads across multiple nodes in an OpenShift cluster simultaneously. This feature is designed to provide faster iteration and experimentation, as well as the ability to use larger datasets, leading to more accurate models.

### [STEP 2: BENEFITS OF DISTRIBUTED WORKLOADS]
The benefits of distributed workloads in OpenShift AI include:

* **Faster Iteration and Experimentation**: Distributed workloads allow users to iterate faster and experiment more frequently because of the reduced processing time. This enables data scientists to quickly test and refine their models, leading to faster development and deployment.
* **Larger Datasets**: By using distributed workloads, users can use larger datasets, which can lead to more accurate models. This is particularly useful for machine learning (ML) and artificial intelligence (AI) workloads, where large datasets are often required to train complex models.
* **Complex Models**: Distributed workloads enable users to use complex models that could not be trained on a single node. This is because distributed workloads can scale to handle large amounts of data and computational resources, making it possible to train complex models that would be impractical or impossible to train on a single node.
* **Flexibility and Scalability**: Distributed workloads provide flexibility and scalability, allowing users to submit workloads at any time and schedule them when the required resources are available. This means that users can take advantage of spare resources in the cluster, reducing waste and increasing efficiency.

### [STEP 3: COMPONENTS OF DISTRIBUTED WORKLOADS]
The distributed workloads infrastructure in OpenShift AI includes the following components:

* **CodeFlare Operator**: The CodeFlare Operator secures deployed Ray clusters and grants access to their URLs. This ensures that users have secure and reliable access to their distributed workloads.
* **CodeFlare SDK**: The CodeFlare SDK defines and controls the remote distributed compute jobs and infrastructure for any Python-based environment. This provides users with a flexible and customizable way to deploy and manage their distributed workloads.
* **KubeRay**: KubeRay is a component of the distributed workloads infrastructure, providing users with a simple and intuitive way to manage and schedule their workloads.

### [STEP 4: OPENSHIFT AI INTEGRATION]
OpenShift AI integrates the following components and services:
```

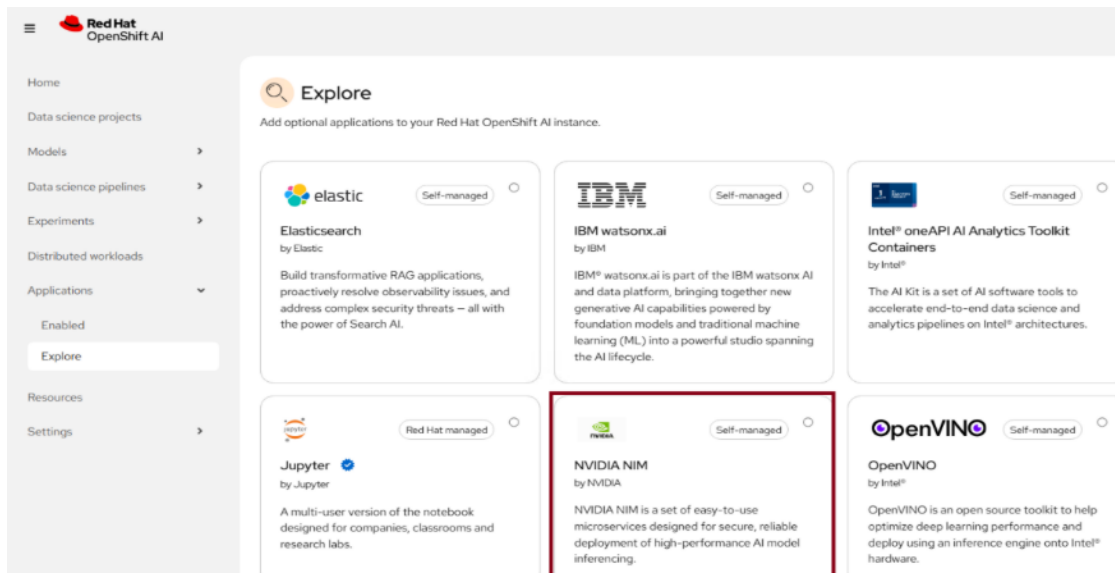
NVIDIA NIM on OpenShift AI

We validated NVIDIA NIM on OpenShift AI. Native support for [NVIDIA NIM](#) microservices is now generally available on [Red Hat OpenShift AI](#) to help streamline inferencing for dozens of AI/ML models on a consistent, flexible hybrid cloud platform. NVIDIA NIM, part of the NVIDIA AI Enterprise software platform, is a set of easy-to-use inference microservices for accelerating the deployment of foundation models and keeping your data secure.

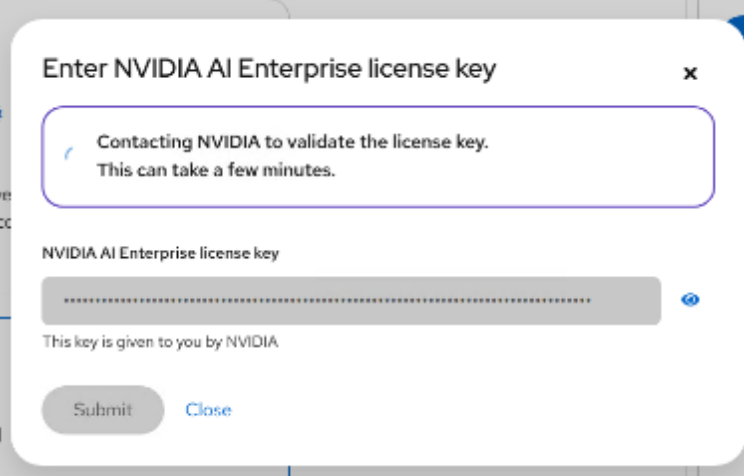
Prerequisite:

- Generate API key from [NVIDIA NGC catalog](#).

1. On the Red Hat OpenShift AI dashboard, go to **Application > Explore**, click **NVIDIA NIM**.

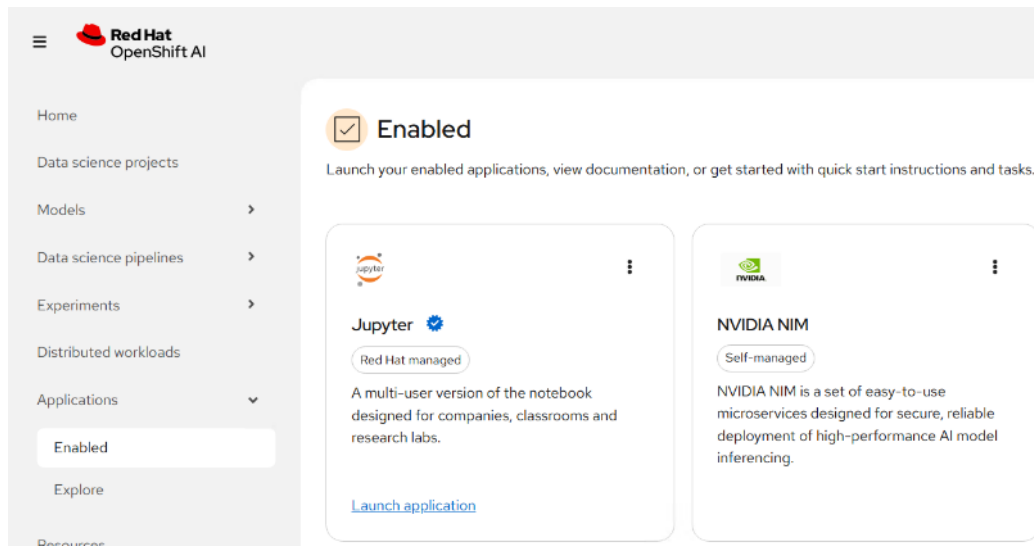


2. Enter the API key and click Submit.

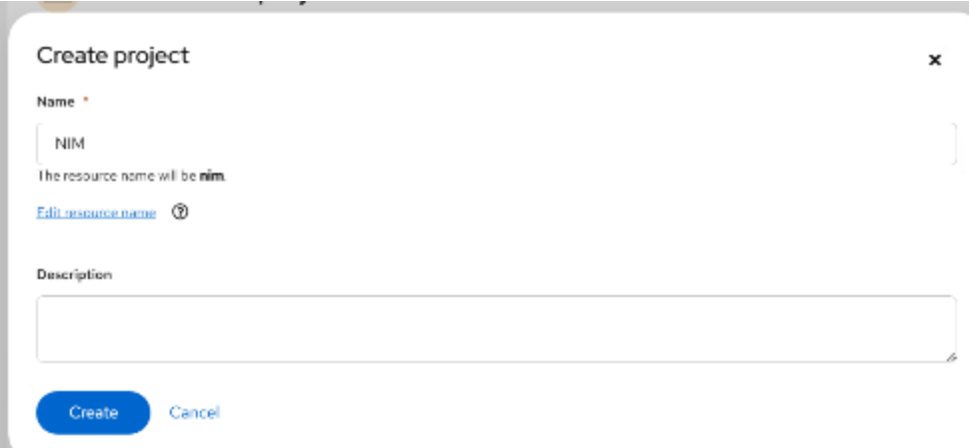


A modal dialog titled "Enter NVIDIA AI Enterprise license key" with a close button (X) in the top right corner. Inside the dialog, there is a status message: "Contacting NVIDIA to validate the license key. This can take a few minutes." Below this is a text input field labeled "NVIDIA AI Enterprise license key" containing a masked key (dots). A note below the field states "This key is given to you by NVIDIA". At the bottom are two buttons: "Submit" and "Close".

3. After the API key validation, NVIDIA NIM card will be enabled.

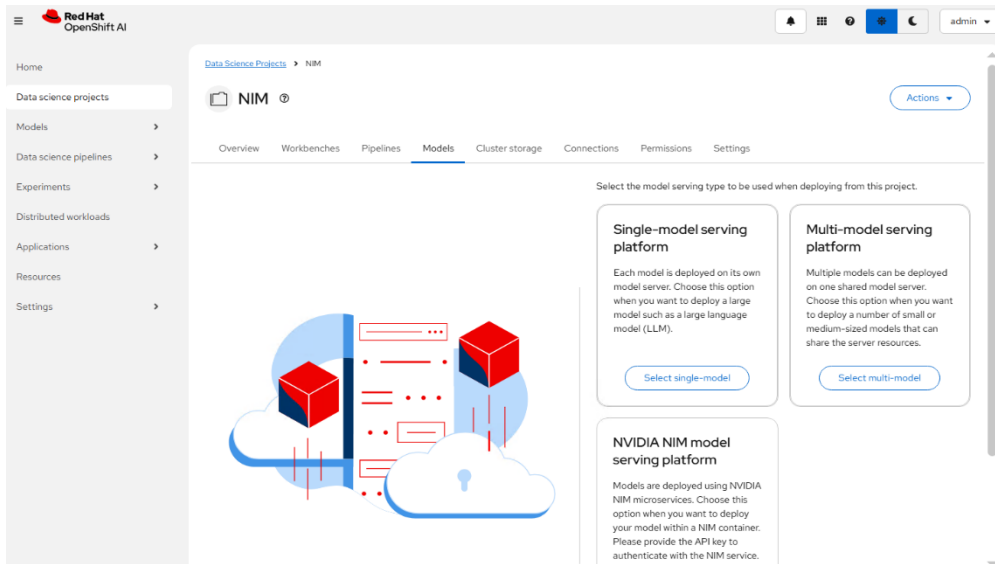


4. Create a project.

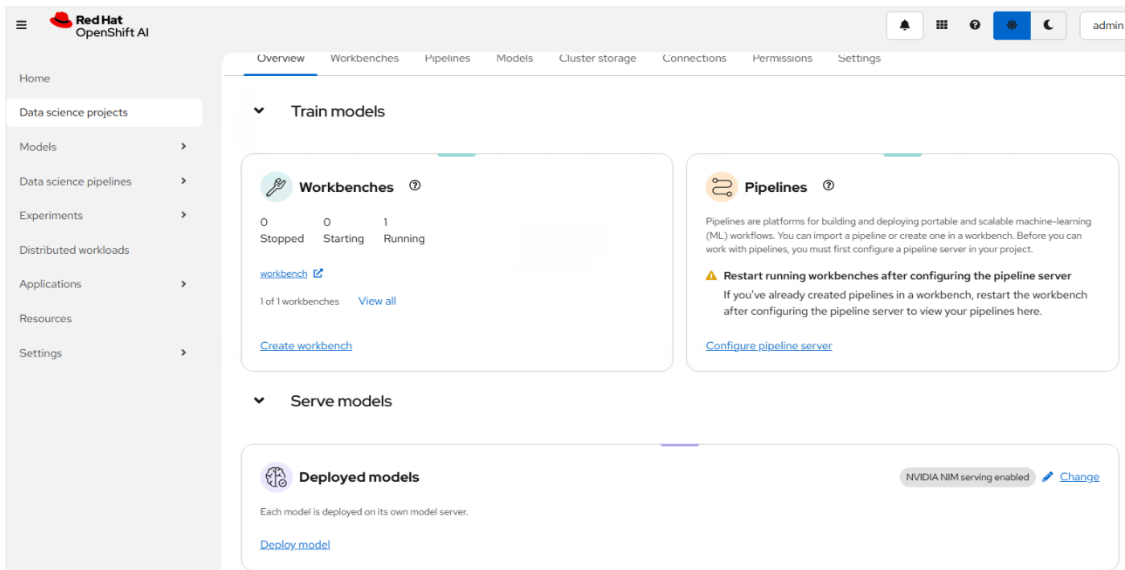


A modal dialog titled "Create project" with a close button (X) in the top right corner. It contains a "Name" field with the value "NIM" and a note "The resource name will be nim." Below the name field is a link "Edit resource name" with a lock icon. There is also a "Description" text area. At the bottom are two buttons: "Create" and "Cancel".

5. In the project we can see NVIDIA NIM serving platform is available. Select NIM to deploy a model.



6. Under Serve model, click Deploy model.



7. Select the desired mode, configure the deployment and click Deploy.

Deploy model with NVIDIA NIM

Configure properties for deploying your model using an NVIDIA NIM.

Project
NIM

Model deployment name *

NVIDIA NIM *

NVIDIA NIM storage size

Specify the size of the cluster storage instance that will be created to store the downloaded NVIDIA NIM.
Make sure your storage size is greater than the NIM size specified by NVIDIA.

Deployment mode * ⓘ

Number of model server replicas to deploy * ⓘ

8. The model deployment started.

Project: nim

Pods > Pod details

llama3-8b-inference-predictor-00001-deployment-644f596cd-l4hp5

Running

Actions

Details

Metrics

YAML

Environment

Logs

Events

Terminal

Log streaming...

kserve-container

Current log

Search

☐ Show full log
 ☐ Wrap lines

41 lines

```

1
2
3  -- NVIDIA Inference Microservice LLM NIM --
4
5
6  NVIDIA Inference Microservice LLM NIM Version 1.0.3
7  Model: meta/llama3-8b-instruct
8
9  Container image Copyright (c) 2016-2024, NVIDIA CORPORATION & AFFILIATES. All rights reserved.
10
11  This NIM container is governed by the NVIDIA AI Product Agreement here:

```

9. Once the container is deployed, the model will be loaded and started.

Serve models

Deployed models

Successful

Failed

NVIDIA NIM serving enabled

Llama3-8B-Inference ⓘ

Serving runtime

NVIDIA NIM

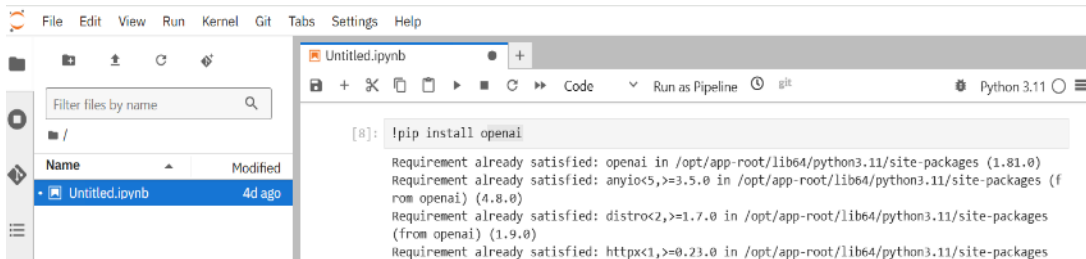
Internal and external endpoint details

1 of 1 models

View all

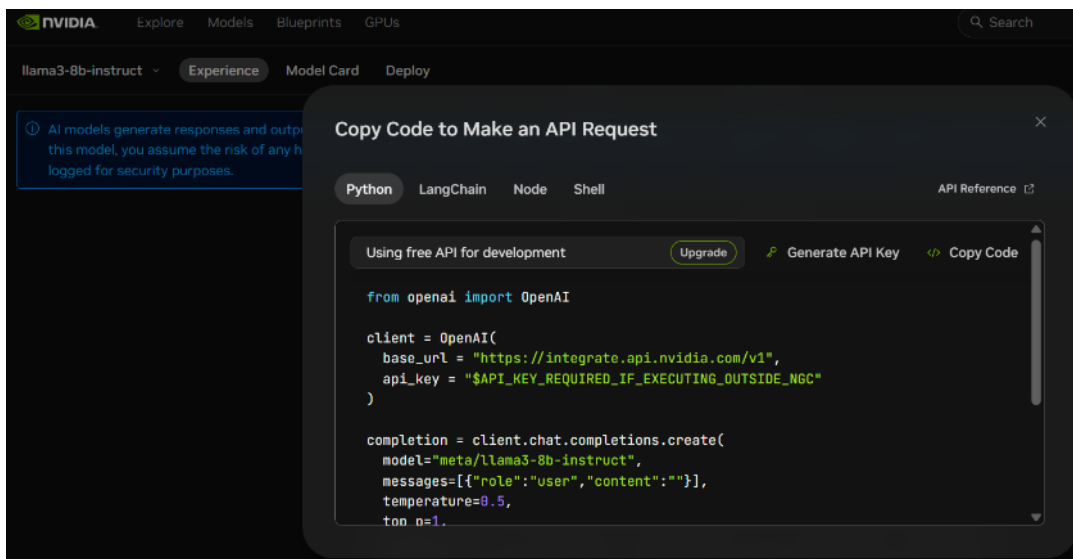
Note: A green check mark appears when the model is ready for inferencing.

10. Switch to the **Models** tab and note down the external URL and access token.
11. Go to **Workbenches** and deploy a workbench with a “Standard Data Science” Image.
12. Open the workbench and launch a new Python Notebook and install the openai library.



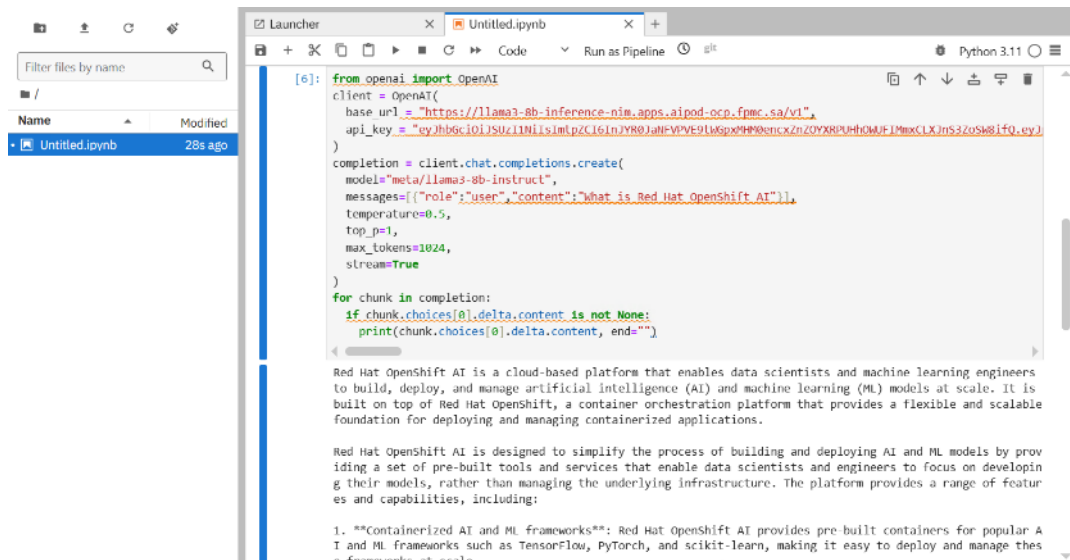
The screenshot shows a Jupyter Notebook titled 'Untitled.ipynb' in a web browser. The left sidebar shows a file explorer with 'Untitled.ipynb' listed. The main area displays the command `!pip install openai` and its output, which lists several requirements already satisfied, including openai (1.81.0), anyio (3.5.0), distro (1.7.0), and httpx (0.23.0).

13. Go to [NVIDIA build cloud](https://build.nvidia.com) and search for the model we deployed and copy the API request code.



The screenshot shows the NVIDIA Build Cloud interface for the 'llama3-8b-instruct' model. A modal window titled 'Copy Code to Make an API Request' is open, showing Python code for using the OpenAI API. The code includes the OpenAI client setup with a base URL and API key, and a chat completion request. The modal also has buttons for 'Upgrade', 'Generate API Key', and 'Copy Code'.

14. In the sample notebook, paste the code and update the `base_url` and `api_key` with external url and access token.



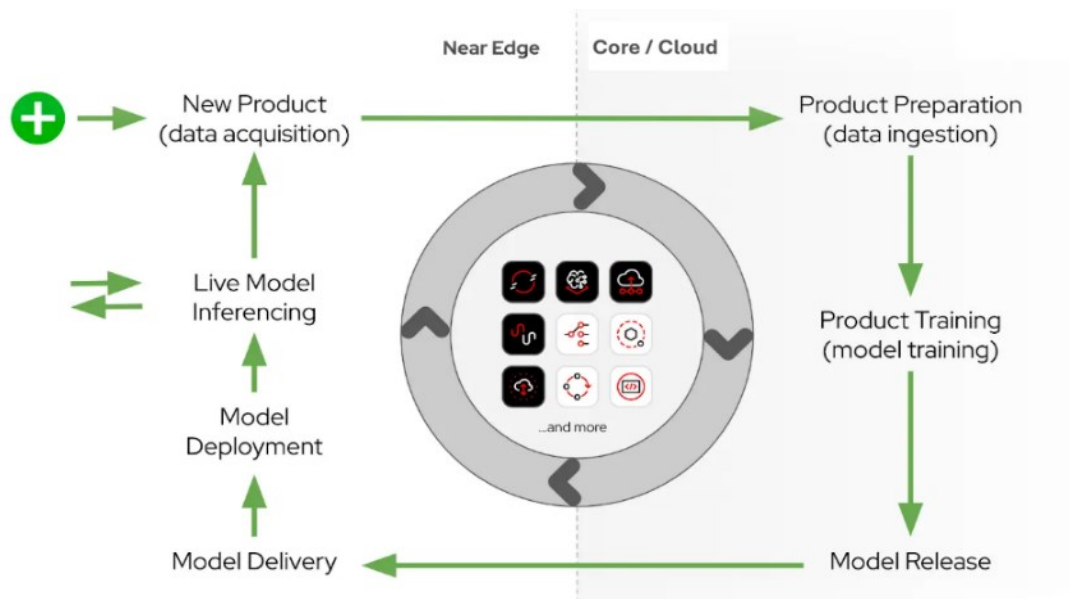
The screenshot shows a Jupyter Notebook titled 'Untitled.ipynb' in a web browser. The left sidebar shows a file explorer with 'Untitled.ipynb' listed. The main area displays the updated Python code for using the OpenAI API. The code includes the OpenAI client setup with a base URL and API key, and a chat completion request. The output shows the response from the API, which includes a message from 'Red Hat OpenShift AI'.

NVIDIA NIM integration on Red Hat OpenShift AI can help enterprises to increase productivity by implementing generative AI to address real business use cases

Edge to Core to Cloud & NetApp Integration

Customers can run their AI/ML workloads and build the pipelines better with NetApp at Edge, Core and to the Cloud with the storage integration on-premises and on cloud. The NetApp ONTAP offering which is available on all the major public cloud providers allows customers to burst into cloud to run training or other workloads without expanding their existing on-premises infrastructure.

Figure 11) Edge to Core/Cloud Pipeline



Conclusion

Transitioning AI/ML projects from proof-of-concept to production poses significant challenges for organizations due to the inherent complexity. Even with a production-ready ML model, integrating it into enterprise applications and data pipelines remains difficult. Scaling these initiatives across multiple applications requires a sustainable operational framework.

The AIPod solution, utilizing Red Hat OpenShift AI running on Single Node OpenShift, provides an entry level comprehensive, flexible, and scalable platform to support an enterprise's AI/ML initiatives. Red Hat OpenShift AI along with Lenovo server, NVIDIA GPU and Networking and NetApp storage offers both pre-integrated and customizable stack to accelerate AI/ML efforts and operationalize AI consistently and efficiently

This solution provides organizations with a compelling on-ramp to AI while powerful performance and management capabilities to enable revolutionary business outcomes by harnessing the power of AI in the data center. The solution empowers data scientists without having any Infrastructure experience deploy, fine-tune and inference the model seamlessly.

Acknowledgments

The authors would like to thank the following people for their support and help during the solution creation.

- Lenovo - Pierce Beary
- NVIDIA – Andy Siegel, Satheesh Iyer
- NetApp – Roney Daniel, Bobby Oommen, Brad Katz, Sriram Sagi, Andy Sayare, Martha DuBois

Bill of Materials

Lenovo ThinkSystem SR675 V3 server (Min quantity 1 for Single Node OpenShift)

Part Number	Product Description	Quantity
7D9RCTOLWW	ThinkSystem SR675 V3 3yr Warranty - HPC&AI with Controlled GPU	1
BR7G	ThinkSystem SR675 V3 4DW PCIe GPU Base	1
BFYA	Operating mode selection for: "Maximum Efficiency Mode"	1
BR2Z	ThinkSystem AMD EPYC 9634 84C 290W 2.25GHz Processor	2
C467	ThinkSystem 128GB TruDDR5 5600MHz (2Rx4) RDIMM-A	4
5977	Select Storage devices - no configured RAID required	1
BLL2	ThinkSystem V3 2U 8x2.5" AnyBay Gen5 Backplane	1
B8P9	ThinkSystem M.2 NVMe 2-Bay RAID Adapter	1
BKSR	ThinkSystem M.2 7450 PRO 960GB Read Intensive NVMe PCIe 4.0 x4 NHS SSD	2
BVBG	ThinkSystem NVIDIA ConnectX-7 NDR200/200GbE QSFP112 2-port PCIe Gen5 x16	1
BPPY	ThinkSystem Intel X710-T4L 10GBase-T 4-Port OCP Ethernet Adapter	1
BT87	ThinkSystem NVIDIA L40 48GB PCIe Gen4 Passive GPU	2
BR7H	ThinkSystem SR675 V3 2x16 PCIe Front IO Riser	1
BR7N	ThinkSystem SR675 V3 x16 PCIe Gen5 Rear IO Riser	1
BK1E	ThinkSystem SR670 V2/ SR675 V3 OCP Enablement Kit	1
BR7Q	ThinkSystem SR675 V3 Direct 4x16 PCIe DW GPU Riser	1
BFPT	ThinkSystem 2400W 230V Platinum Hot-Swap Gen2 Power Supply v2	4
6252	2.5m, 16A/100-250V, C19 to C20 Jumper Cord	4
BRUD	ThinkSystem SR675 V3 Front Video/USB/Diagnostic for 4-DW GPU model	1
C3KA	ThinkSystem SR670 V2/SR675 V3 Heavy Systems Toolless Slide Rail Kit	1
B4QT	2m Mellanox HDR IB Passive Copper QSFP56 Cable	2
AVG0	3m Green Cat6 Cable	1
B7Y0	Enable IPMI-over-LAN	1
BR7V	ThinkSystem SR675 V3 System Board	1
BK15	High voltage (200V+)	1
BR7W	ThinkSystem SR670 V2/ SR675 V3 System Documentation	1
BE0E	N+N Redundancy with Over-Subscription	1
BR82	ThinkSystem SR670 V2/ SR675 V3 WW Packaging	1
C07K	ThinkSystem SR675 V3 Agency Labels w/o EnergyStar	1
BS6Y	ThinkSystem 2U V3 M.2 Signal & Power Cable, SLx4 with 2X10/1X6 Sideband, 330/267/267mm	1
BABV	ThinkSystem Screw for fix M.2 Adapter	1
BFD6	ThinkSystem SR670 V2/ SR675 V3 Power Mezzanine Board	1
BFTH	ThinkSystem SR670 V2/ SR675 V3 Front Operator Panel ASM	1
BRUC	ThinkSystem SR675 V3 CPU Heatsink	2
BFNU	ThinkSystem SR670 V2/ SR675 V3 Intrusion Cable	1
BSD2	ThinkSystem SR675 V3 GPU Supplemental Power Cable 4	2
AVEN	ThinkSystem 1x1 2.5" HDD Filler	8
BR8W	ThinkSystem SR675 V3 Front PCIe Riser Cable 3	1
BFCZ	ThinkSystem SR670 V2/ SR675 V3 PCIe Rear Riser Bracket Filler	1
BR8R	ThinkSystem SR675 V3 Front PCIe Riser Cable 4	1

BR8S	ThinkSystem SR675 V3 Direct DW/SW GPU Riser Cables 1	1
BRUS	ThinkSystem SR675 V3 Rear OCP Cable	1
BS03	ThinkSystem SR675 V3 2400W Power Supply Caution Label	1
C5WV	ThinkSystem SR675 V3 Dual Rotor System Standard Fan	5
BR85	ThinkSystem SR670 V2/ SR675 V3 Branding Label	1
BR88	ThinkSystem SR670 V2/ SR675 V3 Service Label	1
BR7U	ThinkSystem SR675 V3 Root of Trust Module	1
BR8F	ThinkSystem SR675 V3 Backplane to MB Cable 4	1
BR8E	ThinkSystem SR675 V3 Backplane to MB Cable 3	1
BFGY	ThinkSystem SR670 V2/ SR675 V3 Backplane Power Cable 3	1
7Q01CTSAWW	SERVER KEEP YOUR DRIVE ADD-ON	1
QAAK	SR675 V3	1
QAK6	KYD	1
QA0Y	Months	72
7Q01CTS4WW	SERVER PREMIER 24X7 4HR RESP	1
QAAK	SR675 V3	1
QA18	Premier	1
QA0Y	Months	72
QA12	24x7 4hr Resp	1
6311	2.8m, 10A/100-250V, C13 to C14 Jumper Cord	1

Network Switches

SN2201

Lenovo PN	Product Description	Quantity
7D5FCTOGWW	Nvidia SN2201 1GbE Managed Switch with Cumulus (oPSE)	1
BPC8	Nvidia SN2201 1GbE Managed Switch with Cumulus (oPSE)	1
6201	1.5m, 10A/100-250V, C13 to C14 Jumper Cord	2
BSNA	NVIDIA SN2201 Enterprise Rack Mount Kit for Recessed Mounting	1
5WS7B14404	Premier Essential - 3Yr 24x7 4Hr Resp NVID SN2201 oPSE	1
BF94	AI & HPC - ThinkSystem Hardware	1

SN3700

Lenovo PN	Product Description	Quantity
7D5FCTOBWW-HPC	Mellanox SN3700V 200GbE Managed Switch with Cumulus (PSE)	1
BJ5T	Mellanox SN3700 200GbE Managed Switch with Cumulus (PSE)	1
AVG0	3m Green Cat6 Cable	2
6311	2.8m, 10A/100-250V, C13 to C14 Jumper Cord	1
5WS7B98246	3Yr Premier 24x7 4Hr Resp MLNX SN3700V	1
B5RV	Mellanox QM87xx Enterprise RMK w/Air Duct	1
BF94	AI & HPC - ThinkSystem Hardware	1

Additional Cables

Part Number/FC	Product Description	Quantity
40K5793	3m Green Cat5e Cable	7
7Z57A03562 (FC: AV20)	Lenovo 3m Passive 100G QSFP28 DAC Cable – For Server and Storage	3

Software

Part Number	Product Description	Quantity
S6Z3	NVIDIA AI Enterprise Subscription License and Support per GPU Socket, 3 Years	2 (1 per GPU)

7S02CTO1WW	NVIDIA Software	1
5641PX3	XClarity Pro, Per Endpoint w/3 Yr SW S&S	1
1340	Lenovo XClarity Pro, Per Managed Endpoint w/3 Yr SW S&S	1
3444	Registration only	1

AFF A30

Part Number	Product Description	Quantity
AFF-A30-001	AFF A30 HA System	2
AFF-A30A-100-C	AFF A30 HA System,-C	1
DATA-AT-REST-ENCRYPTION	Data at Rest Encryption Capable Operating Sys	2
X800-42U-R6-C	Jumper Crd,In-Cab,C13-C14,-C	2
X97602A-C	Power Supply,1600W,Titanium,-C	2
X66211A-05-N-C	Cable,100GbE,QSFP28-QSFP28,Cu,0.5m,-C	2
X66211B-2-N-C	Cable,100GbE,QSFP28-QSFP28,Cu,2m,-C	8
X5532A-N-C	Rail,4-Post,Thin,Rnd/Sq-Hole,Sm,Adj,24-32,-C	1
X4024A-2-A-C	Drive Pack 2X1.92TB,NVMe4,SED,-C	6
X60130A-C	IO Module,2PT,100GbE,-C	6
SW-ONTAPO-FLASH-A30-C	SW,ONTAP One Package,Per TB,Flash,A30,-C	23
CS-G1A-SE-ADVISOR	SupportEdge Advisor	1
CS-4HR-REPLACEMENT-A	4hr Parts Replacement	1
PS-DEPLOY-STAND-AFF-L	PS Deployment,Standard,AFF,Low	1

Where to Find Additional Information

To learn more about the information that is described in this document, see the following resources:

- Compute:
 - Lenovo ThinkSystem
[Lenovo ThinkSystem SR675 V3 Servers](#)
- NetApp persistent storage for containers:
 - NetApp Trident
<https://netapp.io/persistent-storage-provisioner-for-kubernetes/>
- NetApp Interoperability Matrix:
 - NetApp Interoperability Matrix Tool
<http://support.netapp.com/matrix>
- Networking:
 - NVIDIA [SN2201](#), [SN3700](#)
- Red Hat OpenShift Operator:
 - OpenShift Operator
<https://www.redhat.com/en/technologies/cloud-computing/openshift/what-are-openshift-operators>
- Red Hat OpenShift AI
 - OpenShift AI-Self Managed
https://docs.redhat.com/en/documentation/red_hat_openshift_ai_self-managed/2.19

Version history

As an option, use the NetApp Table style to create a Version History table. Do not add a table number or caption.

Version	Date	Document version history
Version 1.0	June 2025	Initial release

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright information

Copyright © 2025 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data—Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.



