



The bridge to possible

Design and Deployment Guide

Cisco Public

FlashStack for AI: MLOps using Red Hat OpenShift AI

Published: February 2024



In partnership with:



About the Cisco Validated Design Program

The Cisco Validated Design (CVD) program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information, go to: <http://www.cisco.com/go/designzone>.

Introduction

Companies are significantly ramping up their investments in Artificial Intelligence (AI), particularly after the launch of OpenAI's Chat GPT. The explosive interest stems from the potential of this technology. Central to this new technology are machine learning (ML) models trained on large quantities of data. Major tech giants such as Meta, Microsoft, and Google have been making their foundational models widely accessible and these models, specifically the large language models (LLMs) have trillions of parameters trained on hundreds of gigabytes of data which require immense data center resources that only a few companies such as these have access to.

Any organization can build AI applications using these models as they are publicly available, but to bring meaningful differentiated business value requires extensive resources and customization using an organization's own data. It also requires collaboration and integration of workflows and processes between existing teams and newer players with specific expertise in this field to deliver machine learning models that can be used in production. Line of Business (LOB) application teams can then use these models to create applications that deliver unique value to their businesses, fully realizing the potential of the models.

One key aspect to a successful deployment of machine learning models is curating and managing enterprise data. Data used as input to the model is as important as the model itself, as it brings unique business insights and value to the organization, making it critical to the success of an AI/ML effort. Continuously managing changing data from various sources has its own challenges but keeping the models updated with the most recent data is necessary for ensuring the accuracy and reliability of the predictions and recommendations from these models.

Both curation of enterprise data and model delivery processes introduce new workflows with lifecycles and methodologies that differ from existing practices that organizations have in place for software development such as CI/CD and DevOps. These differences are not limited to the workflows but also extend to the development languages, tools, and frameworks used by the teams. With models being an integral part of delivering AI/ML applications, the existing software development and delivery process will need to eventually align with the model life cycle.

Given the complexity of the entire delivery pipeline - from data curation to model delivery to development of new applications - it should not be a surprise that, according to Gartner estimates, [on average, 54% of AI projects make it from pilot to production](#). This figure may improve as organizations continue to invest heavily in AI; however, operationalizing AI remains a daunting challenge. Ensuring successful outcomes requires a strategic, holistic approach to operationalizing and delivering AI/ML models in production that can be leveraged across the organization. Scaling, both in terms of the number of models delivered and the number of applications leveraging them, further adds to the challenge of operationalizing the entire pipeline.

A crucial first step in addressing these challenges is to implement Machine Learning Operations (MLOps) for a more streamlined approach to model delivery. Instead of siloed, ad-hoc model delivery efforts that are highly inefficient, MLOps enables organizations to innovate, scale and bring sustainable value to the business. Although what exactly constitutes MLOps may vary, the end goal is to operationalize and accelerate model delivery with consistency and efficiency. Like DevOps practices that integrated software development and IT operations with continuous integration and delivery to make the application delivery process more agile, MLOps aims to do the same for model delivery using similar principles. Therefore, any organization that wants to operationalize and have any level of scale in production, should implement MLOps as siloed efforts are not sustainable in the long term.

In this solution, Red Hat OpenShift AI is deployed as the MLOps platform on FlashStack Virtual Server Infrastructure (VSI) running Red Hat OpenShift. FlashStack VSI is a leading infrastructure solution in enterprise

data centers, supporting a range of workloads including SAP, Oracle, Virtual Desktop Infrastructure (VDI) and High-Performance Computing (HPC) workloads. Existing FlashStack deployments can be expanded to support AI/ML use cases or deploy standalone, dedicated infrastructure for AI/ML. Additionally, optional components from NVIDIA AI Enterprise were installed to extend specific GPU capabilities of the NVIDIA A100 GPU.

OpenShift AI, an add-on to Red Hat OpenShift, is integrated into the FlashStack AI infrastructure to support a range of AI/ML use cases. OpenShift AI enables organizations to streamline and standardize their model delivery process, while providing flexibility with both integrated and custom options. OpenShift AI also provides pipeline automation and scalable model serving with support for inferencing servers such as Intel OpenVINO and NVIDIA Triton. OpenShift AI leverages the underlying OpenShift platform to provide the infrastructure resources for model development, the NVIDIA GPU, and Pure Storage operators to manage GPU resources and storage for hosting image registries, model registries and pipeline artifacts that are all required for model delivery. The model pipeline can be integrated seamlessly with Red Hat application delivery eco-system using GitOps pipelines making it easier for organizations to integrate these models with intelligent AI-enabled applications.

The Cisco UCS X9508 chassis used in the solution can support up to 8 x NVIDIA A100-80 GPUs and 24 x NVIDIA T4 GPUs, depending on the need. Additionally, the Cisco UCS C-series rack server platform can support 3 x A100-80 GPUs. Both platforms also support a wide range of other NVIDIA GPU models. As more GPUs are needed, the solution can be easily expanded by incrementally adding either Cisco UCS X-Series or C-series servers with GPUs. Cisco UCS systems can also reduce operational costs by minimizing energy requirements. Cisco UCS X-Series earned the [2023 SEAL Sustainable Product Award](#) for products that are “purpose-built” for a sustainable future. UCS systems are managed from the cloud using Cisco Intersight so bringing a new server or chassis online can be as easy as deploying a pre-built server profile template.

The Pure Storage platforms in the solution include both a FlashArray for Unified File and Block storage and a FlashBlade as an S3 compatible object store. Portworx, backed by Pure Storage FlashArray, provides persistent container storage for Red Hat OpenShift and ML model work in OpenShift AI. Pure Storage FlashBlade is introduced in this solution to meet specific AI/ML storage requirements and serves as a model repository for model serving and for storing automation pipeline artifacts.

The FlashStack infrastructure in the solution connect into access layer Cisco Nexus 9300-GX series switches, capable of 400 GbE uplink connectivity. These switches can integrate into existing or new data center fabrics for a high throughput, low latency, and lossless Ethernet fabric connectivity to support AI/ML use cases. Design options for the upstream network fabric can be found in the [Cisco Data Center Networking Blueprint for AI/ML Applications](#) document. Recommended design options include an MP-BGP EVPN VXLAN fabric and an IP BGP design based on Massively Scalable Data Center (MSDC) designs with support for RDMA over Converged Ethernet version 2 (RoCEv2).

This document serves as a design and deployment guide. Detailed implementation steps and use case code are available in the Cisco UCS solutions repository for [this solution](#).

Audience

This document is intended for, but not limited to, sales engineers, technical consultants, solution architecture and enterprise IT, and machine learning teams interested in learning how to design, deploy, and manage a production-ready AI/ML infrastructure for hosting machine learning models and AI-enabled applications.

Purpose of this Document

This document serves as a reference architecture for MLOps using Red Hat OpenShift AI to accelerate AI/ML efforts and deliver models that application teams can use to build intelligent applications. This document also

provides design and deployment guidance for building a production-ready AI/ML infrastructure based on Red Hat OpenShift deployed on a FlashStack VSI using optional NVIDIA AI Enterprise software and GPUs.

Technology Overview

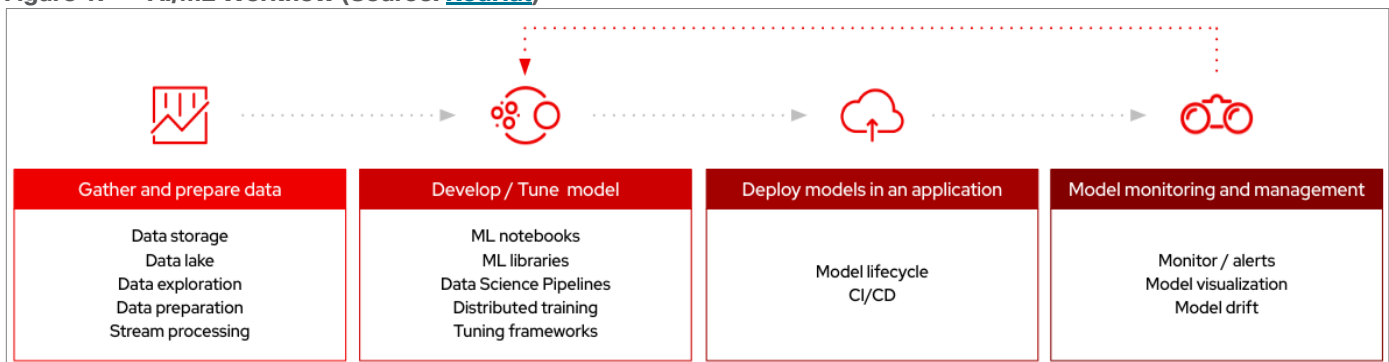
This chapter contains the following:

- [Operationalizing AI](#)
- [Machine Learning Operations](#)
- [Red Hat OpenShift AI](#)
- [Red Hat OpenShift](#)
- [NVIDIA AI Enterprise](#)
- [NVIDIA A100 Tensor Core GPU](#)
- [Compute Unified Device Architecture](#)
- [GPU Deployment Options](#)
- [VMware DirectPath I/O](#)
- [Virtual GPU](#)
- [Single Root I/O Virtualization \(SR-IOV\)](#)
- [Multi-Instance GPU](#)
- [MIG and vGPUs](#)
- [FlashStack VSI](#)
- [FlashStack for AI](#)

Operationalizing AI

The goal of any Enterprise AI/ML effort is to deploy intelligent applications that bring value to the business. AI/ML introduces new roles, technologies, tools, and processes that need to be streamlined with continuous integration and continuous delivery, similar to how Enterprise applications are developed and maintained. The [Figure 1](#) illustrates the end-to-end life cycle of an AI/ML effort. It shows the high-level workflow for developing, deploying, and maintaining AI-enabled applications in production. The individual stages in the workflow will typically involve a series of additional stages and can be thought of as individual pipelines, each with its own life-cycle.

Figure 1. AI/ML Workflow (Source: [RedHat](#))



Data Management – This is represented by the first box in the AI/ML workflow. In this stage, data is collected and consolidated from various data sources. This is the data that the organization owns and is important for

achieving the unique value ML models can provide. Enterprises must continuously manage a steady-stream of changing data from different sources. Data engineers may have to perform activities such as ingestion, exploration, labeling and preparation to deliver the curated data to the second stage in the above workflow, the model delivery or ML pipeline. This stage has a life cycle of its own and can be thought of as data management pipeline.

Model Delivery - This is represented by the second box in the AI/ML workflow (see [Figure 1](#)). It includes the development and deployment stages of delivering a model in production (model serving). The published models are made available using a standard interface that Enterprise LOB teams can use to build their applications. This ML pipeline is the focus of MLOps and typically involves the following stages. The next section will look at ML pipelines and MLOps in greater detail.

- **Access ML-ready data** - This is the output from the **Gather and Prepare Data** box in the above pipeline. It takes as input consolidated structured or unstructured data that has been cleaned, labeled, and formatted to use in model training (and related functions such as fine-tuning). The data pipeline for AI delivers a curated dataset that serves as input to ML pipeline (and MLOps).
- **Model Training** - In this stage, data from the previous stage can be used to develop and train a new model from scratch or retrain/fine-tune a publicly-available foundational model using Enterprise data. This stage may also involve experimentation to identify a candidate model that best suits the needs of the use case in question. Other model customizations such as fine-tuning, and prompt engineering may also be done in this stage.
- **Model Validation** - In this stage, the model that has been selected and trained using Enterprise data is tested to ensure that it is ready for production deployment.
- **Model Serving** - In this stage, the model is deployed into production. Models are deployed as a service and made available through a standard interface for the application teams to use. In this stage, you may use an inferencing server to address performance requirements.
- **Automation** - Here, the model delivery workflow or pipeline is automated for continuous integration and delivery to adapt to new data and other feedback from use in production.

AI-Enabled Application Deployment - In this stage, enterprise LOBs and application teams take the delivered model and integrate it into existing software development processes with CI/CD, and other DevOps and GitOps practices to deliver intelligent applications using ML models. The models and applications are continuously monitored in production with a feedback loop to continuously improve the model's performance and accuracy.

Machine Learning Operations

Machine Learning Operations (MLOps) are a set of best-practices to streamline and accelerate the delivery and machine learning (ML) models. The delivery of these ML models for production use or **model serving** is key to operationalizing AI so that Enterprises can develop intelligent AI-enabled applications. Once delivered, the maintenance of the models are critical for ensuring the accuracy and reliability of model predictions and other outputs. MLOps leverages DevOps and GitOps principles to enable continuous retraining, integration, and delivery. MLOps brings consistency and efficiency to the model delivery process. This coupled with automation minimizes technical debt across models, enabling Enterprises to deliver and maintain models at scale. MLOps pipelines also need to continuously retrain models to keep up with everchanging data to ensure model performance. Updating data pipeline will trigger ML pipeline to deliver a model which in turn should trigger the AI-enabled application to integrate the changes.

MLOps involve new roles such as data scientists and ML engineers that weren't part of traditional software/application development. MLOps typically involve a wide ecosystem of technologies, libraries, and

other components. MLOps platforms will typically provide tools that data scientists and ML engineers can use in different stages of the ML pipeline, including automation capabilities.

Red Hat OpenShift AI

Red Hat OpenShift AI (previously known as Red Hat OpenShift Data Science or RHODS) is a flexible and scalable platform for MLOps using Red Hat OpenShift as the foundation. Along with OpenShift AI, all AI/ML workloads, including ML models and AI-enabled applications can be hosted on OpenShift. IT operations teams that manage Kubernetes cluster resources for existing application environments, can continue to do the same for OpenShift AI and AI/ML workloads. Once provisioned, the resources will be directly accessible from the OpenShift AI console for AI/ML teams to use.

Red Hat OpenShift AI includes key capabilities to accelerate the delivery of AI/ML models and applications in a seamless, consistent manner, at scale. The platform provides the development environment, tools, and frameworks that data scientists and machine learning teams need to build, deploy, and maintain AI/ML models in production. OpenShift AI streamlines the ML model delivery process from development to production deployment (model serving) with efficient life cycle management and pipeline automation. From the OpenShift AI console, AI teams can select from a pre-integrated, Red Hat supported set of tools and technologies or custom components that are enterprise managed, providing the flexibility that teams need to innovate and operate with efficiency. OpenShift AI also makes it easier for multiple teams to collaborate on one or more efforts in parallel.

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. Other key features of OpenShift AI include:

- **Collaborative Workspaces:** OpenShift offers a collaborative workspace where teams can work together and collaborate on one or more models in parallel.
- **Development Environments:** ML teams can use Jupyter notebooks as a service using pre-built images, common Python libraries and open-source technologies such as TensorFlow and PyTorch to work on their models. In addition, administrators can add customized environments for specific dependencies or for additional IDEs such as RStudio and VSCode.
- **Model Serving at scale:** Multiple Models can be served for integration into intelligent AI-enabled applications using inferencing servers (for example, Intel OpenVINO, NVIDIA Triton) using GPU or CPU resources provided by the underlying OpenShift cluster without writing a custom API server.
- **Innovate with open-source capabilities:** Like Red Hat OpenShift, OpenShift AI integrates with open-source tools and leverages a partner ecosystem to enhance the capabilities of the platform, minimizing vendor lock-ins.
- **Data Science Pipelines for GUI-based automation using OpenShift pipelines:** OpenShift AI leverages OpenShift pipelines to automate ML workflow using an easy to drag-and-drop web UI as well as code driven development of pipelines using a Python SDK.

By using Red Hat OpenShift AI, enterprises can manage and maintain AI/ML models and the applications that use the models on a single, unified platform that IT organizations may already be using.

Red Hat OpenShift

Red Hat OpenShift is an application platform that drives innovation, anywhere. It empowers organizations to modernize their applications and infrastructure, build new cloud-native applications, accelerate their digital transformation, and fuel growth. AI/ML workloads typically run as docker containers or on Linux virtual

machines. The most popular ML models, frameworks and test applications from Hugging Face and NVIDIA GPU Cloud (NGC) are available as pre-packaged containers. Red Hat OpenShift AI leverages OpenShift's capabilities in application development and container infrastructure management to enable a robust, scalable, and secure environment for model delivery and MLOps. OpenShift Administrators manage all aspects of the underlying infrastructure, from GPU resources to storage to user access. This eases the operational burden on ML engineers and data scientists, enabling them to focus on model delivery and less on managing the infrastructure. This operational benefit is a key advantage of using OpenShift AI such as the underlying infrastructure is administered by IT teams that currently manage OpenShift. The provisioned resources (for example, GPUs), and other aspects such as identity management and user access are seamlessly available and integrated into OpenShift AI, making it significantly easier to use the platform.

Kubernetes Operators

AI/ML workloads, like many modern applications, are using containers and Kubernetes (K8S) orchestration as the de facto development environment for model development and AI-enabled applications. Kubernetes offer several benefits, but one key attribute is its extensibility. Kubernetes provides an Operator framework that vendors and open-source communities can use to develop and deploy self-contained operators that extend the capabilities of the K8s cluster. These operators generally require minimum provisioning and are usually self-managed with automatic updates (unless disabled) and handle life-cycle management. Kubernetes operators are probably the closest thing to an easy-button in infrastructure provisioning (short of IaC). In the Red Hat OpenShift environment that this solution uses, it is even easier to deploy and use operators. Red Hat OpenShift provides an embedded OperatorHub, directly accessible from the cluster console. The Red Hat OperatorHub has hundreds of Red Hat and community certified operators that can be deployed with a few clicks.

To support AI/ML workloads and OpenShift AI, the following Red Hat OpenShift operators are deployed in this solution to enable GPU, storage, and other resources:

- Red Hat Node Feature Discovery Operator to identify and label hardware resources (for example, NVIDIA GPUs)
- NVIDIA GPU Operator deploys and manages the GPU resource on a Red Hat OpenShift cluster (for example, Guest OS vGPU drivers)
- Portworx Operator from Pure Storage for managing persistent storage required for model delivery (for example, image registry)
- Red Hat Data OpenShift AI Operator deploys OpenShift AI on any OpenShift cluster
- OpenShift Pipelines for automating model pipelines in OpenShift AI

For more information on Red Hat OpenShift Operators, see: <https://www.redhat.com/en/technologies/cloud-computing/openshift/what-are-openshift-operators>.

NVIDIA AI Enterprise

NVIDIA AI Enterprise (NVAIE) is a comprehensive suite of enterprise-grade, cloud-native software, hardware, and support services offered by NVIDIA for artificial intelligence (AI) and machine learning (ML) applications. NVIDIA describes NVAIE as the "Operating System" for enterprise AI. NVIDIA AI Enterprise includes key enabling technologies for rapid deployment, management, and scaling of AI workloads. It includes NVIDIA GPUs, Kubernetes Operators for GPUs, virtual GPU (vGPU) technology, and an extensive software library of tools and frameworks optimized for AI that make it easier for enterprises to adopt and scale AI solutions on NVIDIA infrastructure.

NVAIE can be broadly categorized into Infrastructure Management, AI Development, and Application Frameworks optimized for AI. For more details on NVAIE, see: <https://www.nvidia.com/en-us/data-center/products/ai-enterprise/>.

This solution optionally leverages the NVIDIA AI Enterprise Software suite along with complementary partner components to extend and operationalize a robust, production-ready FlashStack AI infrastructure. To operationalize and scale ML models in production requires a streamlined approach that **MLOps** offers. In his solution, MLOps is provided by Red Hat OpenShift AI, with Red Hat OpenShift providing cloud-native (Kubernetes) cluster management and orchestration, utilizing NVIDIA’s GPU operator for seamless deployment and management of GPU resources and CUDA libraries for optimal use of GPU to support various AI/ML use cases. NVIDIA AI Enterprise can be used to extend those capabilities even further.

NVAIE is a licensed software from NVIDIA that must be certified to run on the infrastructure servers. For more information on the licensing and certification, please see the links below:

- **Licensing:** <https://resources.nvidia.com/en-us-ai-enterprise/en-us-nvidia-ai-enterprise/nvidia-ai-enterprise-licensing-guide?pflpid=5224&lb-mode=preview>
- **Certification:** <https://www.nvidia.com/en-us/data-center/products/certified-systems/>

For additional information on NVIDIA AI Enterprise, please see NVIDIA’s website at: <https://www.nvidia.com/>.

NVIDIA A100 Tensor Core GPU

NVIDIA A100 Tensor Core Graphical Processing Unit (GPU) is based on NVIDIA’s Ampere architecture, designed to accelerate computationally intensive workloads such as Artificial intelligence (AI) training and inferencing, deep learning, data science, data analytics, and HPC workloads in the enterprise data center.

The NVIDIA A100 is available as PCIe adapter on Cisco UCS servers with support for either 40GB or 80GB of GPU memory. Each A100 GPU can consume approximately 250W (40G) to 300W (80G) of power.

Table 1. NVIDIA A100 – Technical Specifications

| | NVIDIA A100 PCIe 40GB | NVIDIA A100 PCIe 80GB |
|--------------------------------|--------------------------|--------------------------|
| FP64 | 9.7 TFLOPS | 9.7 TFLOPS |
| FP64 Tensor Core | 19.5 TFLOPS | 19.5 TFLOPS |
| FP32 | 19.5 TFLOPS | 19.5 TFLOPS |
| Tensor Float 32 (TF32) | 156 TFLOPS 312 TFLOPS* | 156 TFLOPS 312 TFLOPS* |
| BFLOAT16 Tensor Core | 312 TFLOPS 624 TFLOPS* | 312 TFLOPS 624 TFLOPS* |
| FP16 Tensor Core | 312 TFLOPS 624 TFLOPS* | 312 TFLOPS 624 TFLOPS* |
| INT8 Tensor Core | 624 TOPS 1248 TOPS* | 624 TOPS 1248 TOPS* |
| GPU Memory | 40GB HBM | 80GB HBM2e |
| GPU Memory Bandwidth | 1,555GB/s | 1,935 GB/s |
| Max Thermal Design Power (TDP) | 250W | 300W |
| Multi-Instance GPU | Up to 7 MIGs @5GB | Up to 7 MIGs @ 10GB |

| | NVIDIA A100 PCIe 40GB | NVIDIA A100 PCIe 80GB |
|----------------|---|---|
| Form Factor | PCIe | PCIe |
| Interconnect | PCIe Gen4: 64 GB/s NVIDIA® NVLink® Bridge for 2 GPUs: 600 GB/s ** | PCIe Gen4: 64 GB/s NVIDIA® NVLink® Bridge for 2 GPUs: 600 GB/s ** |
| Server Options | Partner and NVIDIA-Certified Systems™ with 1–8 GPUs | Partner and NVIDIA-Certified Systems™ with 1–8 GPUs |

Cisco UCS Server Options for NVIDIA are listed in [Table 2](#).

Table 2. NVIDIA A100 on Cisco UCS

| | Server Options |
|------------------|---|
| NVIDIA A100 PCIe | UCS X-series: <ul style="list-style-type: none"> • Cisco UCS X210M6 with X440p PCIe node (Up to 2 x A100-80 per PCIe node) • Cisco UCS X210M7 with X440p PCIe node (Up to 2 x A100-80 per PCIe node) • Total of 8 x A100-80 s on a UCS-X9508 chassis Cisco UCS 240 M7: Up to 3 Cisco UCS 240 M6: Up to 3 Cisco UCS 245 M6 (AMD): Up to 3 |

For more information, see: <https://www.nvidia.com/en-us/data-center/a100/>

Compute Unified Device Architecture

Compute Unified Device Architecture (CUDA) is a parallel computing platform and application programming interface (API) model from NVIDIA that enables general purpose computing on GPUs that were originally designed for graphics. CUDA excels in complex mathematical computations and data processing tasks that can run on thousands of GPU cores in parallel, making it well suited for compute-intensive AI/ML use cases. It also provides memory management to enable efficient data transfers between the CPU and GPU. NVIDIA’s **CUDA Toolkit** provides developers with the software tools and libraries for developing GPU-accelerated applications that harness the parallel processing capabilities of the GPUs. It includes a compiler, debugger, runtime libraries, and other tools that simplify the process of GPU programming.

GPU Deployment Options

VMware vSphere provides two primary options for using GPU accelerators in a virtualized environment, making it a good choice for hosting containerized AI/ML workloads. The two options are:

- VMware DirectPath I/O
- Virtual GPU

VMware DirectPath I/O

In VMware vSphere, a physical PCIe device can be deployed in **VMware DirectPath I/O** or PCI passthrough mode, to enable virtual machines direct access to the physical PCIe device by bypassing the ESXi hypervisor. This can be used with different types of PCIe devices such as networking adapters or GPU accelerators. GPU passthrough is typically used when a single VM application requires the full resources of one or more physical GPUs installed on that server. For compute-intensive AI/ML workloads such as training or fine-tuning of large

language models, GPU passthrough is an important capability for achieving near native performance, similar to that of bare-metal deployments. To operate in this mode, a GPU driver must be installed on the Guest OS running on the VM to directly interact with the GPU(s). The UCS server and ESXi hypervisor must also support this capability.

For workloads that do not require the resources of a full physical GPU or want the flexibility to partition and share the GPU resources among multiple virtual machine workloads, enterprises can use NVIDIA's **virtual GPU** (vGPU) technology as outlined in the next section.

GPU Passthrough support for a given Cisco UCS Server model on specific VMware vSphere versions can be verified using the VMware Compatibility guide by filtering on **VM Direct Path I/O for General GPU**:

<https://www.vmware.com/resources/compatibility/search.php?deviceCategory=vmdirect>

NVIDIA supports VMware DirectPath I/O for AI/ML workloads on NVIDIA A100 Tesla Core GPUs using the **publicly** available [NVAIE drivers](#) available on NVIDIA GPU Cloud (NGC) registry. You can also login to NGC and navigate to **CATALOG > Containers > GPU Drivers** to see the available drivers.

Note: Some vSphere features maybe unavailable or limited when using DirectPath I/O such as hot adding or removing of devices. Dynamic DirectPath I/O using the **Assignable Hardware** feature removes some of these limitations. For more information, see: <https://blogs.vmware.com/vsphere/2020/03/vsphere-7-assignable-hardware.html>. The **Assignable Hardware** feature is also used by NVIDIA's vGPU technology discussed in the next section.

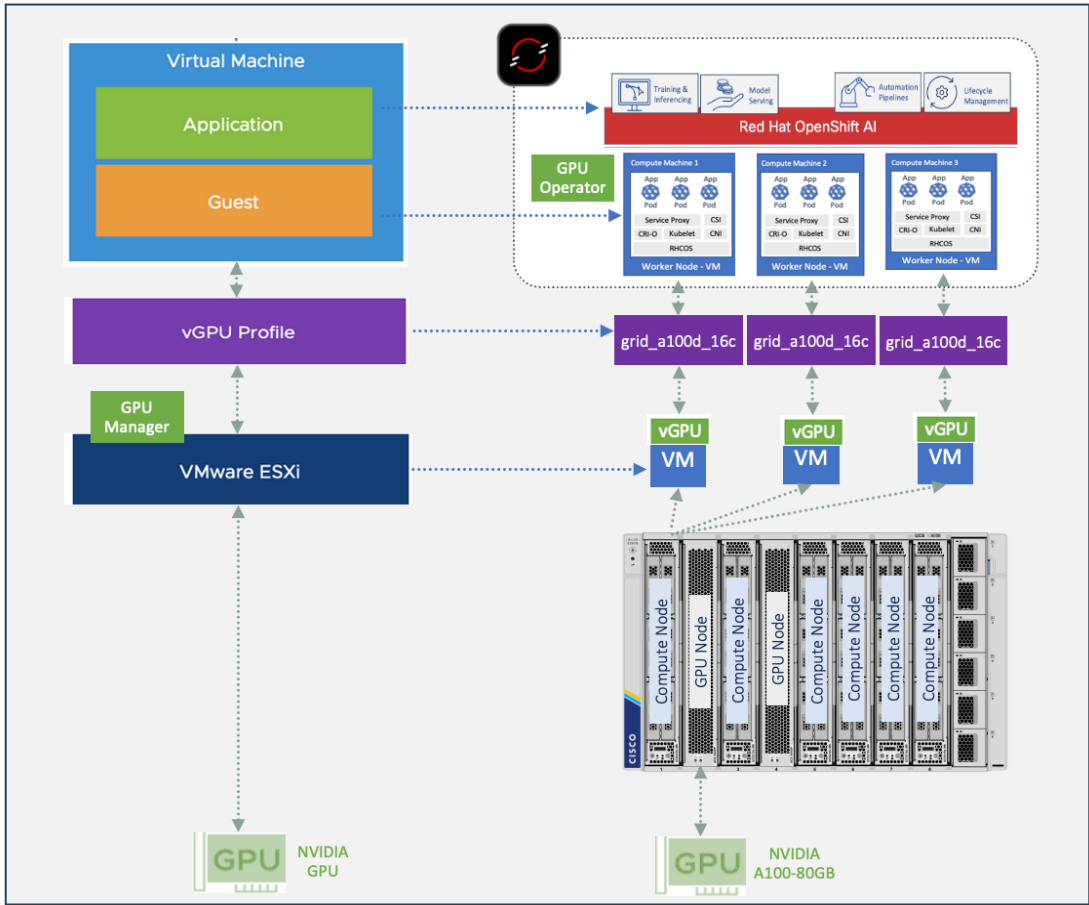
Virtual GPU

Virtual GPUs refer to the virtualization of GPU resources, enabling multiple virtual machine workloads to share the capabilities of a single physical GPU. Combined with Single Root I/O Virtualization (SR-IOV) and co-developed with VMware, NVIDIA's vGPU technology is a key capability in virtualized environments for providing unparalleled flexibility with near bare-metal performance. vGPUs can be assigned individually or combined as needed to meet the specific requirements of the AI/ML workload running on the VMs. If needed, a virtual GPU that uses the full GPU resources can also be assigned. As the workload demands change, vGPUs can be added or removed. This provides enterprises with the ability to right-size the GPU resources to the workload needs and maximize the utilization of their physical GPUs. Another advantage of vGPU is that each vGPU is allocated a slice of the physical GPU's time and any underutilized time is made available to other vGPUs, thereby maximizing overall GPU usage. Conversely, if all vGPUs are operating at maximum capacity, additional resources may need to be added to avoid a performance impact. Therefore, as you size an environment for AI/ML, it is important to understand and monitor the performance characteristics of the AI/ML workloads that will be sharing a GPU to maximize performance and GPU resource utilization.

Note: In vGPU mode, only the GPU compute time is shared or time-sliced and a given vGPU could potentially use all cores on the GPU if they're not being used, however the memory is statically partitioned.

[Figure 2](#) provides an architectural view of virtual GPUs that are available for AI/ML workloads to use. In this solution, vGPUs are used by the containerized AI/ML workloads running on Red Hat OpenShift worker nodes VMs.

Figure 2. Virtual GPU



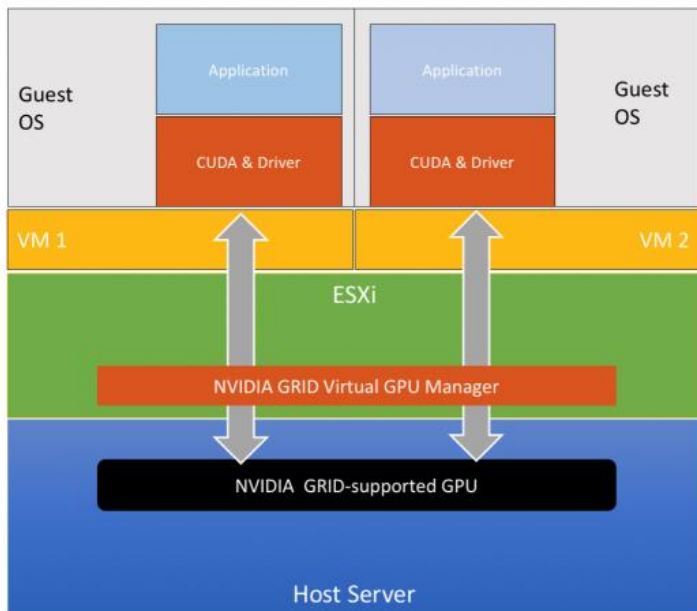
The vGPU profiles supported on the NVIDIA A100-80G are provided in [Table 3](#). All vGPU profiles on the same physical must have the same memory size. The table also lists the number of vGPUs per GPU for an NVIDIA A100-80GB GPU that can be allocated to VMs running AI/ML workloads.

Table 3. vGPU profiles Supported on NVIDIA A100-80GB Tensor Core GPUs

| NVIDIA vGPU Profile | Memory Buffer (MB) | Number of vGPUs per GPU |
|---------------------|--------------------|-------------------------|
| grid-a100d-80c | 81920 | 1 |
| grid-a100d-40c | 40960 | 2 |
| grid-a100d-20c | 20480 | 4 |
| grid-a100d-16c | 16384 | 5 |
| grid-a100d-10c | 10240 | 8 |
| grid-a100d-8c | 8192 | 10 |
| grid-a100d-4c | 4096 | 20 |

[Figure 3](#) shows the driver components and CUDA toolkit and libraries that are deployed when using vGPUs to support AI/ML workloads. The deployed drivers include a ESXi host driver, Guest OS driver, and CUDA toolkit deployed by GPU operator on OpenShift worker node VMs with GPUs.

Figure 3. NVIDIA vGPU Grid software (Source: VMware)



In summary, vGPUs in VMware vSphere environments enable efficient use of enterprise GPU resources and granular resource allocation to match the needs of VM workloads without compromising on performance. Enterprises can also continue to benefit from management ease and vSphere features like vMotion that VMware offers. This, coupled with the operational expertise and familiarity that IT teams already have with VMware, makes vGPUs a good design choice for enterprise AI/ML deployments.

Single Root I/O Virtualization (SR-IOV)

Single Root Input/Output Virtualization (SR-IOV) is an extension to the PCI Express (PCIe) specification that must be enabled for using vGPUs in a virtualized environment. It is a supported feature in VMware vSphere used in this solution. SR-IOV allows virtual machines to have direct access to the physical GPU resources, bypassing the hypervisor. The direct access maximizes GPU performance by reducing any hypervisor related overhead.

SR-IOV makes it possible to share physical GPU resources by creating multiple virtual functions (VFs) representing independent and dedicated slices of the GPU. Each VF can then be assigned to a given VM, enabling multiple VMs to share the GPU. VFs provide resource isolation which is critical for a given AI/ML VM workload to run without contention.

SR-IOV must be enabled on the servers to use vGPUs with the NVIDIA A100 Tensor Core GPUs used in this solution. This is enabled on the Cisco UCS servers using the server BIOS policy.

NVIDIA A100 GPUs also supports SR-IOV with up to 20VFs to enable sharing and virtualization of a single PCIe connection for use by multiple Virtual Machines.

Multi-Instance GPU

NVIDIA's Multi-Instance GPU (MIG) technology allows a physical GPU to be partitioned into multiple instances. Each of these instances can be independently used and managed to support different types of workloads. MIG is a feature first released on NVIDIA's A100 Tensor Core GPUs which partitions the GPU into as many as seven instances. MIG aims to ensure predictable performance by providing each GPU instance with dedicated resources like compute, memory, cache, and bandwidth. This means that even though a single physical GPU is being used, different types of workloads can run in parallel, but with complete isolation from workloads running on another instance. Both vGPU and MIG technologies are about optimally managing and using GPU resources

but unlike MIG, vGPUs enable multiple VMs to share the physical GPU resources while MIG partitions the GPU resources for use by multiple VMs.

Note: When using MIG on A100 GPUs, NVIDIA does not support direct point-to-point peer transfers from GPU to GPU (either PCIe or NVLink).

Note: MIG profiles supported on NVIDIA A100-80GB Tensor Core GPUs.

Table 4. NVIDIA GPU details

| NVIDIA MIG Profile Name | Memory Buffer (MB) | Number of vGPUs per GPU | Slices per vGPU | Compute Instances per vGPU | Corresponding GPU Instance Profile |
|-------------------------|--------------------|-------------------------|-----------------|----------------------------|------------------------------------|
| grid-a100d-7-80c | 81920 | 1 | 7 | 7 | MIG 7g.80gb |
| grid-a100d-4-80c | 40960 | 1 | 4 | 4 | MIG 4g.40gb |
| grid-a100d-3-40c | 40960 | 2 | 3 | 3 | MIG 3g.40gb |
| grid-a100d-2-20c | 20480 | 3 | 2 | 2 | MIG 2g.20gb |
| grid-a100d-1-10c | 10240 | 7 | 1 | 1 | MIG 1g.10gb |

For more information on NVIDIA MIG technology, see:

- <https://www.nvidia.com/en-us/technologies/multi-instance-gpu/>
- <https://docs.nvidia.com/datacenter/tesla/mig-user-guide/index.html>
- <https://docs.nvidia.com/datacenter/cloud-native/openshift/latest/mig-ocp.html> (for MIG on OpenShift)

MIG and vGPUs

NVIDIA’s MIG can be combined with the vGPU feature on NVIDIA A100 GPUs where the support for MIG was first introduced. vGPUs and MIG both fall under the NVAIE software umbrella, both are designed for compute intensive workloads in enterprise data centers and supported in all NVAIE releases. Previously, vGPUs for AI/ML type workloads were supported through NVIDIA’s Virtual Compute Server (vCS) software product and remnants of this may still be seen in some documentation and CLI outputs but for AI/ML and NVIDIA A100 GPU used in this solution, they are part of the NVAIE software set.

Note: The release numbering for NVAIE and vCS are different though constituent components maybe the same.

NVIDIA supports MIG-backed vGPUs in NVIDIA A100 GPUs that provide you with the flexibility to choose what best fits the needs of your environment. MIG-backed vGPUs enable enterprises to take advantage of MIG capabilities while leveraging the operational benefits of VMware vSphere.

Key differences between vGPU and MIG are:

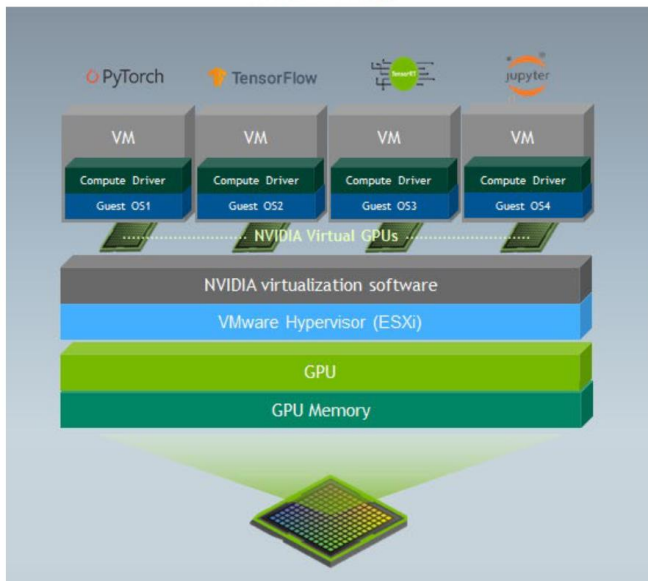
- Shared vs. Independent GPU compute resources
- Time sliced (software) compute vs. Spatial (hardware) partitioning
- Static versus Dynamic
- Homogeneous vs Heterogeneous partitions (instance profiles)

- QoS: vGPUs use round-robin, best-effort scheduler that allows workloads to use under-utilized GPU resources vs. MIG that is partitioned at the hardware level with isolated resources (memory, L2 cache) and therefore inherently, no contention.
- Varied Performance (can use underutilized GPU cycles) vs. Predictable Performance
- Management: vGPUs are managed by VMware vSphere or hypervisor IT teams that already have operational experience and can leverage hypervisor’s virtualization features. MIG is managed using **nvdiia-smi** tool.

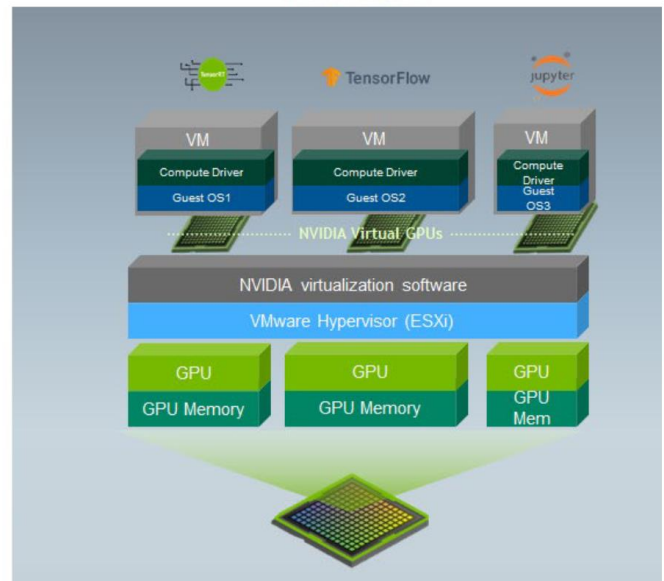
However, when used together, it can offer unparalleled flexibility, however the performance will depend on many factors such as the number of instances, type of workload, and so on.

[Figure 4](#) shows the high-level architecture of vGPU and MIG-backed vGPUs.

Figure 4. vGPU vs MIG (Source: VMware)
vGPU Only



MIG-vGPU



For more information on using MIG with vGPUs, see: <https://docs.nvidia.com/datacenter/tesla/mig-user-guide/index.html>

In this solution, the A100 GPUs are deployed using vGPU technology for validating the solution.

FlashStack VSI

The FlashStack VSI is a reference architecture for hosting a wide range of enterprise workloads (SAP, Oracle, SQL, HPC, VDI) on virtualized infrastructure in enterprise data centers. The FlashStack CVDs not only provide comprehensive design and implementation guidance, but also Infrastructure as Code (IaC) automation to accelerate enterprise data center infrastructure deployments. The designs incorporate product, technology, and industry best practices to deliver a highly-available, scalable, and flexible architecture.

The key infrastructure components used in FlashStack VSI designs for compute, network, and storage are:

- Cisco Unified Computing System (Cisco UCS) Infrastructure
- Cisco Nexus 9000 switches
- Cisco MDS 9000 SAN switches

- Pure Storage FlashArray

The design is flexible and can be scaled up or scaled out without sacrificing feature/functionality. FlashStack solutions are built and validated in Cisco labs to ensure interoperability and minimize risk in customer deployments. The CVDs saves enterprise IT teams valuable time that would otherwise be spent on designing and integrating the solution in-house.

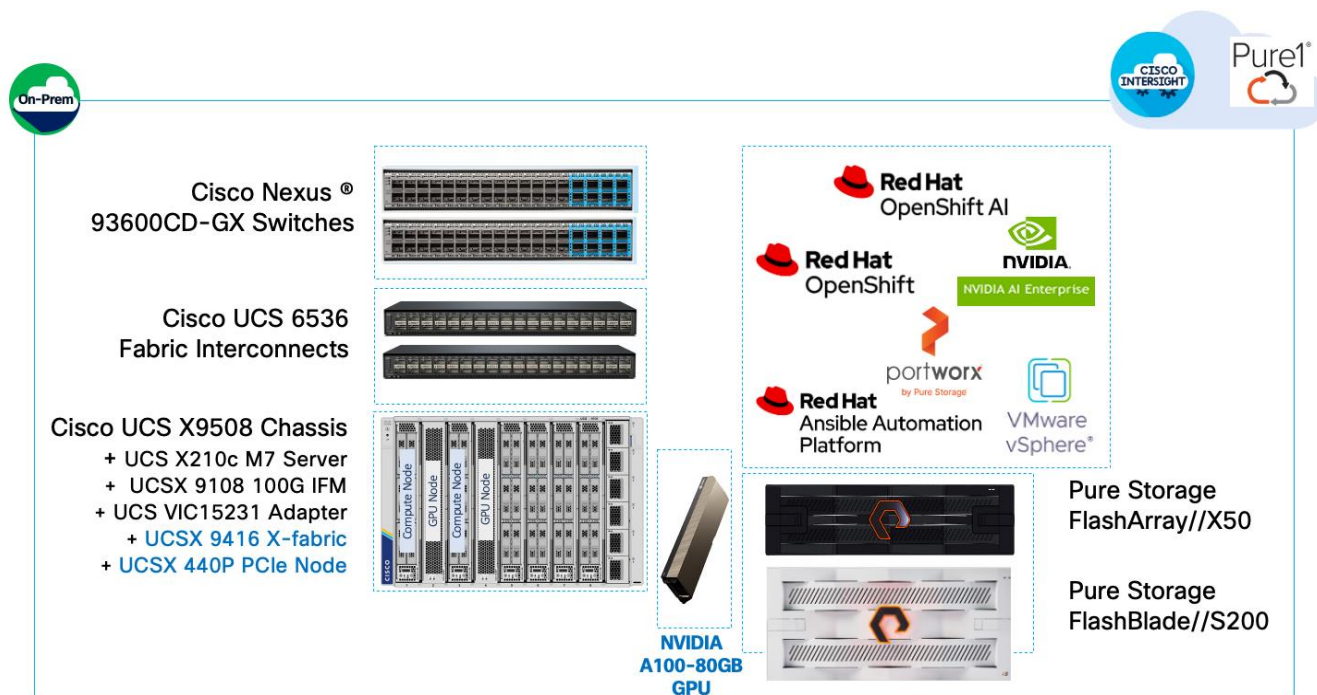
All FlashStack CVDs are available on Cisco Design Zone located here:

<https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/data-center-design-guides-all.html#FlashStack>

FlashStack for AI

The FlashStack solution in this document uses the [FlashStack VSI with VMware vSphere 8.0](#) as a foundational infrastructure architecture, offering enterprises a robust, scalable portfolio of compute, networking and storage options for their AI initiatives. Using this solution, enterprises can quickly start on their AI journey and incrementally expand as the enterprise needs grow. [Figure 5](#) shows the components in the solution.

Figure 5. Solution Components



To support AI/ML workloads with MLOps, the **FlashStack for AI** design adds the following components to the foundational architecture documented in the FlashStack VSI with VMware vSphere 8.0 solution.

- Cisco UCS X440p PCIe nodes, capable of hosting up to four GPUs (only some models). Each PCIe node is paired with a Cisco UCS compute node, specifically the Cisco UCS X210c M7 server but a Cisco UCS X410c can also be used) with a UCS. Connectivity between the compute node and PCIe node requires a PCIe mezzanine card on the compute node and a pair of X-fabric modules on the Cisco UCS X9508 server chassis.
- NVIDIA GPUs (A100-80GB) for accelerating AI/ML workloads and model delivery pipeline.
- NVIDIA AI Enterprise software.
- Red Hat OpenShift for Kubernetes based container orchestration and management.

-
- Portworx from Pure Storage for persistent storage (backed by Pure Storage FlashArray).
 - Pure Storage FlashBlade for S3 compatible object store.
 - Red Hat OpenShift AI for MLOps.

The next sections provide a brief overview of the new components in the solution that have not been previously discussed in this document.

Cisco UCS X440p PCIe Node

The Cisco UCS X440p PCIe node (UCSX-440P-U) is the first PCIe node supported on a Cisco UCS X-Series fabric. It is part of the Cisco UCS-X Series modular system, managed using Cisco Intersight and integrated to provide GPU acceleration for workloads running on Cisco UCS compute (X210c, X410c) nodes. GPUs can be installed on the PCIe node and then paired with a compute node in an adjacent slot to support AI/ML, VDI, and other workloads that require GPU resources. GPUs. The PCIe node requires riser cards to support different GPU form factors, either full height, full length (FHFL) or half height, half length (HHHL) GPUs as outlined below:

- **Riser Type A:** Supports 1 x 16 PCIe connectivity for FHFL GPUs (**UCSX-RIS-A-440P**)
- **Riser Type B:** Supports 1 x 8 PCIe connectivity for HHHL GPUs (**UCSX-RIS-B-440P**)

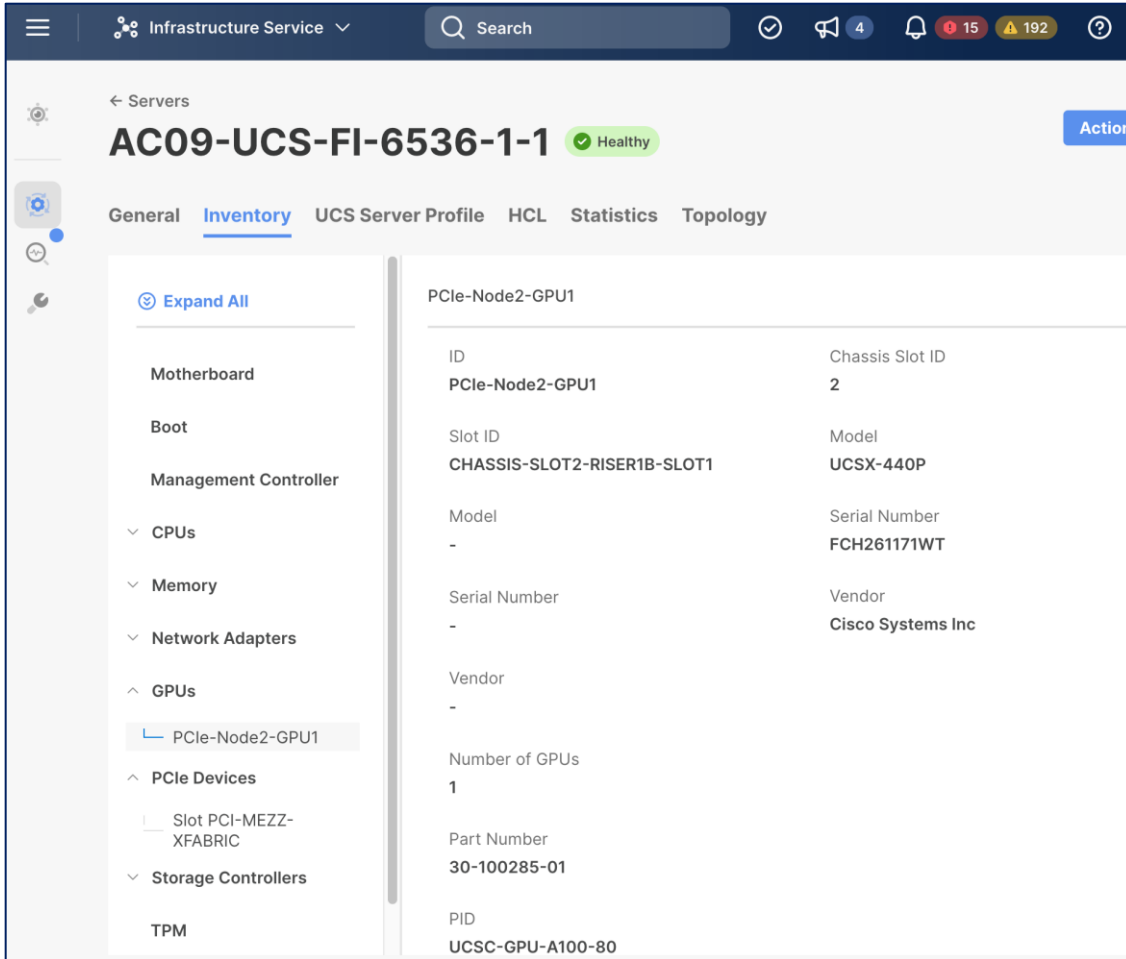
Each PCIe node supports a maximum of two riser cards, with each riser card capable of supporting up to:

- 1 x 16 FHFL dual slot PCIe cards, one per riser card for a total of two FHFL cards
- 1 x 8 HHHL single slot PCIe card, two per riser card for a total of four HHHL cards

Note: Each PCIe node must have the same type of risers and GPUs. You cannot mix and match riser types and GPU types in the same PCIe node.

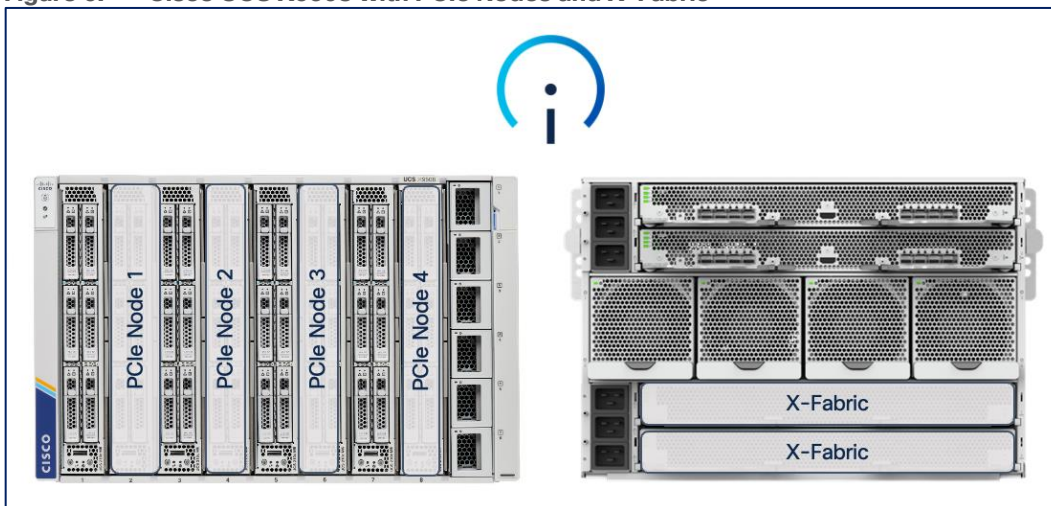
The NVIDIA A100-80G Tesla Core GPU (**UCSX-GPU-A100-80**) deployed in this solution is a FHFL GPU and uses the Type A riser card.

When deployed in a slot adjacent to the compute node, the PCIe node will be recognized and seen as an extension of the compute node in Cisco Intersight as shown below. The PCIe node is in slot 2, as indicated by the node name: PCIe-**Node2**-GPU1 but appears as a GPU adapter on server in slot 1 with which it is paired.



A single Cisco UCS X9508 is a 7RU chassis with eight slots that can support up to 4 compute nodes and 4 PCIe nodes as shown in [Figure 6](#).

Figure 6. Cisco UCS X9508 with PCIe Nodes and X-Fabric

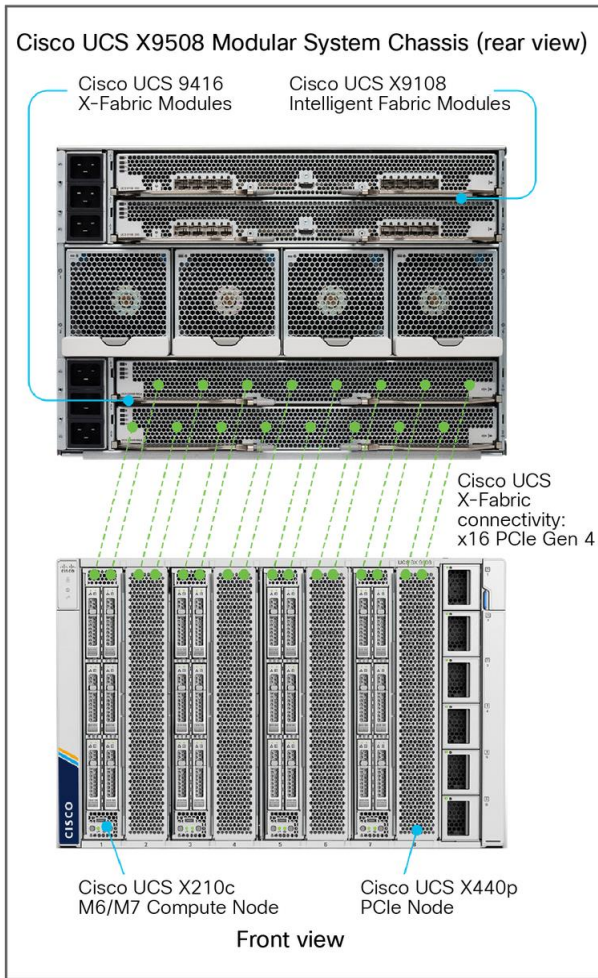


To enable connectivity between GPUs in PCIe nodes and the compute nodes they are paired, the following additional components are required to enable PCIe Gen4 connectivity between compute and GPU nodes.

- PCIe mezzanine card on the compute node (**UCSX-V4-PCIME**)

- Pair of UCS X9416 X-fabric modules on UCS X-series server chassis (**UCSX-F-9416**) (see [Figure 7](#))
- The Cisco UCS X9508 Chassis has no midplane design, provides fewer obstructions for better airflow. The vertically oriented Cisco UCS X210c or X410c compute nodes and the X440p PCIe nodes connect directly to horizontally oriented X-Fabric modules, located at the back of the chassis (see [Figure 7](#)). The innovate design enables Cisco UCS X-Series to easily upgrade to newer technologies and hardware without requiring forklift upgrades.

Figure 7. Cisco UCS X-Fabric Connectivity



As stated earlier, each PCIe node allows you to add up to four HHL GPUs to accelerate workloads running on either a Cisco UCS X210c or UCSX410c compute node. This provides up to 16 GPUs per chassis. As of the publishing of this document, the following GPU models are supported on a Cisco UCS X440p PCIe node.

Table 5. GPU Options on Cisco UCS X-Series Server System

| GPU Model | GPUs Supported per PCIe node (UCS X440p) | GPUs Supported on UCS-X9508 server chassis with 4 x UCS X210c servers |
|-----------------------------|--|---|
| NVIDIA A100 Tensor Core GPU | Max of 2 | Max of 8 |
| NVIDIA A16 GPU | Max of 2 | Max of 8 |
| NVIDIA A40 GPU | Max of 2 | Max of 8 |

| GPU Model | GPUs Supported per PCIe node (UCS X440p) | GPUs Supported on UCS-X9508 server chassis with 4 x UCS X210c servers |
|---------------------------------|--|---|
| NVIDIA T4 Tensor Core GPUs | Max of 4 | Max of 24* |
| NVIDIA H100 Tensor Core GPU | Max of 2 | Max of 8 |
| NVIDIA L40 GPU | Max of 2 | Max of 8 |
| NVIDIA L4 Tensor Core GPU | Max of 4 | Max of 16 |
| Intel® Data Center GPU Flex 140 | Max of 4 | Max of 24* |
| Intel Data Center GPU Flex 170 | Max of 2 | Max of 8 |

*Using the optional front mezzanine GPU adapter (UCSX-X10C-GPUFM-D) on Cisco UCS X210c compute node

If additional GPUs are needed, up to two GPUs can be added using an optional GPU front mezzanine card on the Cisco UCS X210c or UCS X410c compute nodes. Only two GPU models are currently supported in this configuration but enables up to have up to 24 GPUs per chassis.

The product IDs for enabling GPU acceleration using Cisco UCS X440p PCIe nodes are summarized in [Table 6](#).

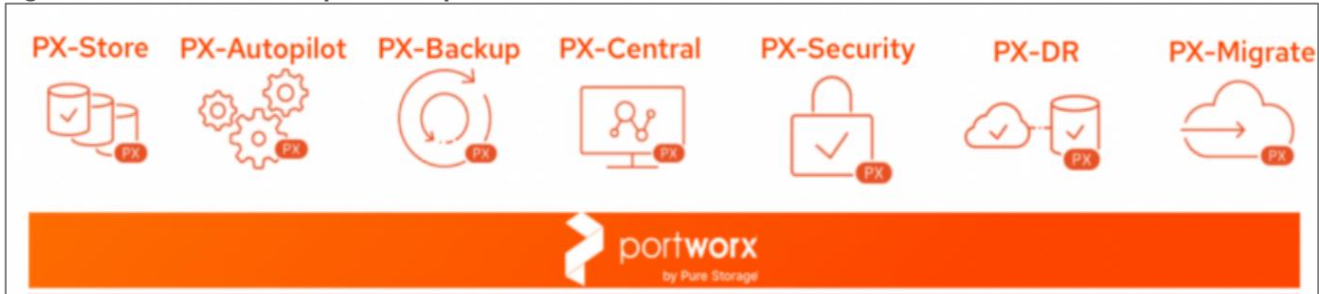
Table 6. Product IDs for GPU acceleration using PCIe Node

| Component | PID |
|--|------------------|
| UCS X-Series Gen 4 PCIe node | UCSX-440P-U |
| Riser A for 1x dual slot GPU per riser, 440P PCIe node <ul style="list-style-type: none"> Riser 1A (controlled with CPU1 on UCS X210c) Riser 2A (controlled with CPU2 on UCS X210c) | UCSX-RIS-A-440P |
| Riser B for 2x single slot GPUs per riser, 440P PCIe node <ul style="list-style-type: none"> Riser 1B (controlled with CPU1 on UCS X210c) Riser 2B (controlled with CPU2 on UCS X210c) | UCSX-RIS-B-440P |
| UCS PCI Mezz card for X-Fabric connectivity | UCSX-V4-PCIME |
| UCS X-Fabric module for UCS-X9508 chassis | UCSX-F-9416 |
| NVIDIA A100 Tensor Core GPUs, PASSIVE, 300W, 80GB | UCSX-GPU-A100-80 |
| NVIDIA A16 GPUs, 250W, 4x16GB | UCSX-GPU-A16 |
| NVIDIA A40 GPUs RTX, PASSIVE, 300W, 48GB | UCSX-GPU-A40 |
| NVIDIA T4 Tensor Core GPUs 75W, 16GB | UCSX-GPU-T4-16 |
| NVIDIA H100 Tensor Core GPU, 350W, 80GB (2-slot FHFL GPU) | UCSX-GPU-H100-80 |
| NVIDIA L40 GPU, 300W, 48GB | UCSX-GPU-L40 |
| NVIDIA L4 Tensor Core GPU, 70W, 24GB | UCSX-GPU-L4 |

Portworx Enterprise for Kubernetes Persistent Storage

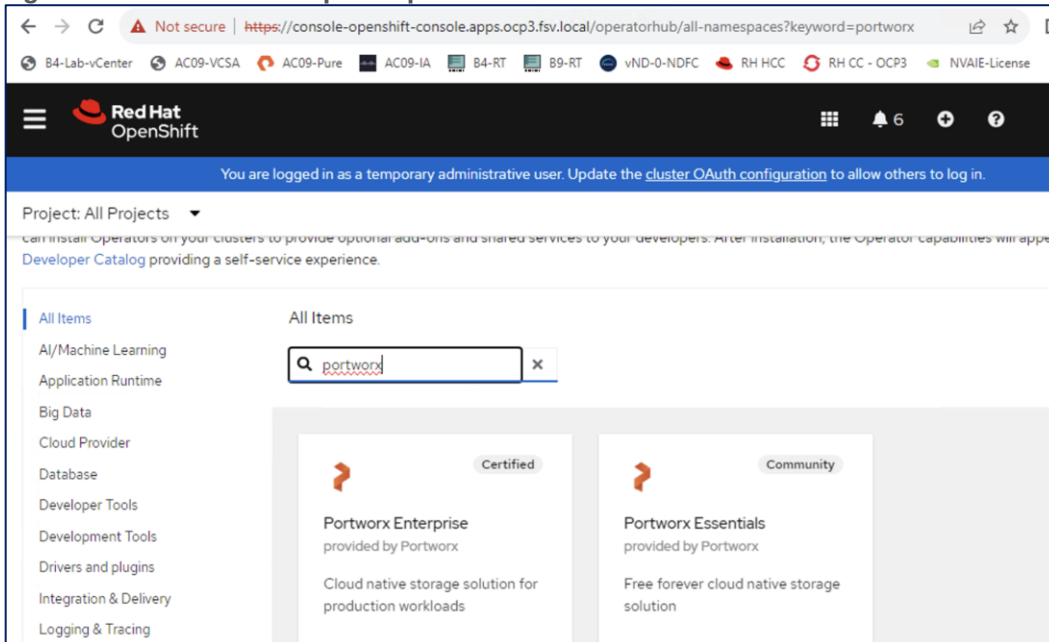
Portworx Enterprise provides container optimized persistent storage with data security and disaster recovery for AI/ML workloads and machine learning pipelines managed by OpenShift AI. Portworx Enterprise consists of the following subcomponents (Figure 8) that can be added as needed to provide additional functionality.

Figure 8. Portworx Enterprise Components



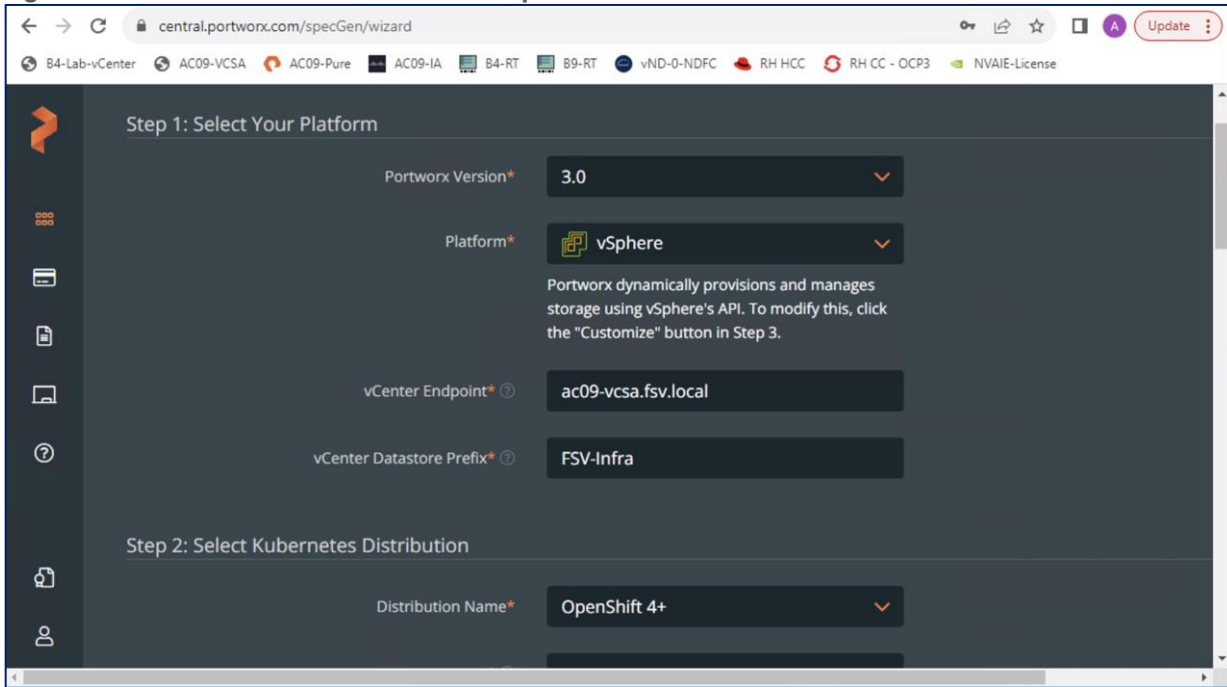
Portworx Enterprise is a multi-cloud solution, providing cloud-native storage for workloads running anywhere, from on-prem to cloud to hybrid/multi-cloud environments. Portworx Enterprise is deployed on Red Hat OpenShift cluster using the Red Hat certified Portworx Enterprise operator, available on Red Hat's OperatorHub.

Figure 9. Portworx Enterprise Operator

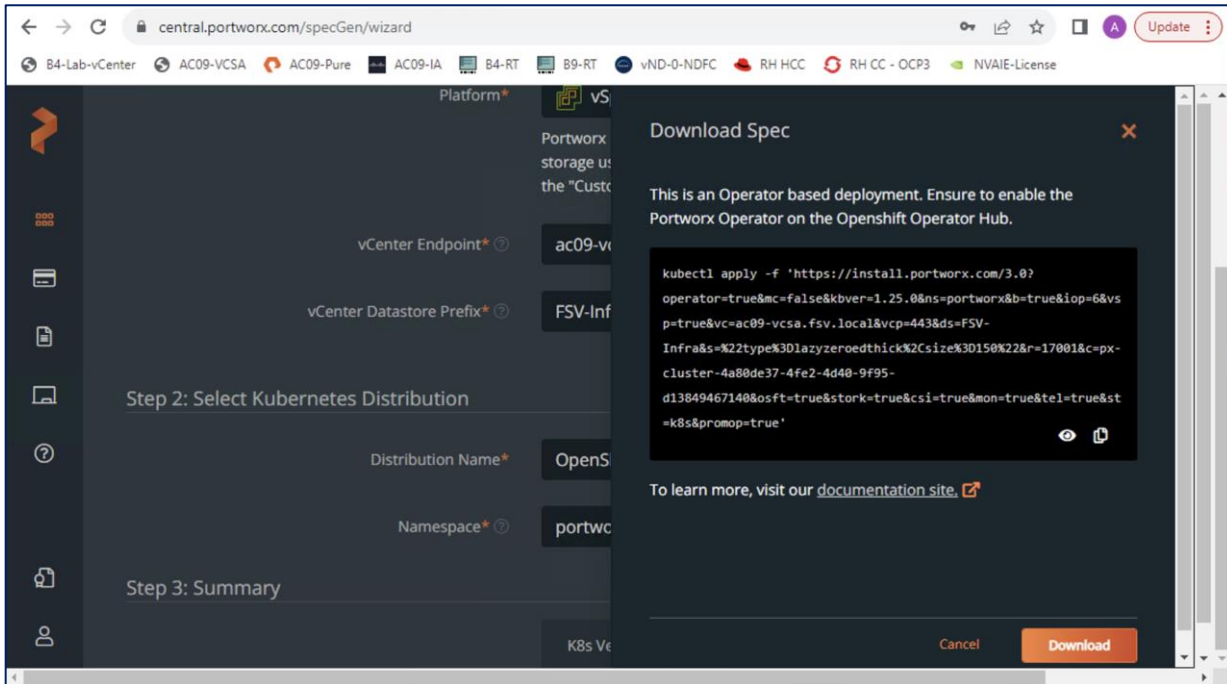


Portworx Enterprise leverages **Portworx Central**, a cloud-based multi-cluster management that provides licensing, backups, and other functionality. It is used in this solution to generate an environment specific **StorageCluster** spec that can be downloaded to quickly provision and license Portworx in Red Hat OpenShift (as shown in Figure 10). The **StorageCluster** is a Kubernetes Custom Resource Definition (CRD) that specifies how a Portworx storage cluster should be configured and deployed. For more information on StorageCluster configuration, see: <https://docs.portworx.com/portworx-enterprise/reference/CRD/storage-cluster>.

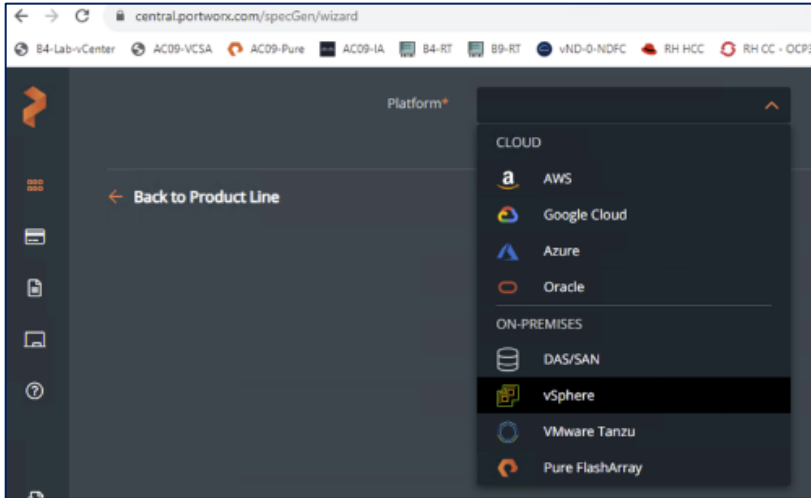
Figure 10. Provision Portworx in Red Hat OpenShift



When the generated **ClusterSpec** is applied, Portworx Operator will deploy a Portworx cluster corresponding to the specification in the StorageCluster object. The CRD can also be generated manually but Portworx Central makes it easier.



The **ClusterSpec** generated also includes information about the VMFS datastore or volume that will be used to back the persistent volumes in Kubernetes. The figure below shows the on-prem options that Portworx Central can be used to generate a cluster configuration.



The Portworx Enterprise Operator will also deploy a **Console Plugin** on Red Hat OpenShift, enabling Portworx to be managed directly from the OpenShift cluster console. Portworx appears as a menu option in the main navigation menu, providing information such as node usage and capacity information, persistent volumes provisioned, and other details as shown below.

| Node Summary | | | | |
|---------------------------|-------------|-----------|-----------------|-----------------------|
| Name | IP | Status | PX Version | Used / Total Capacity |
| ocp3-qg7j6-worker-0-fk429 | 10.119.3.13 | status ok | 3.0.3.0-1f99161 | 8GiB / 150GiB |
| ocp3-qg7j6-worker-0-j7ps2 | 10.119.3.15 | status ok | 3.0.3.0-1f99161 | 8GiB / 150GiB |
| ocp3-qg7j6-worker-0-fz8rw | 10.119.3.14 | status ok | 3.0.3.0-1f99161 | 8GiB / 150GiB |

1 - 3 of 3 << < 1 of 1 > >>

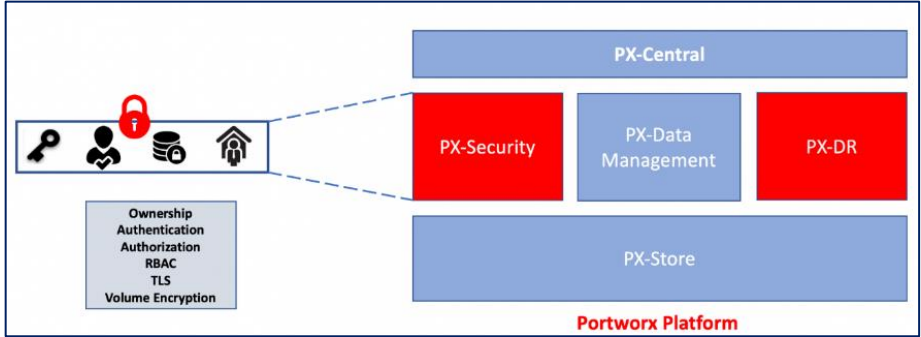
| Volumes | | | | | | | |
|--|-------------|--------------|--------|-----------------|---------|-----------|--|
| Name ↓ | Namespace ↓ | PVC | Status | Attached Node ↓ | Replica | Capaci... | |
| pvc-aa6f2f93-80a7-4091-8b00-e69c2b8bf2b3 | portworx | px-check-pvc | UP | - | 3 | 2GiB | |

Portworx, backed Pure Storage FlashArray provides some key advantages:

- Pure Storage FlashArray provides all-flash storage backed by an enterprise-class array with six-nines reliability, data-at-rest encryption, and industry-leading data-reduction technology. Although Portworx supports any storage type including Direct Attached Storage (DAS) and array-based storage, using Portworx replicas to ensure data availability for application pods across nodes, then having all replicas provisioned from the same underlying FlashArray will multiply your standard data-reduction rate, for the application data, by the number of replicas for the persistent volume.
- Portworx combined with Pure Storage FlashArray can be used as a cloud storage provider. This allows administrators to store your data on-premises with FlashArray while benefiting from Portworx cloud drive features, automatically provisioning block volumes, Expanding a cluster by adding new drives or expanding existing ones with support for Portworx Autopilot. Pure Storage FlashArray with Portworx on Kubernetes can attach FlashArray as a Direct Access volume. Used in this way, Portworx directly provisions FlashArray volumes, maps them to a user PVC, and mounts them to pods. FlashArray Direct Access volumes support the CSI operations like filesystem operations. snapshots and QOS.

Portworx Security secures the containers with access controls and encryption. It includes cluster wide encryption and BYOK encryption with storage class or container granular based. Role based control for Authorization, Authentication, Ownership and integrates with Active Directory and LDAP.

Figure 11. Portworx RBAC

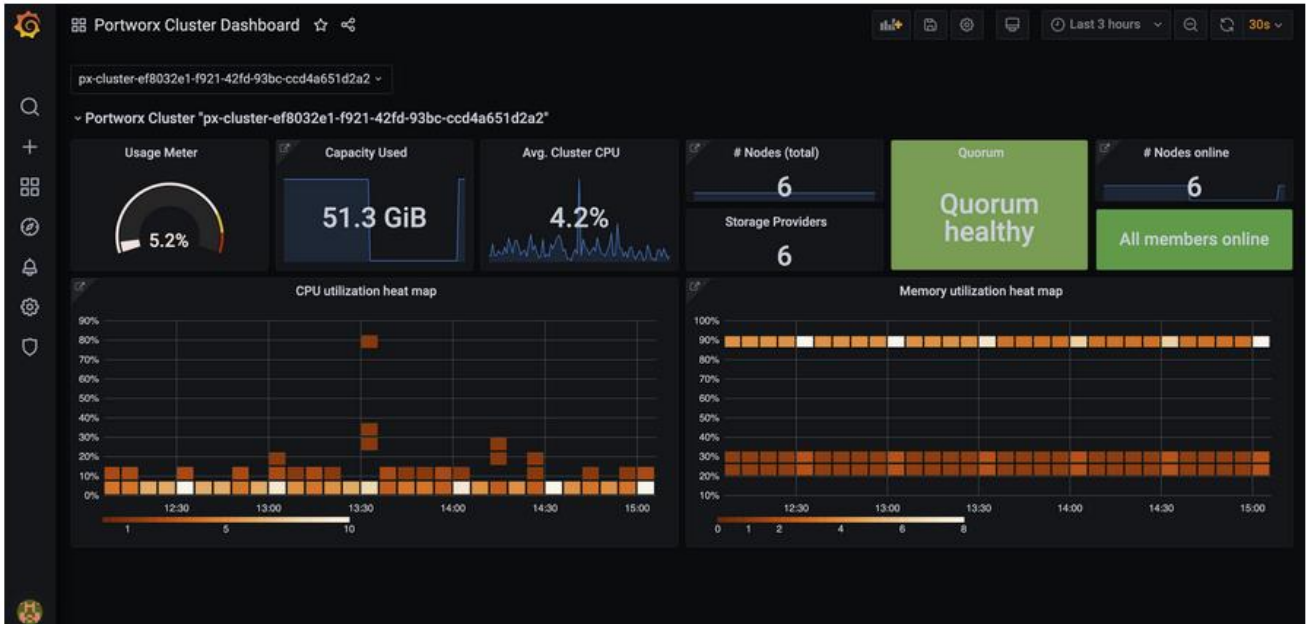


Portworx Disaster Recovery (DR) offers a near RPO-zero failover across data centers in a metropolitan area network and in addition to HA within a single datacenter. PX-DR offers continuous incremental-backups, with the ability to set data protection policies at the container level. Portworx also supports Asynchronous DR.

Monitoring Portworx Cluster

Portworx cluster can be monitored by Prometheus to collect data, Alertmanager to provide notifications and Grafana to visualize your data. Prometheus Alertmanager handles alerts sent from the Prometheus server based on rules you set. You can connect to Prometheus using Grafana to visualize your data as in [Figure 12](#).

Figure 12. Portworx monitoring using Grafana



For more information, see the Portworx documentation: <https://docs.portworx.com/portworx-enterprise>.

Pure Storage FlashBlade - Designed for AI

Pure Storage® FlashBlade//S is a scale-out storage system that is designed to grow with your unstructured data needs for AI/ML, analytics, high performance computing (HPC), and other data-driven file and object use cases in areas of healthcare, genomics, financial services, and more. FlashBlade//S provides a simple, high-performance solution for Unified Fast File and Object (UFFO) storage with an all-QLC based, distributed architecture that can support NFS, SMB, and S3 protocol access. The cloud-based Pure1® data management platform provides a single view to monitor, analyze, and optimize storage from a centralized location.

FlashBlade//S is the ideal data storage platform for AI, as it was purpose-built from the ground up for modern, unstructured workloads and accelerates AI processes with the most efficient storage platform at every step of your data pipeline. A centralized data platform in a deep learning architecture increases the productivity of AI engineers and data scientists and makes scaling and operations simpler and more agile for the data architect.

An AI project that uses a single-chassis system during early model development can expand non-disruptively as data requirements grow during training and continue to expand as more live data is accumulated during production. FlashBlade//S systems can scale from 168TB to 19.2PB of physical capacity (about 180TB to 25PB of usable storage) and are available in either capacity-optimized or performance-optimized configurations.

Figure 13. FlashBlade//S - Scale and Capacity



FlashBlade//S supports data scientists developing models and training jobs by enabling intensive random I/O loads and checkpoint writes, and production I/O in a single system. It automatically adapts data placement and

I/O distribution to utilize all available resources effectively. Each chassis contains redundant fabric I/O modules (FIOMs) that interconnect its blades with 8 x 100GbE ports per chassis, there is adequate bandwidth for data scientists to experiment, even as training jobs impose heavy I/O loads.

The technical specifications of the FlashBlade//S system is provided in [Table 7](#).

Table 7. FlashBlade//S specifications

| | Scalability | Physical | Capacity | Connectivity |
|---------------|--|--|---------------------------------|---|
| FlashBlade//S | Start with a minimum of 7 blades and scale up to 10 blades in a single chassis | Up to 4 DirectFlash Modules per blade (24TB or 48TB DirectFlash Modules) | Uplink networking 8 x 100GbE | 5U per chassis Dimensions: 8.59" x 17.43" x 32.00" x 32.00" |
| | Independently scale capacity and performance with all-QLC architecture | Up to 192TB per blade | Future-proof midplane | 2,400W (nominal at full configuration) |

Purity//FB

Purity is the heart of FlashBlade//S, enabling it to scale tremendously in capacity and performance. Purity//FB is an all-inclusive software with enterprise-grade data services. Purposefully architected to run on FlashBlade's all-flash hardware, Purity//FB has a variable block metadata engine and scale-out metadata architecture. It can handle billions of files and objects and delivers unmatched performance for any workload, whether it's with sequential or random access. Purity//FB delivers a rich set of enterprise capabilities including compression, global erasure coding, always-on encryption, SafeMode, file replication, object replication, and multiple other features.

Solution Design

This chapter contains the following:

- [Solution Architecture](#)
- [Solution Topology](#)
- [Design Details](#)

The FlashStack VSI for AI with Red Hat OpenShift AI solution aims to address the following design goals:

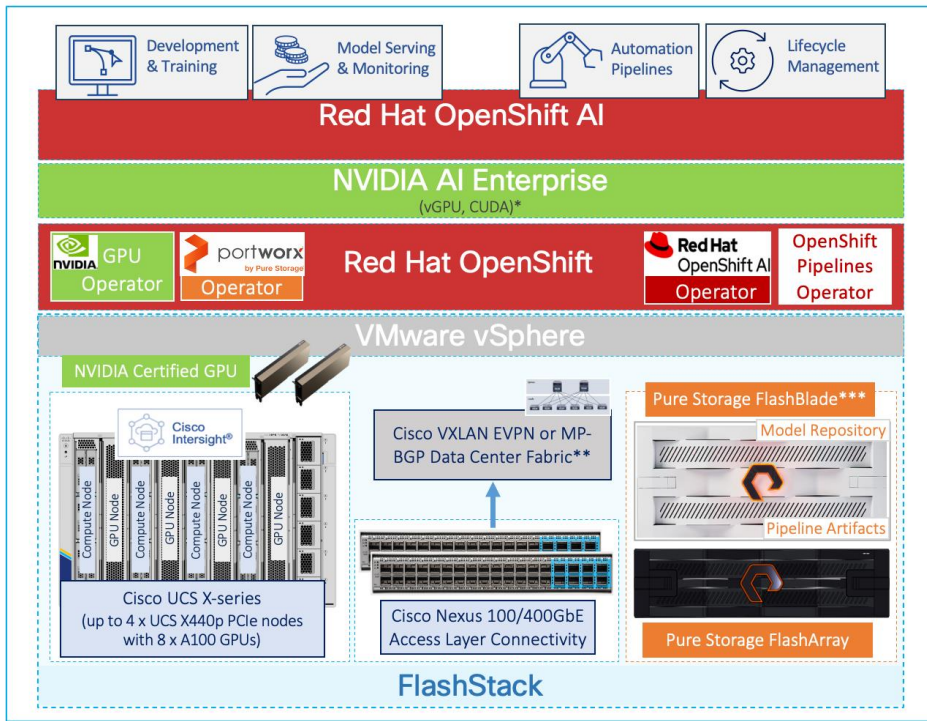
- Best-practices based design for AI/ML workloads, incorporating product, technology, and industry best practices.
- Simplify and streamline operations for AI/ML. Ease integration into existing deployments and processes.
- Flexible design with options for tools, technologies and individual components and sub-systems used in the design can be modified to adapt to changing requirements (for example, storage access, network design)
- Modular design where sub-system components (for example, links, interfaces, model, platform) can be expanded or upgraded as needed.
- Scalable design: As deployments grow, FlashStack VSI can be scaled up or out to meet enterprise needs. Each FlashStack VSI deployment unit can also be replicated as needed to meet scale requirements.
- Resilient design across all layers of the infrastructure with no single point of failure.

The upcoming sections cover the solution architecture and design that meets these design requirements.

Solution Architecture

A high-level architecture of the FlashStack AI MLOps solution using Red Hat OpenShift AI is shown in [Figure 14](#).

Figure 14. Solution Architecture



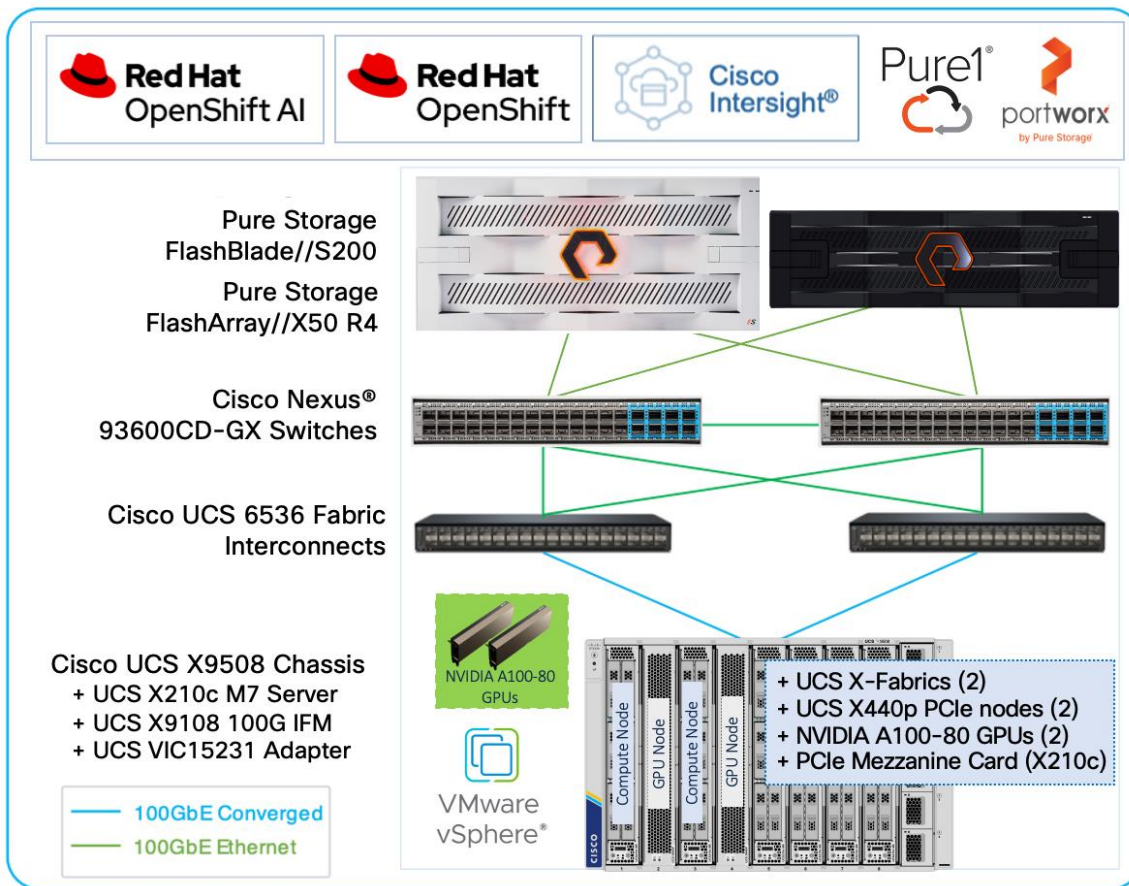
* Components from the NVAIE software suite used in this solution
 ** See Cisco Nexus [Data Center Networking Blueprint for AI/ML Applications](#) - not validated as a part of this solution
 *** Pure Storage FlashBlade is directly accessed from Red Hat OpenShift AI

The solution uses the foundational infrastructure from the latest FlashStack VSI CVD. It expands and builds on the design to enable a FlashStack VSI design for AI using NVIDIA GPUs, optional NVIDIA AI Enterprise software, Pure Storage FlashBlade, Red Hat OpenShift, and MLOps provided by Red Hat OpenShift AI.

Solution Topology

The high-level infrastructure design and topology built in Cisco labs for validating this FlashStack VSI solution for AI with OpenShift AI is shown in [Figure 15](#).

Figure 15. Solution Topology



Design Details

As stated earlier, the infrastructure design in this solution is based on the latest [FlashStack VSI solution](#) for Enterprise data centers. The FlashStack VSI solution is a 100Gb Ethernet solution that supports unified file and block storage with options for iSCSI or Fibre Channel (FC) boot and storage access using FC and FC-NVMe or IP/Ethernet-based storage using iSCSI, NVMe-TCP, NVMe-RoCEv2, NFS or SMB. The solution uses VMware vSphere 8.0 running on the UCS M7 servers with the latest Intel processors. The compute and storage components in the solution will connect into Cisco Nexus 9000 series switches capable of 100GbE/400GbE networking. These access layer switches will integrate into a larger Enterprise data center fabric using either multiple 100GbE or 400GbE uplinks. The following blueprint and CVD provides Cisco’s recommendations for building a high-speed, low latency, lossless data center fabric for AI/ML.

- [Cisco Data Center Networking Blueprint for AI/ML Applications](#)
- [Cisco Validated Design for Data Center Networking Blueprint for AI/ML Applications](#)

The solution incorporates design, technology, and product best practices to deliver a highly scalable and flexible architecture with no single point of failure.

The FlashStack for AI solution in this document, extends the FlashStack VSI design to support AI/ML workloads and deliver ML models in production. The design incorporates the necessary components to deliver a robust design for enterprise AI/ML initiatives as outlined in the next sections.

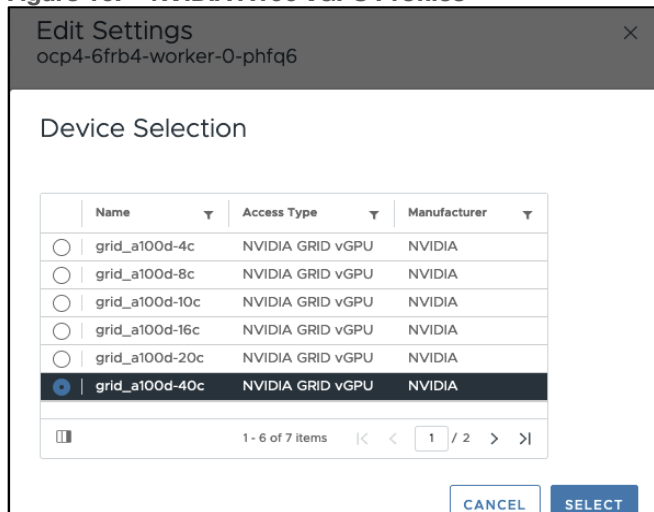
Cisco UCS servers with NVIDIA GPUs

The design uses the Cisco UCS X-Series server chassis with NVIDIA GPUs to provide the compute and GPU resources in the solution. The Cisco UCS X9508 server chassis is a 7RU chassis with eight slots where UCS servers and PCIe nodes can be deployed with GPUs to provide a total of 8 FHFL GPUs or 16 HHHL GPUs per chassis. The Cisco UCS X9508 supports up to four PCIe nodes, with each PCIe node paired with a compute node, either a Cisco UCS X210c or X410c M6/M7 server. This design was validated using NVIDIA A100-80GB Tesla Core GPUs; up to two NVIDIA A100 GPUs can be installed on a single PCIe node. Alternatively, Cisco UCS C-series rackmount servers can also be used. A UCS C-series M7 server with Intel processors can support up to 3 x A100-80 GPUs.

Virtual GPU Deployment in VMware vSphere

The NVIDIA A100 GPUs in the FlashStack VSI design are deployed in Virtual GPU (vGPU) mode to provide enterprises with more flexibility to right-size the resources to the needs of the AI/ML workloads with near bare-metal performance. By using vGPUs, multiple VMs can share the physical GPU resource. To enable vGPU mode, NVIDIA provides a GPU manager OVA that can be downloaded and deployed on the vSphere cluster. The GPU manager will integrate with VMware vCenter and dynamically deploy a vSphere plugin. The plugin enables vSphere administrators to manage the NVIDIA GPU directly from vCenter. Through the vSphere plugin and GPU manager, vSphere administrators can download NVAIE host drivers that are required for vGPU mode. The same host driver supports graphics (default) and vGPU functionality so the GPU must be reconfigured for the latter. In vGPU mode, the GPU will present several profiles to pick based on the amount of GPU memory that each profile uses. Each VM running on the server with the GPU can pick one of the profiles to use. In vGPU mode, only homogeneous profiles are supported so the first profile selected and assigned to a virtual machine will be the only profile that can be assigned to subsequent VMs. The vGPU profile supported on the NVIDIA A100 in vGPU mode is shown in [Figure 16](#).

Figure 16. NVIDIA A100 vGPU Profiles



The last profile (**grid_a100d-80c**) is not shown in this screenshot. Depending on the profile selected, you can assign anywhere from 1 vGPU to 20vGPUs on an A100-80GB GPU. Other NVIDIA GPU models may have different profile options based on the amount of physical GPU memory it has.

Red Hat OpenShift Application Platform

Red Hat OpenShift Application platform clusters deployed in Enterprise data centers provides secure, enterprise-class Kubernetes orchestration and container management for developing, deploying, and managing cloud-native applications. Red Hat OpenShift clusters can be deployed on bare-metal servers or in virtualized

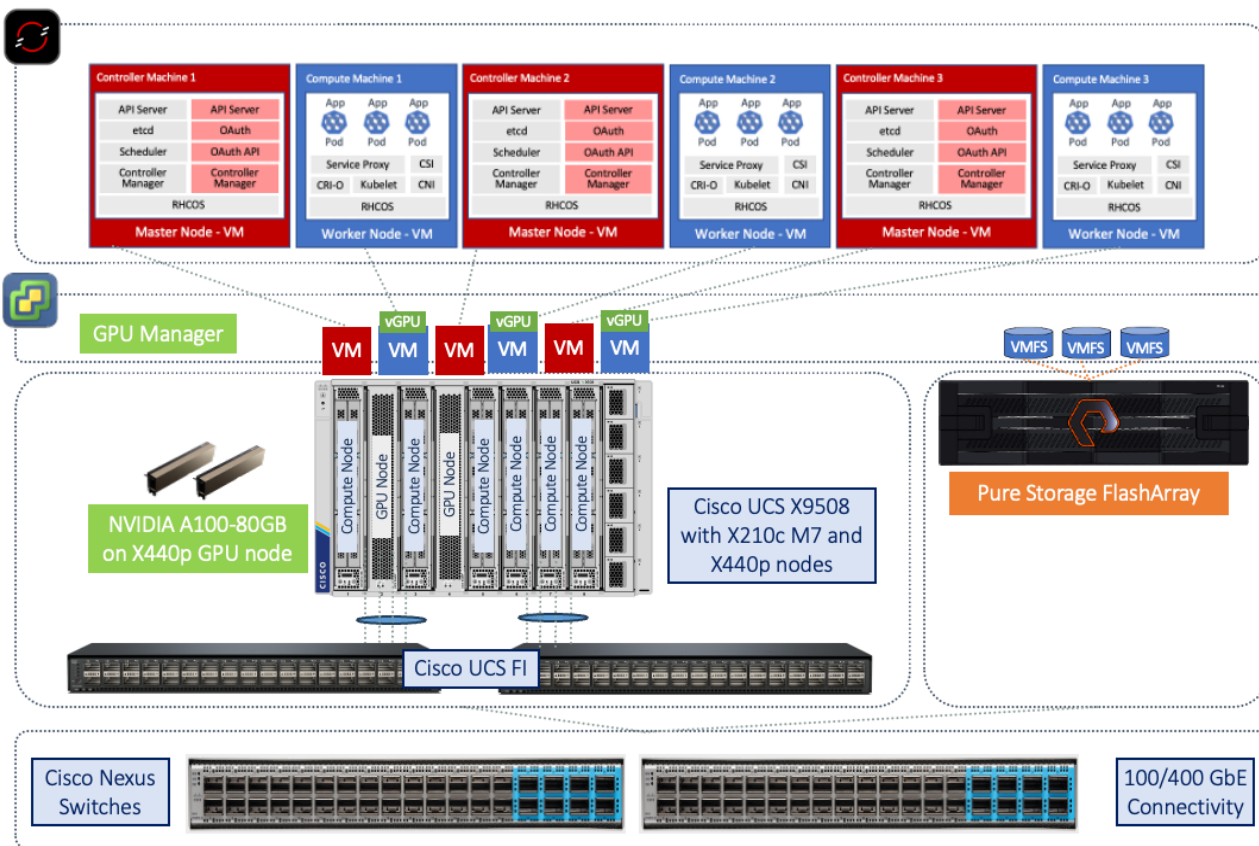
environments as a managed service or self-managed, either on-prem or in the public cloud. In this solution, Red Hat OpenShift is deployed as a self-managed service running on a VMware vSphere cluster.

To deploy and manage self-managed OpenShift clusters, Enterprises can use either a cloud-based [Red Hat Hybrid Cloud Console](#) (HCC) or Advanced Cluster Management (ACM). The OpenShift clusters deployed for validating the solution was deployed using Red Hat Hybrid Cloud Console. Red Hat Hybrid Cloud Console provides access to the latest installer package and other tools necessary for securely accessing and managing the cluster in different environments (bare metal, VMware vSphere, AWS, Azure, GCP, and so on) using an installer/management workstation.

To install an OpenShift cluster in a VMware vSphere environment, Red Hat provides multiple installation methods based on the level automation and customization required. The fully **Automated** or Installer Provisioned Installation (IPI) method was used to deploy the OpenShift clusters in this solution as it is the quickest way to deploy a cluster. IPI installer deploys a Kubernetes cluster using an opinionated, prescriptive approach with minimal input from the user. To deploy the OpenShift cluster on vSphere, the installer will require API access to VMware vCenter with read/create privileges to deploy the master and worker nodes VMs. When the install is initiated, the installer will also prompt the user for some information such as the VLAN, vSphere virtual switch and datastore to use.

[Figure 17](#) shows the high-level OpenShift cluster design deployed by the IPI installer. The default IPI install deploys 3 x control/master and 3 x compute/worker nodes (VMs) – additional worker nodes can be added as needed.

Figure 17. Solution Design - Red Hat OpenShift Application Platform on FlashStack AI



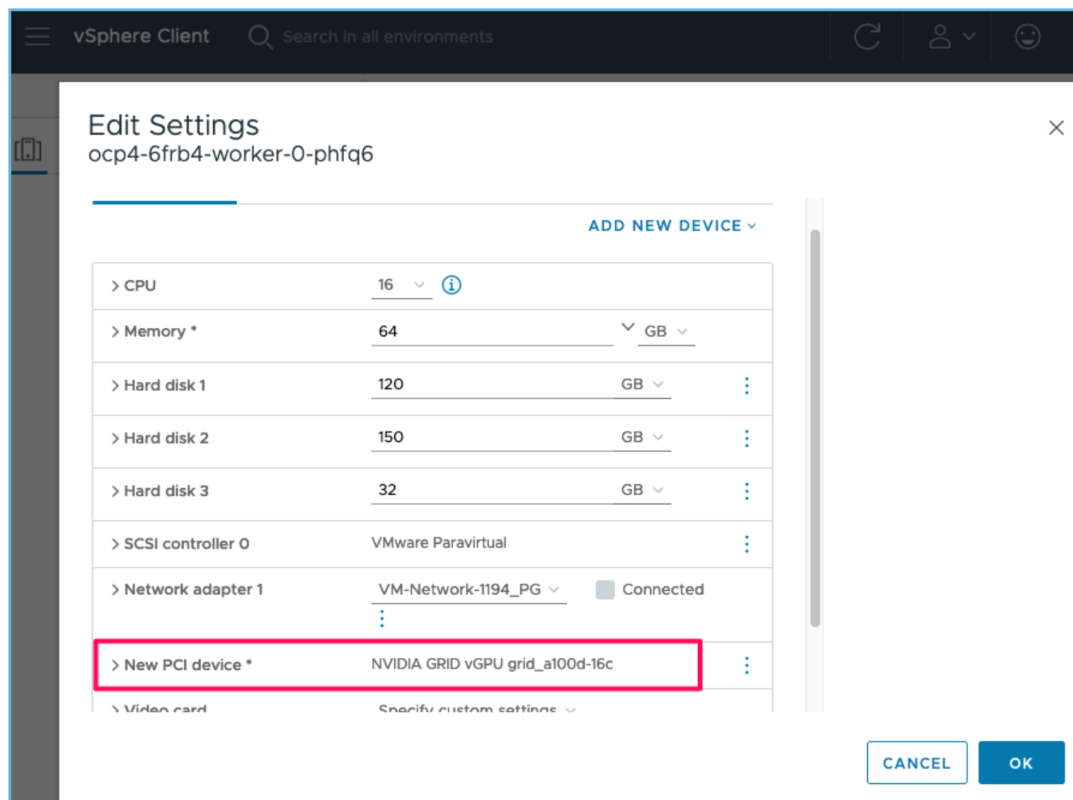
The OpenShift cluster is deployed on a VMware VM network that was provisioned during FlashStack VSI deployment. Post-install, the CPU, memory, and disk space allocated to the OpenShift worker nodes were

adjusted to meet the needs of the AI/ML workloads. [Table 8](#) lists the worker node virtual machine configuration used in this solution.

Table 8. Worker Node Virtual Machine Configuration

| Component | Configuration |
|-----------|--|
| CPU | 16 vCPUs |
| RAM | 64GB |
| Disk | 256GB (VMFS Datastore backed by Pure Storage FlashArray) |

To support AI/ML workloads on the worker nodes, vGPUs, previously enabled, are added as a PCIe device on each worker node. To add a PCIe device to a worker node VM, the VM must first be powered down.



With vGPUs deployed on worker nodes that require it, VMware’s VM-Host Affinity rules were provisioned to specify the following:

- VMs with vGPUs (some worker nodes) should only run on ESXi hosts with GPUs
- VMs not using vGPUs (master nodes, some worker nodes) should only run on ESXi hosts without GPUs

The VM-Host Affinity rules configured for the OpenShift cluster in this solution are shown in the following figures.

Figure 18. VM/Host Rules for Master and Worker Nodes in the vSphere cluster

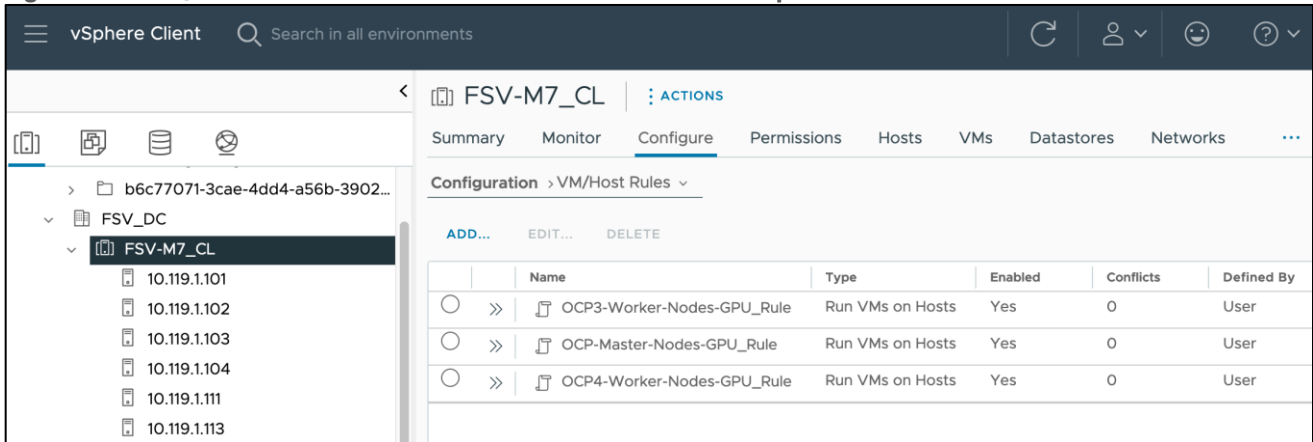


Figure 19. VM/Host Rules - Worker Nodes VMs with vGPUs on ESXi Host with GPU

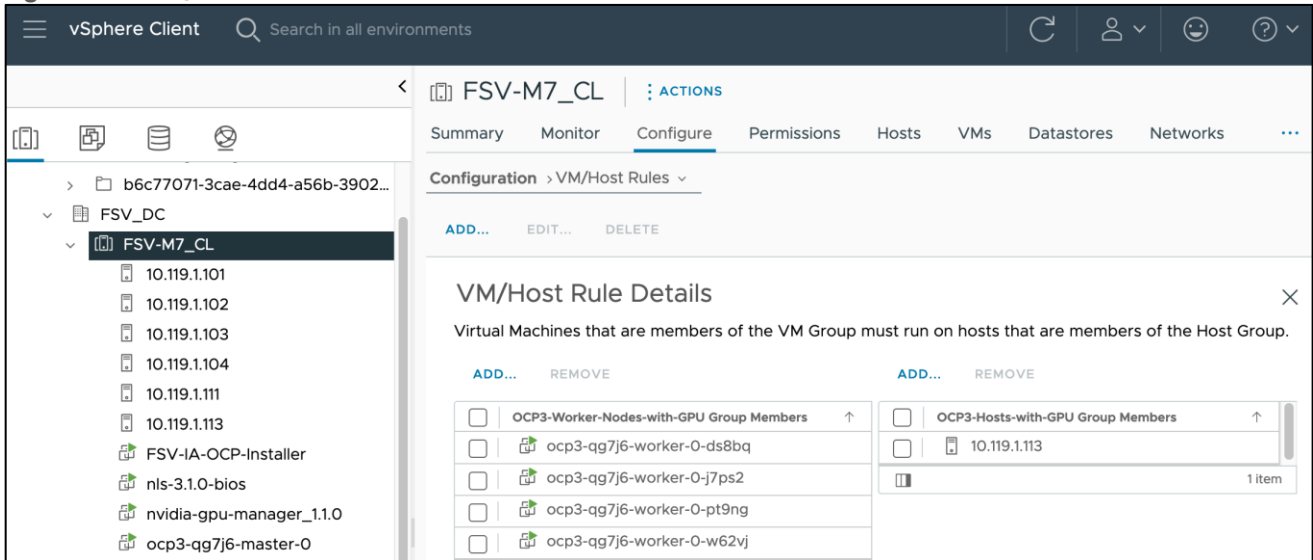
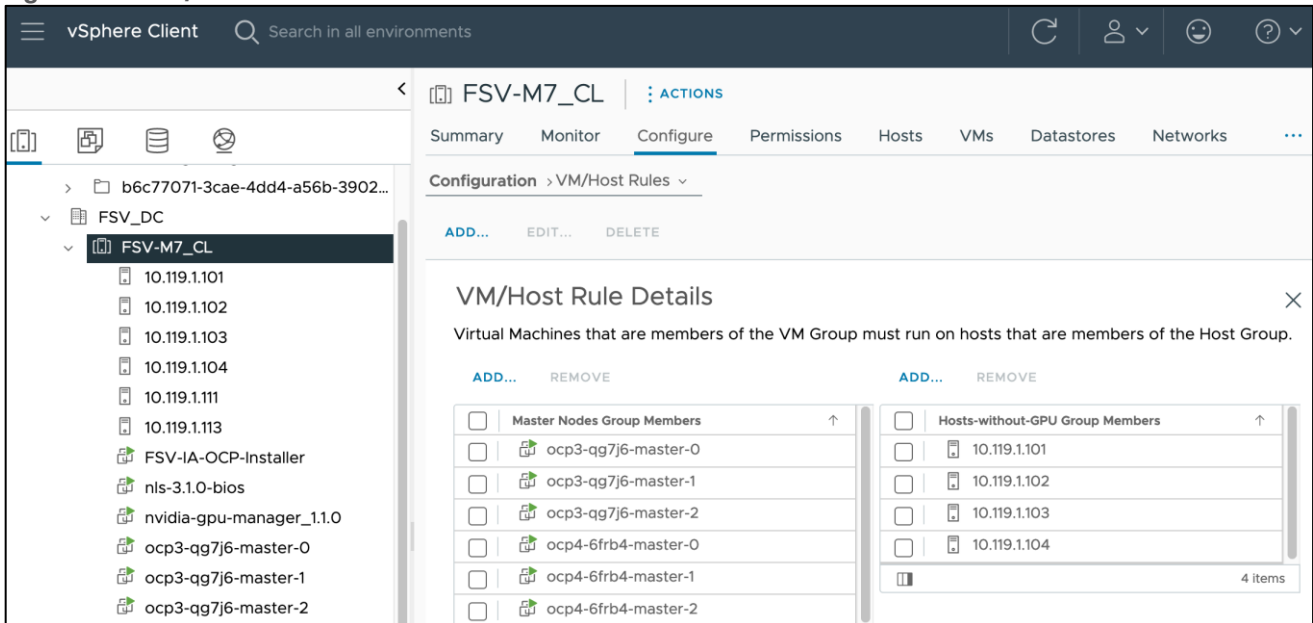


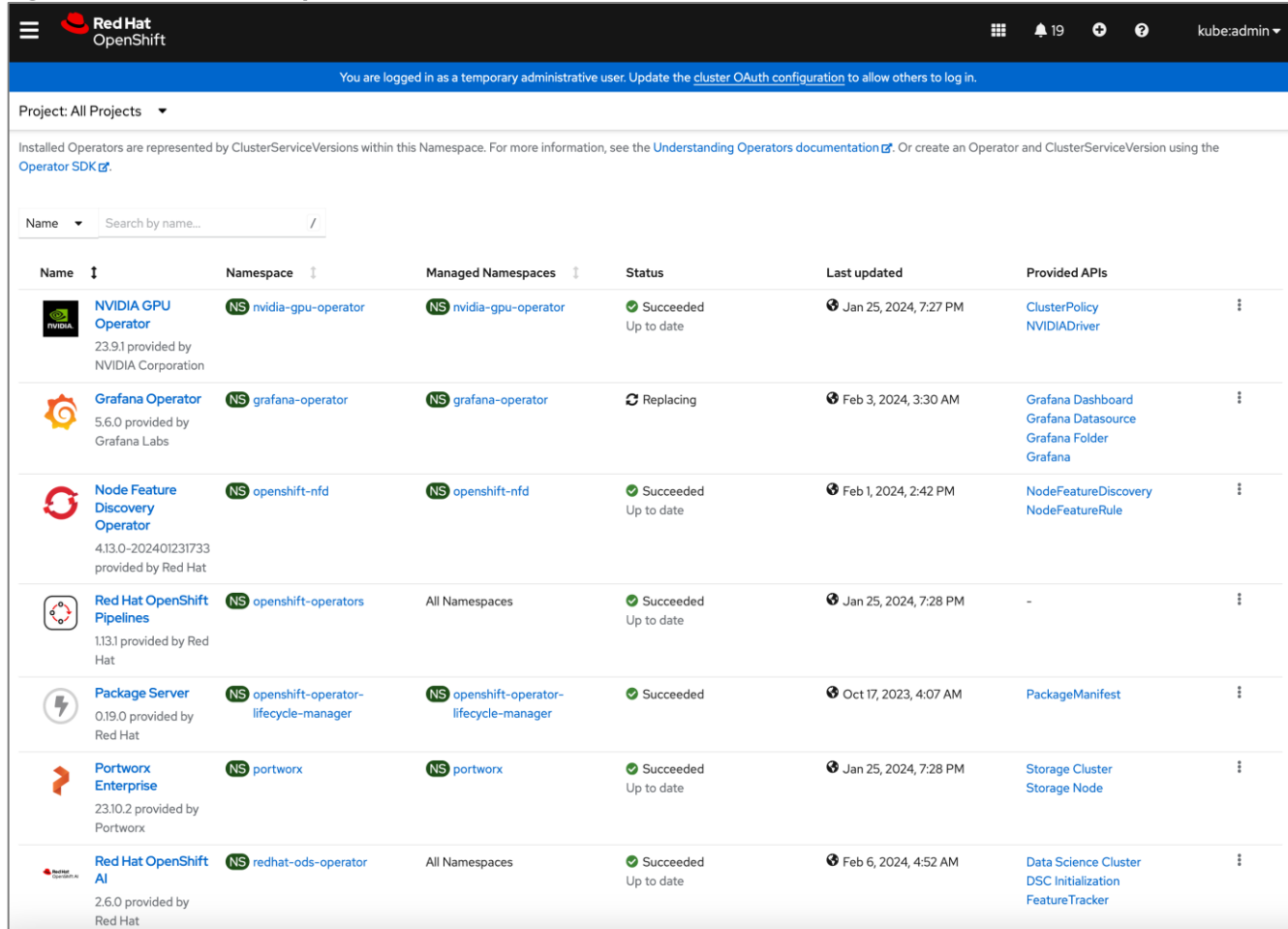
Figure 20. VM/Host Rules - Master Nodes VMs on ESXi Hosts without GPU










Kubernetes Operators

Operators are a powerful tool in Kubernetes. It was designed to extend the capabilities of a Kubernetes cluster without changing the core Kubernetes code. Once a cluster is deployed, Red Hat OpenShift uses Kubernetes Operators to deploy additional capabilities and services on the cluster. Red Hat OpenShift provides an embedded OperatorHub that can be accessed directly from the cluster console. The OperatorHub is a library of certified and community operators from partners and open-source projects that can be deployed on the cluster. These operators enable a range of services and capabilities with automated lifecycle management. The operators deployed for this solution are shown in [Figure 21](#).

Figure 21. Kubernetes Operators



| Name | Namespace | Managed Namespaces | Status | Last updated | Provided APIs |
|---|---|---|---------------------------|-------------------------|--|
|  NVIDIA GPU Operator 23.9.1 provided by NVIDIA Corporation | NS nvidia-gpu-operator | NS nvidia-gpu-operator | ✔ Succeeded Up to date | 🕒 Jan 25, 2024, 7:27 PM | ClusterPolicy NVIDIADriver |
|  Grafana Operator 5.6.0 provided by Grafana Labs | NS grafana-operator | NS grafana-operator | 🔄 Replacing | 🕒 Feb 3, 2024, 3:30 AM | Grafana Dashboard Grafana Datasource Grafana Folder Grafana |
|  Node Feature Discovery Operator 4.13.0-202401231733 provided by Red Hat | NS openshift-nfd | NS openshift-nfd | ✔ Succeeded Up to date | 🕒 Feb 1, 2024, 2:42 PM | NodeFeatureDiscovery NodeFeatureRule |
|  Red Hat OpenShift Pipelines 1.13.1 provided by Red Hat | NS openshift-operators | All Namespaces | ✔ Succeeded Up to date | 🕒 Jan 25, 2024, 7:28 PM | - |
|  Package Server 0.19.0 provided by Red Hat | NS openshift-operator-lifecycle-manager | NS openshift-operator-lifecycle-manager | ✔ Succeeded | 🕒 Oct 17, 2023, 4:07 AM | PackageManifest |
|  Portworx Enterprise 23.10.2 provided by Portworx | NS portworx | NS portworx | ✔ Succeeded Up to date | 🕒 Jan 25, 2024, 7:28 PM | Storage Cluster Storage Node |
|  Red Hat OpenShift AI 2.6.0 provided by Red Hat | NS redhat-ods-operator | All Namespaces | ✔ Succeeded Up to date | 🕒 Feb 6, 2024, 4:52 AM | Data Science Cluster DSC Initialization FeatureTracker |

NVIDIA GPU Operator

The NVIDIA GPU Operator uses the [operator framework](#) within Kubernetes to automate the management of all NVIDIA software components needed to provision and monitor GPUs. These components include:

- NVIDIA drivers (to enable CUDA)
- Kubernetes device plugin for GPUs
- NVIDIA Container Runtime
- Automatic node labeling
- NVIDIA DCGM exporter

The NVIDIA GPU operator also requires Red Hat's Node Feature Discovery Operator to detect the vGPUs assigned to a worker node. The GPU operator is responsible for deploying the vGPU Guest OS driver on the worker nodes.

Red Hat Node Feature Discovery Operator

The Node Feature Discovery Operator (NFD) is responsible for detecting hardware capabilities and labeling the nodes with the hardware-specific information so that OpenShift cluster can use them. In the case of vGPUs assigned to worker nodes, the NFD Operator detects that it is a PCIe GPU from NVIDIA and labels it accordingly. The label on NVIDIA GPU nodes will be:

```
feature.node.kubernetes.io/pci-10de.present=true
```

Portworx Enterprise Operator

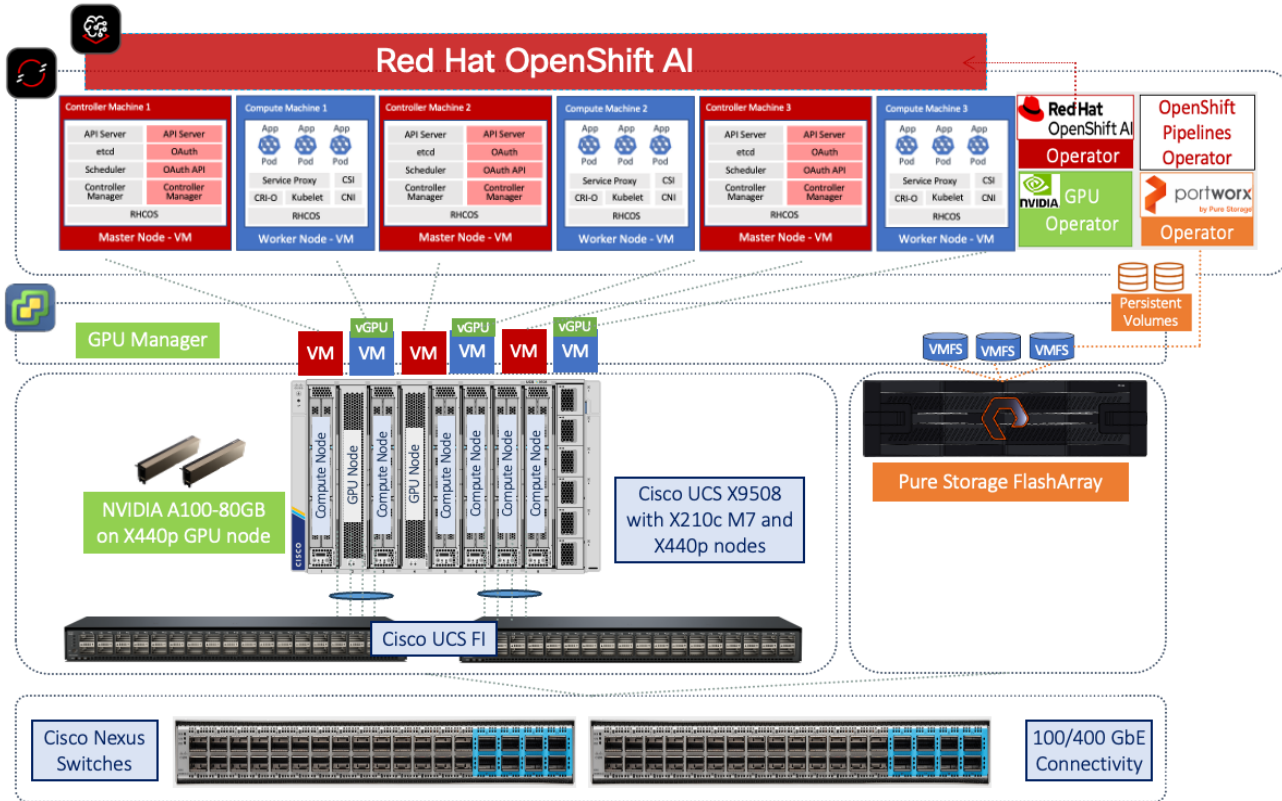
The Portworx Enterprise Operator also uses the operator framework within Kubernetes to automate and manage persistent storage for Kubernetes applications and workloads. The Portworx Enterprise operator will enable dynamic provisioning of container-granular volumes, automated capacity management, business continuity and disaster recovery, and Encryption and Role Based Access Control. This design will leverage Portworx to provide persistent storage for AI/ML workloads as needed, backed by VMFS datastores running on a Pure Storage FlashArray. In this design, the image registry for model delivery will use Portworx provisioned storage.

Red Hat OpenShift AI Operator

Red Hat OpenShift AI operator deploys OpenShift AI on the OpenShift cluster that enables a fully supported environment for MLOps. The OpenShift AI environment deployed by the operator provides a core environment with built-in tools, libraries, and frameworks that ML engineers and data scientists need to train and deploy models. The GPU resources deployed on the OpenShift cluster are automatically available from the OpenShift AI UI and can be used as needed during various stages of model delivery (for example, GPUs can be assigned to a Jupyter notebook for use in model experimentation). OpenShift AI includes project workspaces to enable multiple AI/ML efforts in parallel, Jupyter Notebooks with different built-in images to pick from (for example, PyTorch, TensorFlow, CUDA), Data Science Pipelines using OpenShift pipelines, model serving using ModelMesh (and Kserve) with Intel OpenVINO inferencing server. Customers can extend this environment by adding custom images, and other partner and open-source technologies. By using the operator framework, it is also simple to lifecycle OpenShift AI.

[Figure 22](#) shows the solution design with the Kubernetes operators deployed on the Red Hat OpenShift cluster. At this point, the worker nodes have functioning GPUs with Guest OS drivers deployed, persistent storage provided by Portworx, and Openshift AI deployed for MLOps.

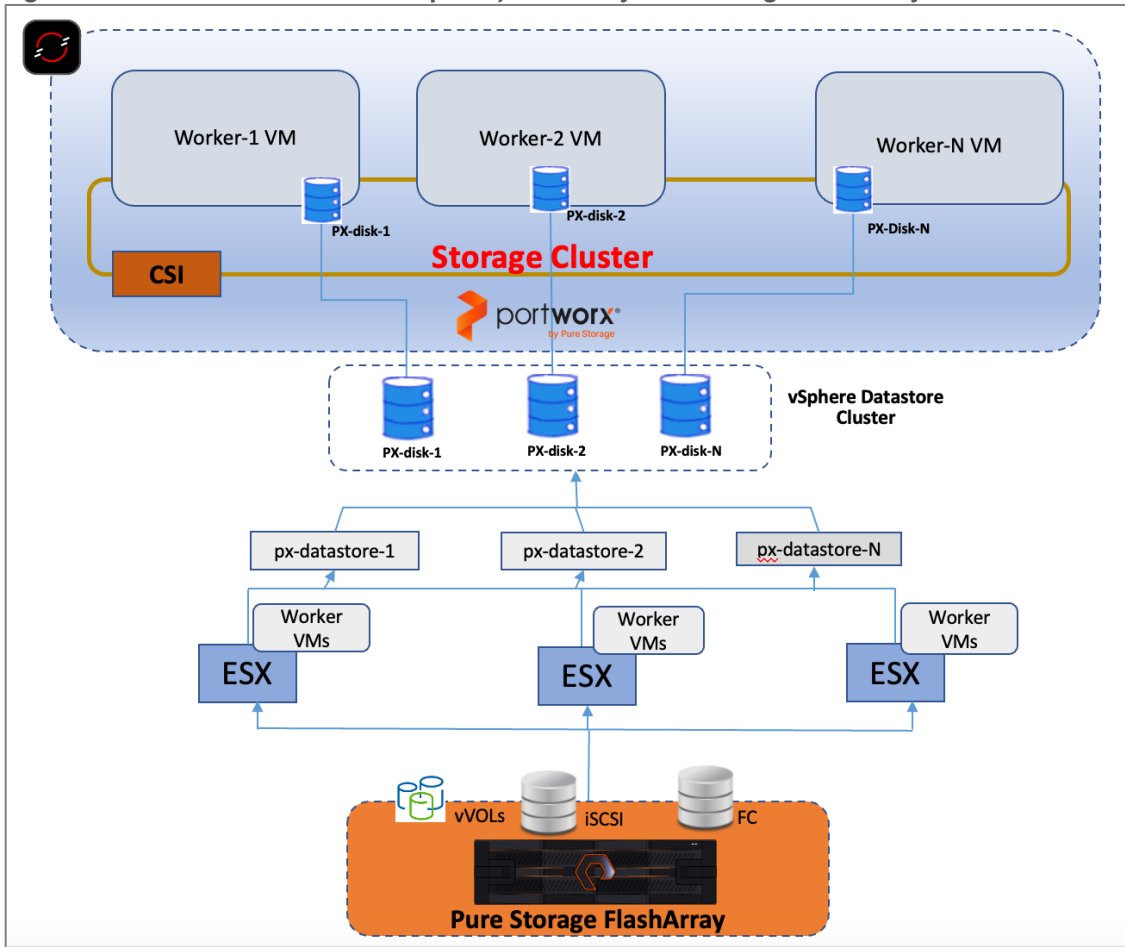
Figure 22. Solution Design - FlashStack AI with Kubernetes Operators (GPU, Portworx, OpenShift)



Portworx on VMware vSphere and Pure Storage FlashArray

In this solution, Portworx is deployed using VMware vSphere datastores backed by Pure Storage FlashArray//X. The FlashArray provides block storage (vVOLs, FC, iSCSI and NVMeoF) to ESXi hypervisors. VMware vSphere datastores are created on vCenter for Portworx and OpenShift cluster to use. You can also create vSphere datastore cluster. Portworx accesses vSphere datastore clusters via API calls to VMware vCenter – this information is provided in the StorageCluster spec that was used to provision the Portworx cluster. Portworx runs on each OpenShift worker Node, and on each node, it will create its disk on the configured datastore or datastore clusters. Portworx will then aggregate these disks to form a single storage cluster. At this point, applications and workloads can make persistent volume claims (PVCs) to create persistent volumes (PVs) as on this storage cluster, including taking snapshots. Portworx tracks and manages the disks that it creates. In a failure event, if a new VM spins up, then the new VM will be able to attach to the same disk that was previously created by the node on the failed VM. [Figure 23](#) shows a high-level view of the Portworx cluster and provisioning of disks on the backend vSphere cluster and storage array.

Figure 23. Portworx on VMware vSphere, backed by Pure Storage FlashArray



When deploying Portworx on VMware vSphere, the following prerequisites must be met:

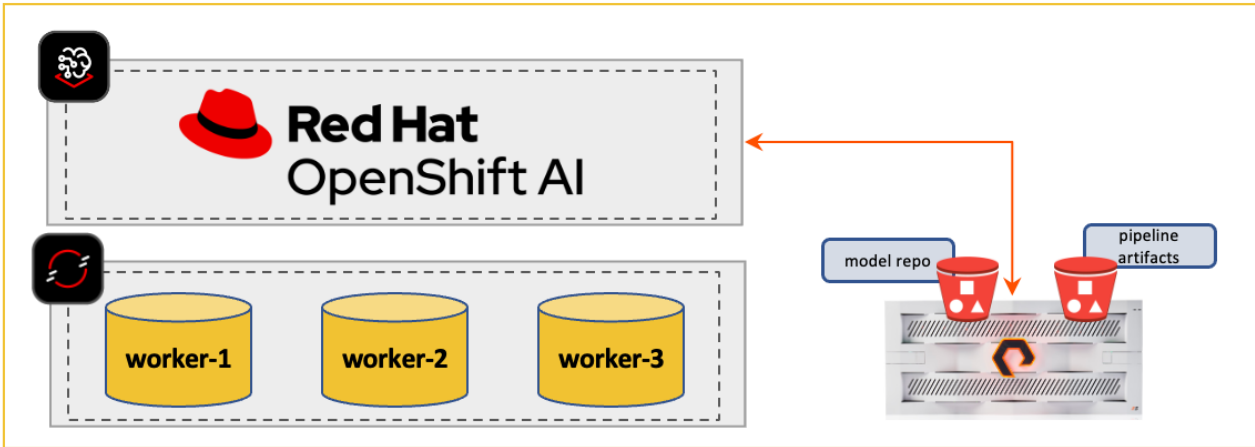
- VMware vSphere version 7.0 or newer.
- **oc** or **kubectl** configured on the machine having access to the cluster.
- Portworx does not support the movement of VMDK files from the datastores on which they were created.
- Cluster must be running OpenShift 4 or higher
- Virtual Machines used for OpenShift nodes for Portworx have Secure Boot disabled.

For more information, see: <https://docs.portworx.com/install-portworx/prerequisites/>

Pure Storage FlashBlade//S for S3 Compatible Object Store (Model Repo, Pipeline Artifacts)

OpenShift AI requires S3 compatible object stores for automation pipeline artifacts and to store models that will then be accessed by inferencing engines for production use. The object stores are presented directly to OpenShift AI for use in this solution. Other FlashBlade deployments options are provided in the next section.

Figure 24. Pure Storage FlashBlade//S for S3 Compatible Object Store



Alternatively, FlashBlade//S can also be used to provide NFS storage as a Direct Access filesystem and object storage to the Portworx cluster.

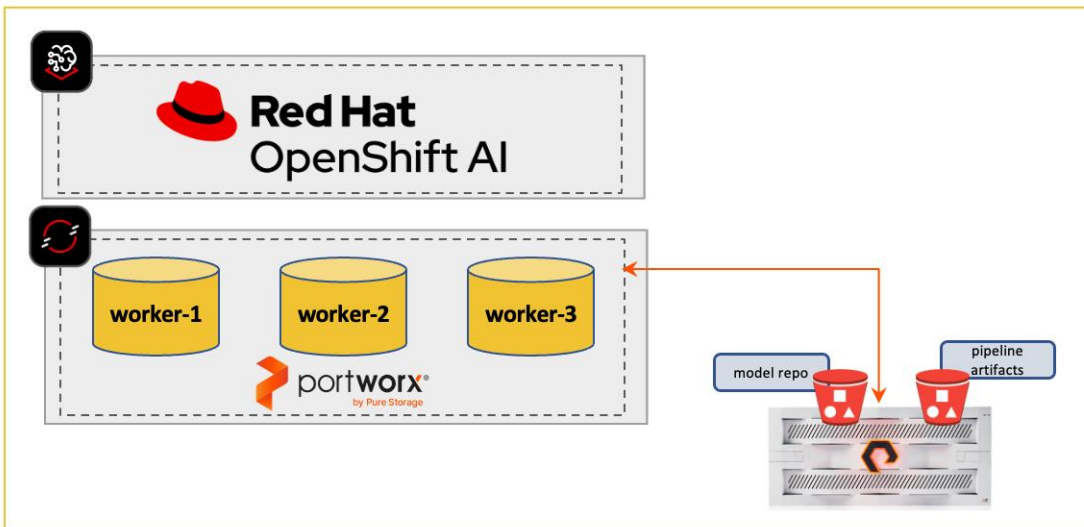
To create a S3 Bucket in FlashBlade, see:

https://support.purestorage.com/FlashBlade/Purity_FB/Data_Protocols/S3/Creating_a_S3_Bucket_in_FlashBlade

Object Store on Portworx backed by Pure Storage FlashBlade

Portworx Enterprise allows users to create and manage S3 compatible object storage on Pure Storage FlashBlade arrays for use by AI/ML workloads and other applications.

Figure 25. Object Stores on Portworx, backed Pure Storage FlashBlade



To set up Pure Storage FlashBlade with Portworx for Object storage:

- **PXBucketClass:** A PXBucketClass is used to create PXBuckets, similar to StorageClass which is used to create PersistentVolumes. Just as a StorageClass describes the types of persistent volumes that can be created, a PXBucketClass describes what types of object storage buckets can be created.

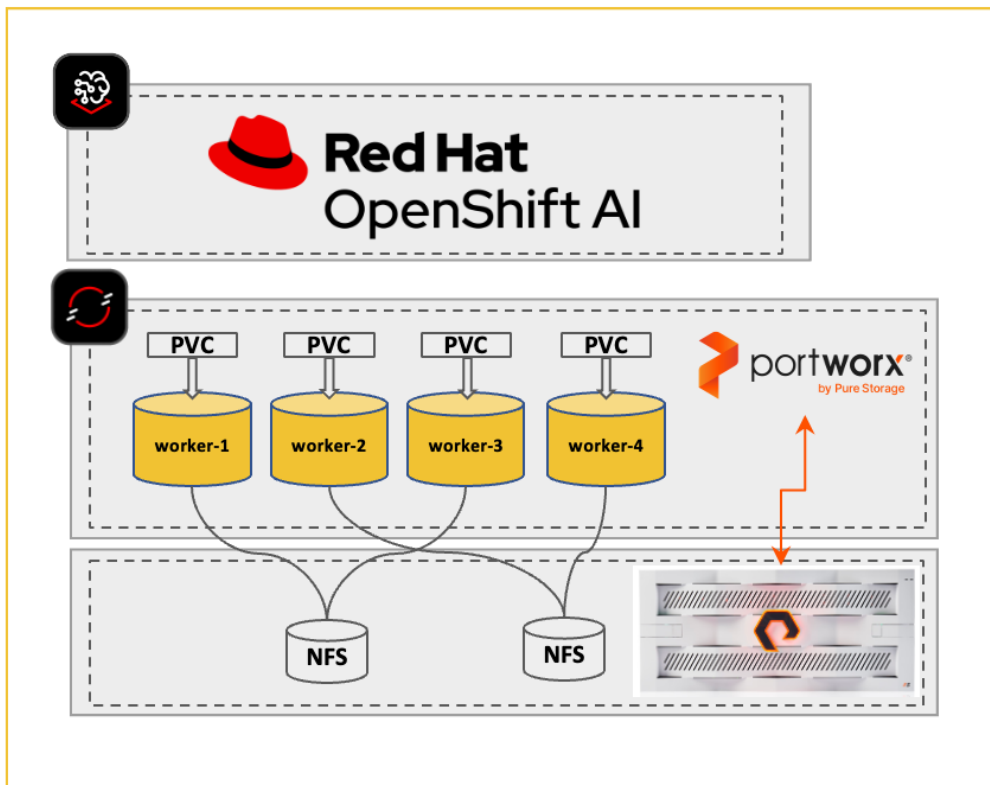
- **PXBucketClaim:** Just as a PXBucketClass acts like a StorageClass, a PXBucketClaim acts like a PersistentVolumeClaim (PVC). PXBucketClaims specify a PXBucketClass to dynamically request an object storage bucket.
- **PXBucketAccess:** PXBucketAccess is a way to put access controls on the object storage created through PXBucketClaims.

NFS - Direct Attached Storage in Portworx backed by FlashBlade

Portworx directly provisions FlashBlade NFS filesystems, maps them to a user PVC, and mounts them to pods. Once mounted, Portworx writes data directly onto FlashBlade. To set up Pure Storage FlashBlade with Portworx for NFS, the following must be setup:

- A single or multiple NFS volume with export rules for appropriate permissions.
- Configured NFS data services by choosing the interfaces in each FIOM and configured LAGs, IP address, subnet and VLAN details in the FlashBlade.
- Configure connection, protocol information and NFS versions v3 or v4.1.
- In Portworx, specify mount options through the CSI mountOptions flag in the StorageClass spec.
- Create PVC by referencing the StorageClass that was created and enter the StorageClass name in the spec.storageClassName field.

Figure 26. Pure Storage FlashBlade NFS as Direct Attached Storage in Portworx



For detailed steps to configure FlashBlade NFS as Direct Attached Storage on Portworx, see: <https://docs.portworx.com/portworx-enterprise/operations/operate-kubernetes/storage-operations/create-pvcs/pure-flashblade>

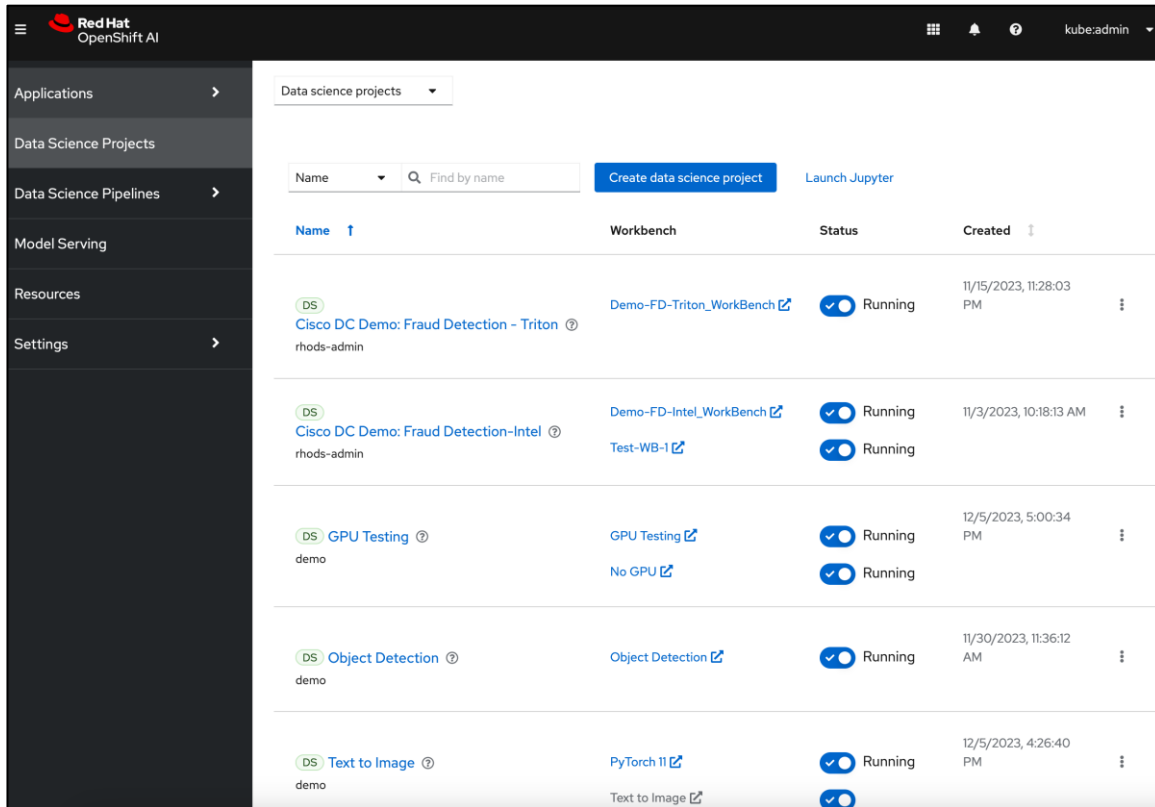
MLOps using Red Hat OpenShift AI

Red Hat OpenShift AI provides key capabilities for implementing MLOps to accelerate model delivery. You should have the following prerequisites in place before deploying Red Hat OpenShift AI:

- Provision or use existing identity provider from Red Hat OpenShift.
- Add user and administrator groups for accessing OpenShift AI.
- Deploy GPU resources for AI/ML efforts (for efforts that require GPUs).
- Deploy Persistent storage for AI/ML efforts.
- Deploy Red Hat OpenShift Pipelines (Operator) to enable automation pipelines for you AI/ML projects
- Deploy S3-compatible object store as model repository and to store pipeline artifacts.
- When using GPUs, if all nodes in the OpenShift cluster are not using GPUs, then taints and tolerations should be provisioned on the nodes to ensure that only workloads requiring GPUs are scheduled on the nodes with GPUs.

The features in Red Hat OpenShift AI for data scientists and ML engineers to work on a given AI/ML effort are listed below. Multiple efforts can run in parallel, including serving multiple production-ready models based on the scale and resources supported by the inferencing server deployed.

- Seamlessly leverage resources and capabilities from the underlying OpenShift cluster (for example, use OpenShift Identity provider to manage users). This allows data scientists and ML engineers to focus on their areas of expertise and less time on managing the environment.
- Support for multiple **Data Science Projects** to enable parallel AI/ML efforts.



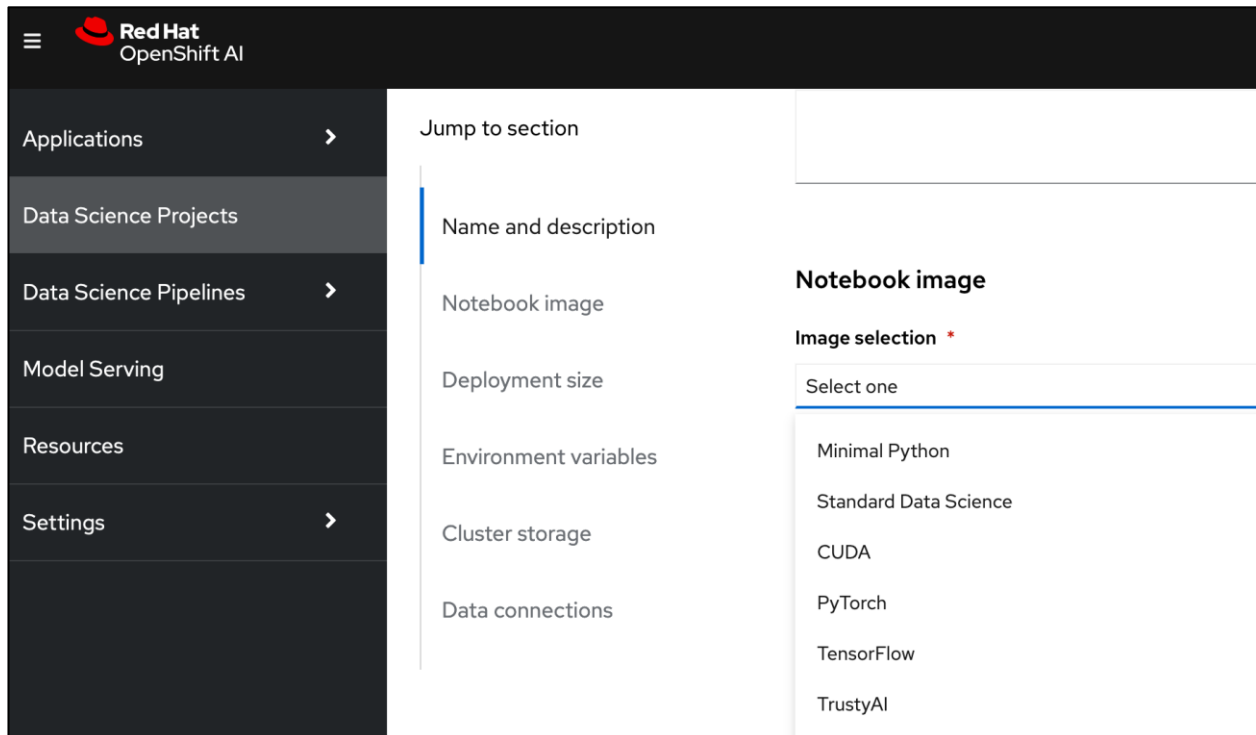
The screenshot displays the Red Hat OpenShift AI console interface. On the left is a navigation sidebar with options: Applications, Data Science Projects, Data Science Pipelines, Model Serving, Resources, and Settings. The main content area is titled 'Data science projects' and includes a search bar, a 'Create data science project' button, and a 'Launch Jupyter' button. Below this is a table listing several data science projects, each with its name, workbenches, status, and creation time.

| Name | Workbench | Status | Created |
|---|--------------------------|---------|-------------------------|
| DS Cisco DC Demo: Fraud Detection - Triton rhods-admin | Demo-FD-Triton_WorkBench | Running | 11/15/2023, 11:28:03 PM |
| DS Cisco DC Demo: Fraud Detection-Intel rhods-admin | Demo-FD-Intel_WorkBench | Running | 11/3/2023, 10:18:13 AM |
| | Test-WB-1 | Running | |
| DS GPU Testing demo | GPU Testing | Running | 12/5/2023, 5:00:34 PM |
| | No GPU | Running | |
| DS Object Detection demo | Object Detection | Running | 11/30/2023, 11:36:12 AM |
| DS Text to Image demo | PyTorch 11 | Running | 12/5/2023, 4:26:40 PM |
| | Text to Image | Running | |

- Support for multiple **Workbenches** within a given data science project to support parallel work efforts within the same projects. The workbenches launch Jupyter notebook environments with pre-built or

custom images with necessary libraries and frameworks. The pre-built image options available in this release are:

- Minimal Python
- Standard Data Science
- CUDA
- PyTorch
- TensorFlow
- TrustyAI

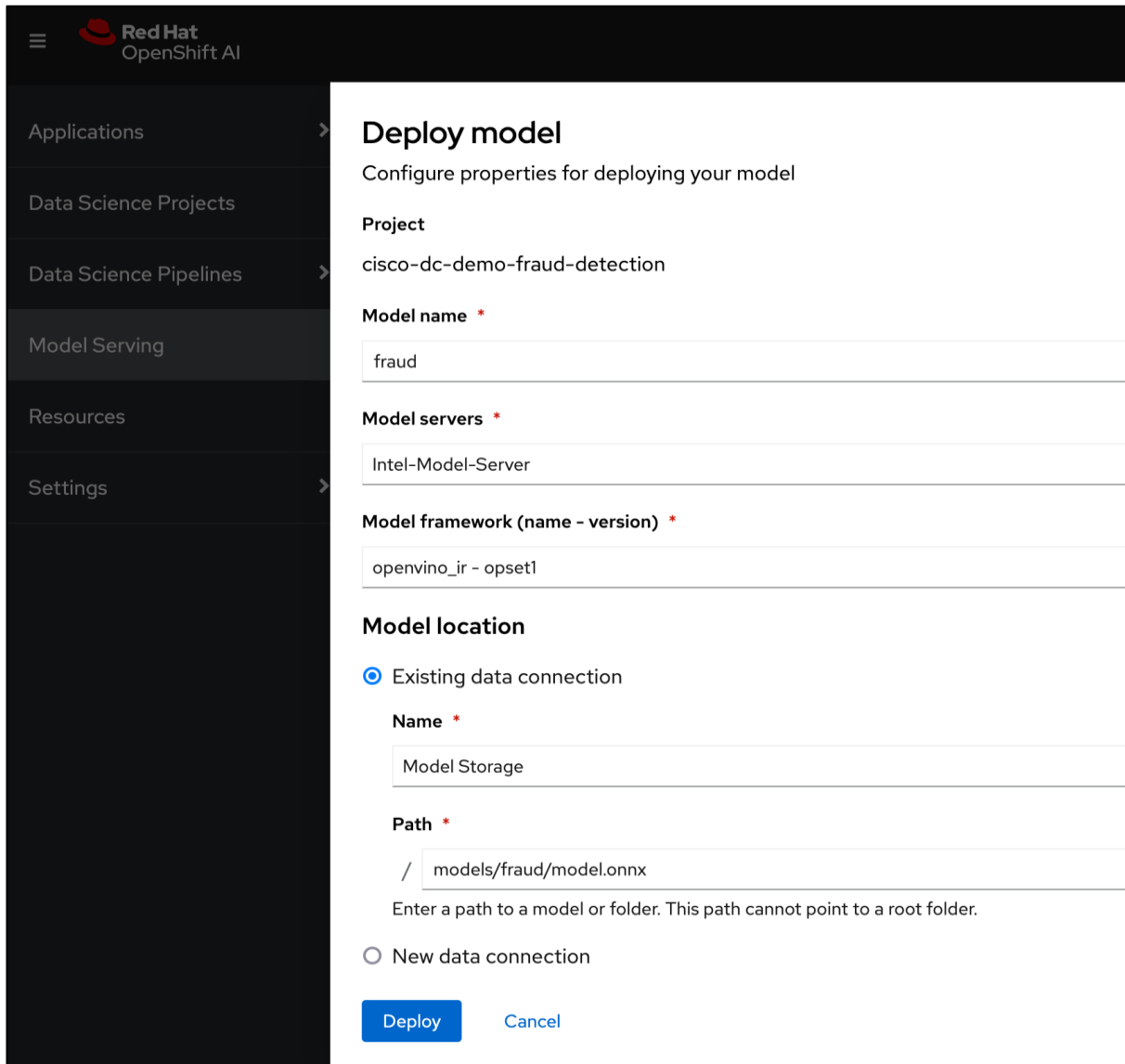


- Other notebook options you can select from include:
 - Container size (Small, Medium, Large, X Large) based on the amount of memory and CPU resources required.
 - Number of GPU accelerators (optional)
 - Persistent Storage - new or existing (provided by Portworx in this solution)
 - Data Connection to access S3-compatible storage on-premise or in the cloud (provided by Pure Storage FlashBlade in this solution)

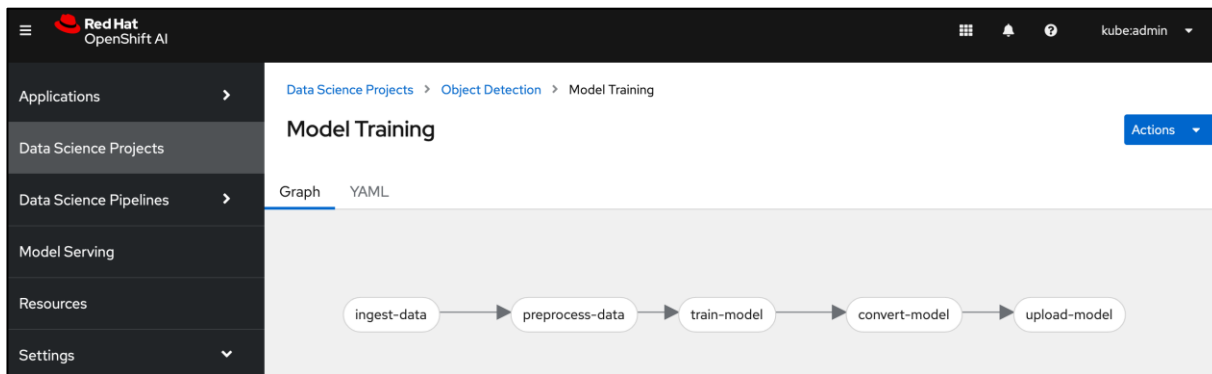
If GPU acceleration is selected, OpenShift AI will detect and make the GPU available for use. The pre-built images that support GPU acceleration will also be updated to indicate that it does as shown below. Otherwise, CPU resources will be used. Within a given data science project, the parallel efforts on different workbenches can individually select whether to use GPU or CPU resources.

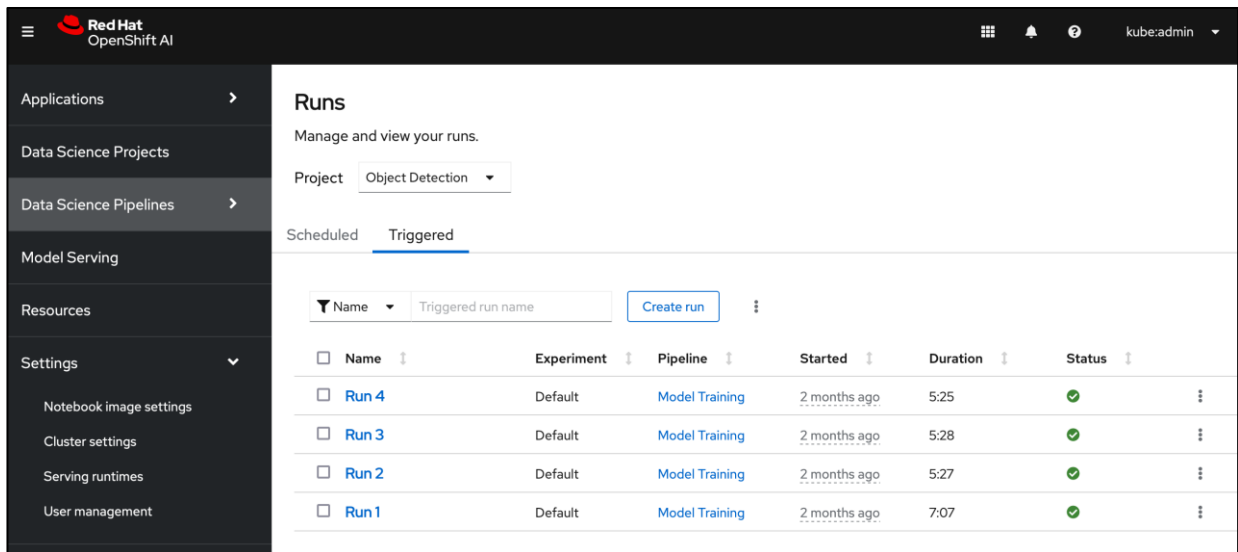
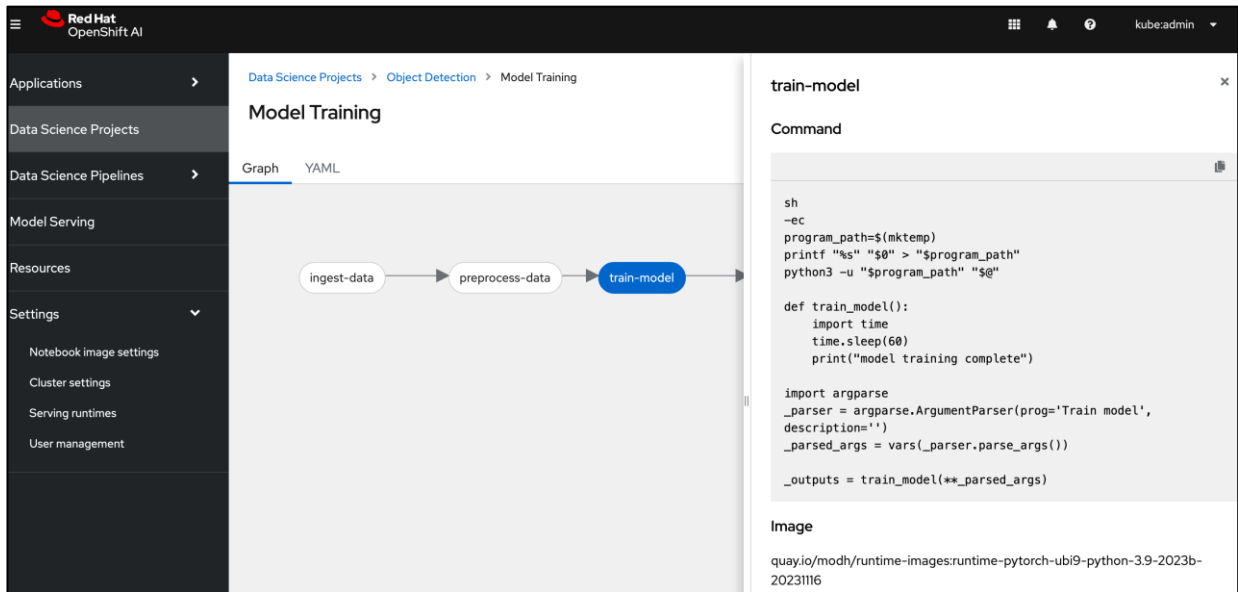
| Jump to section | Notebook image |
|-----------------------|---|
| Name and description | <p>Image selection *</p> <p>PyTorch</p> |
| Notebook image | Minimal Python |
| Deployment size | Standard Data Science |
| Environment variables | CUDA Compatible with accelerator |
| Cluster storage | PyTorch Compatible with accelerator |
| Data connections | TensorFlow Compatible with accelerator |
| | TrustyAI |

- Support for Model Serving using pre-integrated Intel OpenVINO inferencing server or use a custom server such as NVIDIA Triton. For model serving, you can specify the model repository where the model is stored, the format or framework the published model uses (for example, onnx, tensorflow, openvino_ir) as well as the number of GPU accelerators to use.



- Simple drag and drop GUI based Pipeline Automation with options to schedule execution runs.



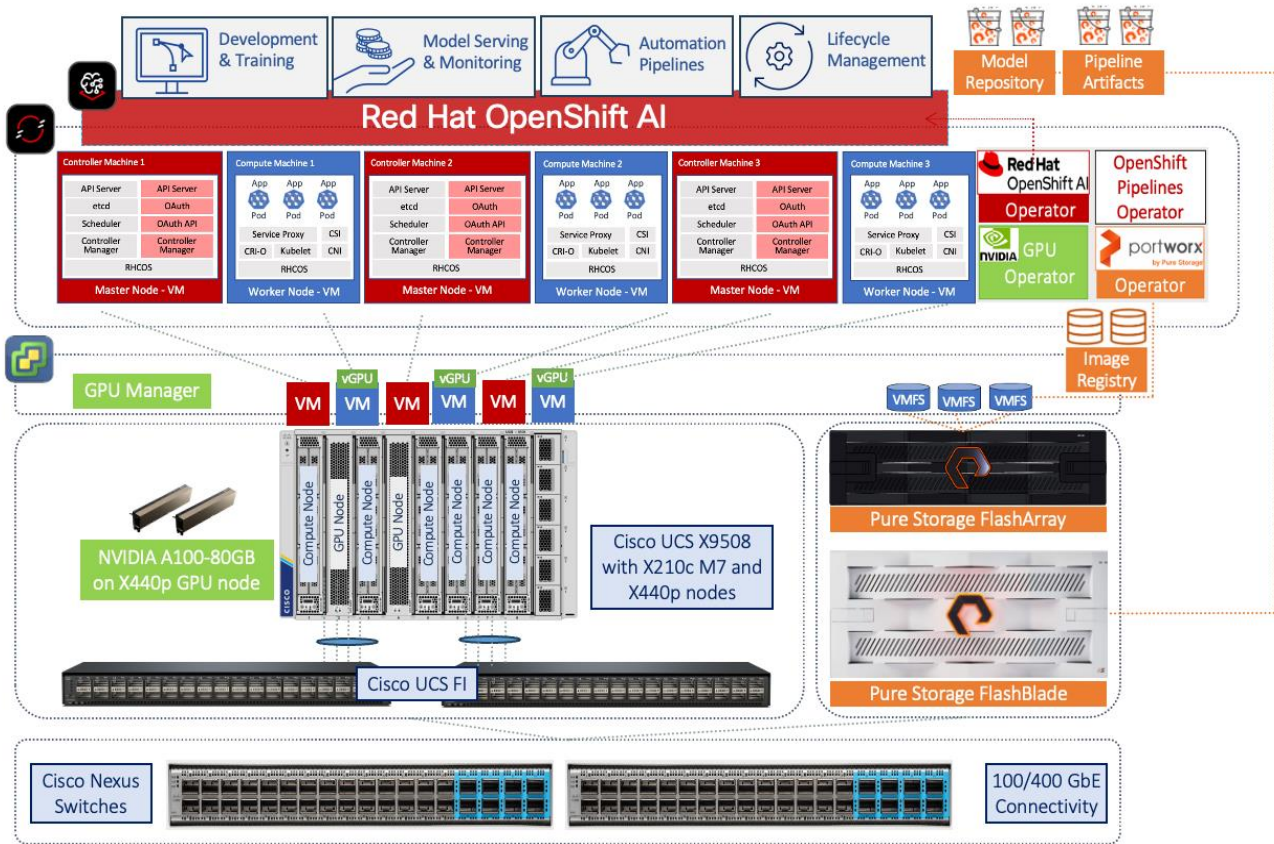


Red Hat OpenShift AI provides flexibility and ease to enable MLOps for Enterprise AI/ML efforts. The scalability of the solution will depend on the infrastructure resources that are available for use. A project running on the platform can seamlessly use the underlying K8s resources (GPU, storage) seamlessly since OpenShift AI runs on Red Hat OpenShift. This provides significant operational benefits for the Enterprise as IT teams managing OpenShift can also manage the ML infrastructure, allowing ML teams to focus on model development and delivery work.

End-to-End Design

The end-to-end design for MLOps for FlashStack AI using Red Hat OpenShift AI solution is shown in [Figure 27](#).

Figure 27. Solution Design – MLOps for FlashStack AI using Red Hat OpenShift AI



Solution Deployment

This chapter contains the following:

- [Deployment Overview](#)
- [Visibility and Monitoring - GPU](#)

This chapter provides a summary of the implementation steps for deploying the solution. The high level steps will include a link to detailed, step-by-step guidance for implementing that particular step in the overall process.

Deployment Overview

The AI/ML infrastructure leverages the latest FlashStack VSI CVD for Cisco UCS M7 servers and vSphere 8 as the foundational design for containerized AI/ML workloads running on Red Hat OpenShift and MLOps for model deployment and maintenance using Red Hat OpenShift AI.

Check Cisco UCS Hardware Compatibility List for NVIDIA GPU support on Cisco UCS for the VMware vSphere version and upgrade UCS server firmware as needed.

With the FlashStack VSI infrastructure in place, an overview of the remaining deployment steps to bring up the AI/ML infrastructure for model serving in production with MLOps are summarized in [Table 9](#).

The detailed procedures for the steps outlined in this table are available in GitHub: <https://github.com/ucs-compute-solutions/FlashStack-OpenShift-AI>

Table 9. Deployment Overview

| Steps | Deployment Action |
|--------|--|
| 01_CVD | <p>UCS Server Prerequisites: Verify that Cisco UCS server settings for NVIDIA GPUs are set correctly. For optimal performance, settings should be enabled or set as outlined below:</p> <ul style="list-style-type: none">• BIOS Policy: Intel VT-d /IOMMU• BIOS Policy: Single Root I/O Virtualization (SR-IOV)• Hyperthreading• Power Setting or System Profile - High Performance• CPU Performance (if applicable) - Enterprise or High Throughput• Memory Mapped I/O above 4-GB - Enabled (if applicable) |
| 02_CVD | <p>Red Hat OpenShift Prerequisites: Setup and/or verify that the following prerequisites for Red Hat OpenShift are in place.</p> <ul style="list-style-type: none">• Deploy an installer workstation for deploying a Red Hat OpenShift cluster.• Valid Red Hat account to access Red Hat Hybrid Cloud Console (HCC) for deploying OpenShift.• Identify FlashStack vSphere infrastructure (hosts, cluster, storage) for hosting the OpenShift cluster.• Identify a VLAN, IP subnet and DNS domain for the Red Hat OpenShift cluster to use.<ul style="list-style-type: none">- Add DNS records for API VIP and Ingress Virtual IP (VIP)- Add DHCP pool for OpenShift cluster nodes to use- Add NTP server for OpenShift cluster to use via DHCP- Add Gateway IP for OpenShift subnet to use via DHCP• Generate public SSH keys on the installer to enable SSH access to OpenShift cluster post-install.• Download VMware vCenter root CA certificates to installer's system trust for secure access. |

| Steps | Deployment Action |
|--------|--|
| | <ul style="list-style-type: none"> Download installation files, tools, and pull-secret from Red Hat HCC for VMware vSphere. |
| 03_CVD | <p>Deploy Red Hat OpenShift: Install OpenShift using the Automated or Installer Provisioned Infrastructure (IPI) method.</p> |
| 04_CVD | <p>Red Hat OpenShift – Post-Deployment Verification:</p> <ul style="list-style-type: none"> Verify access to OpenShift cluster by navigation to cluster console URL Setup/Verify NTP setup on all OpenShift cluster virtual machines (master and worker nodes) Verify cluster is registered with console.redhat.com Provision affinity rules for master and worker nodes on VMware vCenter From Red Hat OpenShift cluster console, provision machineset to modify CPU, memory as needed. NVIDIA User Guide provides the following guidance for worker nodes when using vGPUs. <ul style="list-style-type: none"> vCPUs=16 RAM=64GB Storage: 500GB thin provisioned NIC: VMXNet3 |
| 05_CVD | <p>NVIDIA AI Enterprise Software (NVAIE) Host Driver Prerequisites:</p> <ul style="list-style-type: none"> Account on NVIDIA's Licensing Portal (NLP). At least one NVIDIA data center GPU installed on a NVIDIA AI Enterprise (NVAIE) Compatible NVIDIA-Certified System. Acquire required Licenses for the GPU model being used (for example, NVAIE Licenses for A100-80G). Deploy a NVIDIA Licensing server, either a Delegated License System (DLS) instance on-prem or use the Cloud License System (CLS) as outlined in this document. Allocate a license pool for the DLS to use. Deploy NVIDIA GPU Manager and vSphere plugin for VMware vCenter as outlined here. <ul style="list-style-type: none"> Register NVIDIA GPU Manager for VMware vCenter Administrator User Register NVIDIA GPU Manager with VMware vCenter so that you can download and manage NVIDIA GPU drivers from vSphere client. The registration process will install the vSphere plugin for the GPU Manager in VMware vCenter. Register NVIDIA GPU manager with NVIDIA Licensing Portal (NLP). <ul style="list-style-type: none"> From NLP, create a Software Downloads API Key – copy it Paste the API key in VMware vCenter > GPU Manager Verify that you can see the Host GPU drivers in GPU manager that are available from NLP. |
| 06_CVD | <p>Deploy NVAIE Host Driver for VMware vSphere. The steps below are for deploying GPUs in vGPU mode. vGPUs also require a guest OS driver (see later step).</p> <ul style="list-style-type: none"> Once the host repo is visible in GPU manager, select and download the NVIDIA GPU Host driver to be deployed. See Cisco UCS HCL and VMware Compatibility List to select a supported driver. Navigate to VMware vSphere Life Cycle Manager (LCM) to see the downloaded driver. With the driver available in GPU Manager repo, navigate to the VMware vSphere cluster with the GPU nodes and use VMware LCM to update the cluster image with the host GPU driver to be loaded (and any others as needed). You can also use software vib install directly from ESXi host to install the NVAIE Host drivers as outlined here. Remediate the ESXi hosts individually to update each node. |
| 07_CVD | <p>NVAIE Host Driver Post-Deployment Verification:</p> <ul style="list-style-type: none"> Verify that you can see the GPU by SSH into ESXi host as root and executing the following |

| Steps | Deployment Action |
|---------|--|
| | <p>commands</p> <ul style="list-style-type: none"> - lspci grep NVIDIA - nvidia-smi -q <ul style="list-style-type: none"> • From VMware vCenter, change the default Graphics Type from Virtual Shared Graphics Acceleration (vSGA) to vGPU (Shared Direct). Host driver supports both in the same software VIB. • SSH into ESXi and confirm GPU Virtualization Mode is Host vGPU and Host vGPU Mode is SR-IOV by executing the command: nvidia-smi -q • Configure Host/VM Affinity rules so that the Red Hat OpenShift worker nodes are on hosts with GPUs. |
| 08_CVD | <p>Add vGPUs to Red Hat OpenShift worker node VMs. Requires VM to be shutdown.</p> <ul style="list-style-type: none"> • From VMware vCenter, select a worker node VM to add vGPU to and gracefully shut it down. • Edit the settings on the OpenShift worker node and configure the following: <ul style="list-style-type: none"> - Assign a new PCIe device – list of vGPU profiles up to the maximum frame buffer capacity of the GPU will be listed. Multiple vGPUs can be assigned from the same or different GPU. The vGPU profiles for any given GPU must all the same such as the frame buffer should all be the same for vGPUs for a given GPU. - Select the Boot Option for Firmware as EFI. - Adjust the Memory Mapped I/O (MMIO) settings for the VM under Advanced Parameters. The size will depend on the NVIDIA GPU model (see NVIDIA documentation for your GPU model: <ul style="list-style-type: none"> o pciPassthru.use64bitMMIO = TRUE o pciPassthru.64bitMMIOSizeGB = 512 • Power the Virtual Machine backup. • Repeat for remaining worker nodes. • Wait for all worker nodes to be in Ready Status from the Red Hat OpenShift console. |
| 09_CVD | <p>Deploy NVIDIA GPU Operator on Red Hat OpenShift.</p> <ul style="list-style-type: none"> • From the Red Hat OpenShift Console, search and deploy Red Hat’s Node Feature Discovery Operator (NFD). • Verify that the worker nodes have a label for the NVIDIA vGPU if it was added to the worker node. • From the Red Hat OpenShift cluster console, search and deploy the NVIDIA GPU Operator. <ul style="list-style-type: none"> - Deploy Cluster Policy instance and ensure that it shows a Status of State:Ready - Use the following command to verify GPU details (for example, vGPU driver) <ul style="list-style-type: none"> o oc exec -it <nvidia-driver-daemonset pod name> -- nvidia-smi - Use the following command to verify vGPU licensing. If it is not licensed, you will see performance issues. <ul style="list-style-type: none"> o oc exec -it <nvidia-driver-daemonset pod name> -- nvidia-smi -q - You can also confirm the licenses have been allocated from DLS instance deployed on-prem for licensing (if you’re using CLS – go to NLP to see the leases) • Enable DCGM GPU Monitoring Dashboard in Red Hat OpenShift for vGPUs • Enable GPU monitoring in VMware vCenter for the physical GPU. |
| 010_CVD | <p>Deploy Persistent Storage on Red Hat OpenShift using Portworx from Pure Storage</p> |

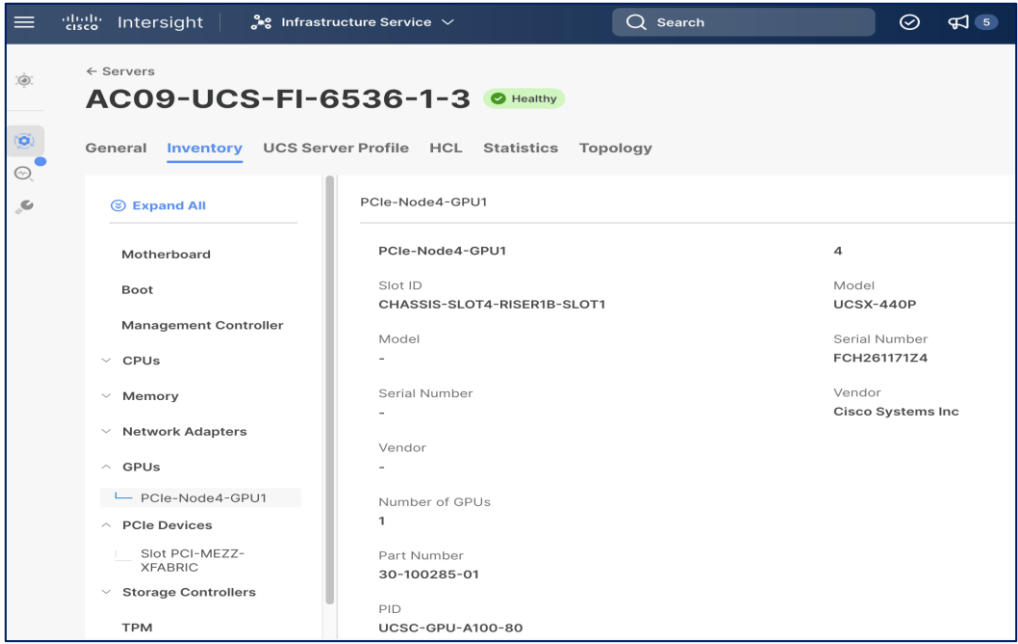
| Steps | Deployment Action |
|---------|---|
| | <p>The persistent storage will be used Red Hat OpenShift AI as an Image Repository for AI/ML model development, experimentation, and so on.</p> <ul style="list-style-type: none"> • Deploy Portworx Enterprise Operator from Operator Hub • Generate a StorageCluster spec from Portworx Central • Create Secret object to enable secure access to VMware vCenter • Deploy Storage Class • Deploy Storage Spec • Verify Portworx Cluster status • Create a Persistent Volume Claim • Make Portworx the default storage class |
| 011_CVD | <p>Deploy Red Hat OpenShift AI for MLOps</p> <p>This involves the following high-level tasks:</p> <ul style="list-style-type: none"> • Confirm that your OpenShift cluster meets all prerequisites (previous steps). • Configure an identity provider – you can use what Red Hat OpenShift uses • Install the Red Hat OpenShift AI Operator. • Configure user and administrator groups to provide user access to OpenShift AI – in Red Hat OpenShift. • Access the OpenShift AI dashboard directly or from Applications menu in Red Hat OpenShift. • Enable GPUs (optional) in OpenShift AI to ensure that your data scientists can use compute-heavy workloads in their models. |
| 012_CVD | <p>Deploy S3-compatible object stores on Pure Storage FlashBlade. These will be used by OpenShift AI as storage for Pipeline Artifacts and Model Repository.</p> |
| 013_CVD | <p>Verify GPU operation</p> <ul style="list-style-type: none"> • GPU Functional Validation – Sample CUDA Application • GPU Burn Test: https://github.com/wilicc/gpu-burn • Sample PyTorch script executed from Red Hat OpenShift AI |

Visibility and Monitoring - GPU

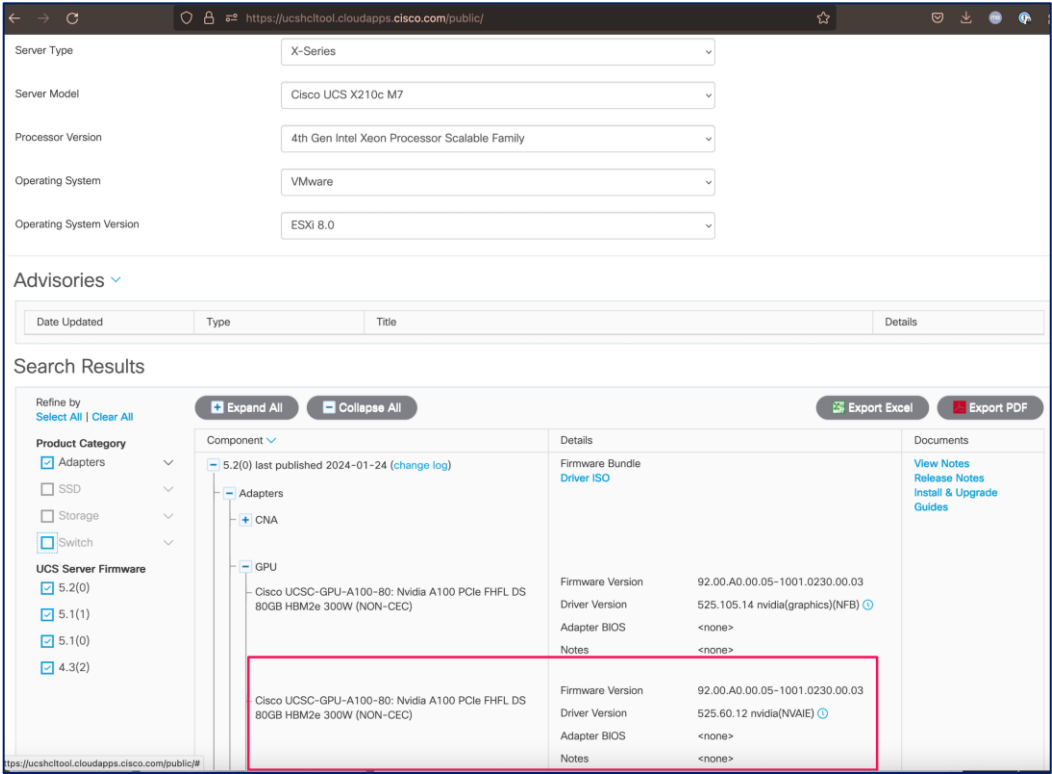
The GPUs deployed on Cisco UCS systems in the solution can be observed and monitored using the tools outlined in this section.

Visibility

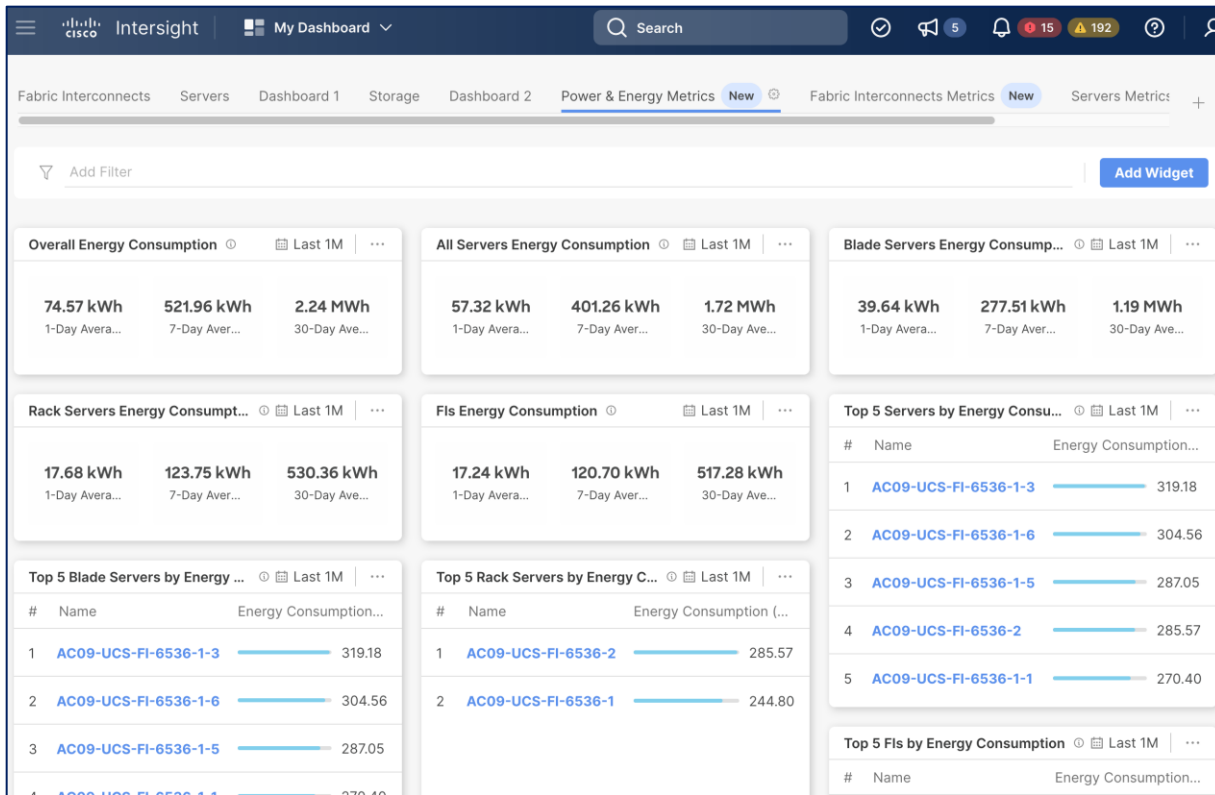
You can view the GPUs deployed on a PCIe (X440p) node associated with a server as shown below:



You can verify the correct drivers supported on the Cisco UCS server by selecting the HCL tab (see above) or by navigating to [UCS Hardware Compatibility List](#) (HCL) as shown below. Note that the per-server HCL check will show the supported driver based on the HCL shown below, but it **does** not validate that this driver is running on the server in question but will pass the HCL validation check anyway.



In addition to centralized provisioning and orchestration that Cisco Intersight provides, it also provides visibility across all sites and locations that an enterprise has. Enterprises can use either built-in or custom dashboards. For example, power and energy consumption is a critical consideration in AI/ML deployments and a dashboard such as the one shown below can help Enterprises understand their consumption pattern more efficiently.



Monitoring

To monitor the GPU utilization, memory, power and metrics, the solution uses the following tools to get a consolidated view. Alternatively, a Grafana dashboard can be set up to enable a consolidated view – this is outside the scope of this solution.

- Red Hat OpenShift observability dashboard available from the OpenShift cluster console
- **nvdi**a-**smi** CLI tool that NIVIDA provides for Red Hat OpenShift
- **nvdi**a-**smi** CLI tool that NIVIDA provides for VMware vSphere hosts
- VMware vCenter (requires the management **vib** from the host driver package to be installed)

vGPU Monitoring from Red Hat OpenShift Dashboard

The OpenShift dashboard uses Prometheus metrics NVIDIA GPU Operator exposes DCGM metrics to Prometheus that the dashboard uses to display GPU metrics available to OpenShift. NVIDIA GPU Operator, when deployed will expose DCGM metrics to the OpenShift that can be viewed from the integrated dashboard.

To view the metrics exposed by DCGM exporter in OpenShift, see the following file available [here](#). When creating custom dashboards using Grafana, the exact metric and query to use can be found here. Also, a JSON file with the metrics is available for [Grafana](#) from the same repo.

The OpenShift dashboard currently displays the following (default) metrics for a vGPU ([Table 10](#)).

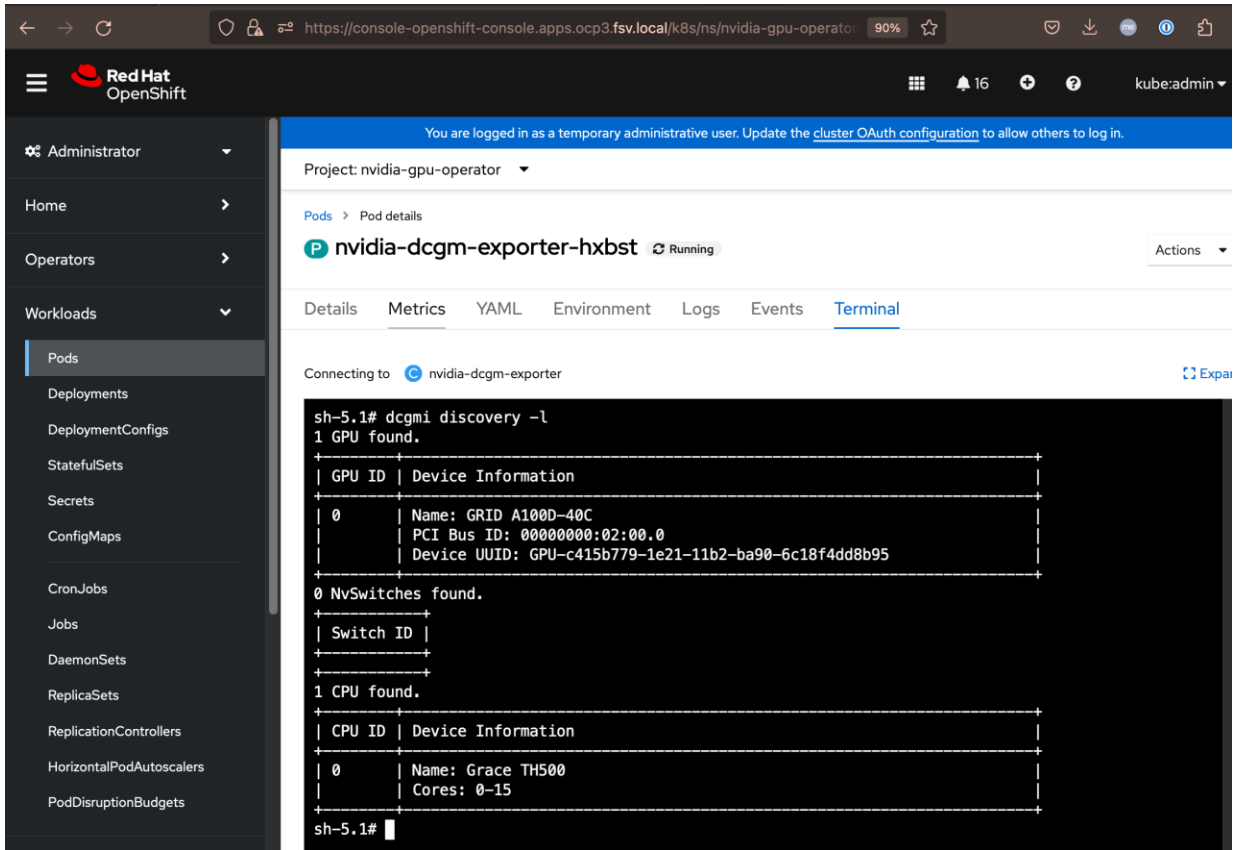
Table 10. OpenShift Metrics

| GPU Metric | Description |
|---------------|------------------------------|
| GPU SM Clocks | SM clock frequency in hertz. |

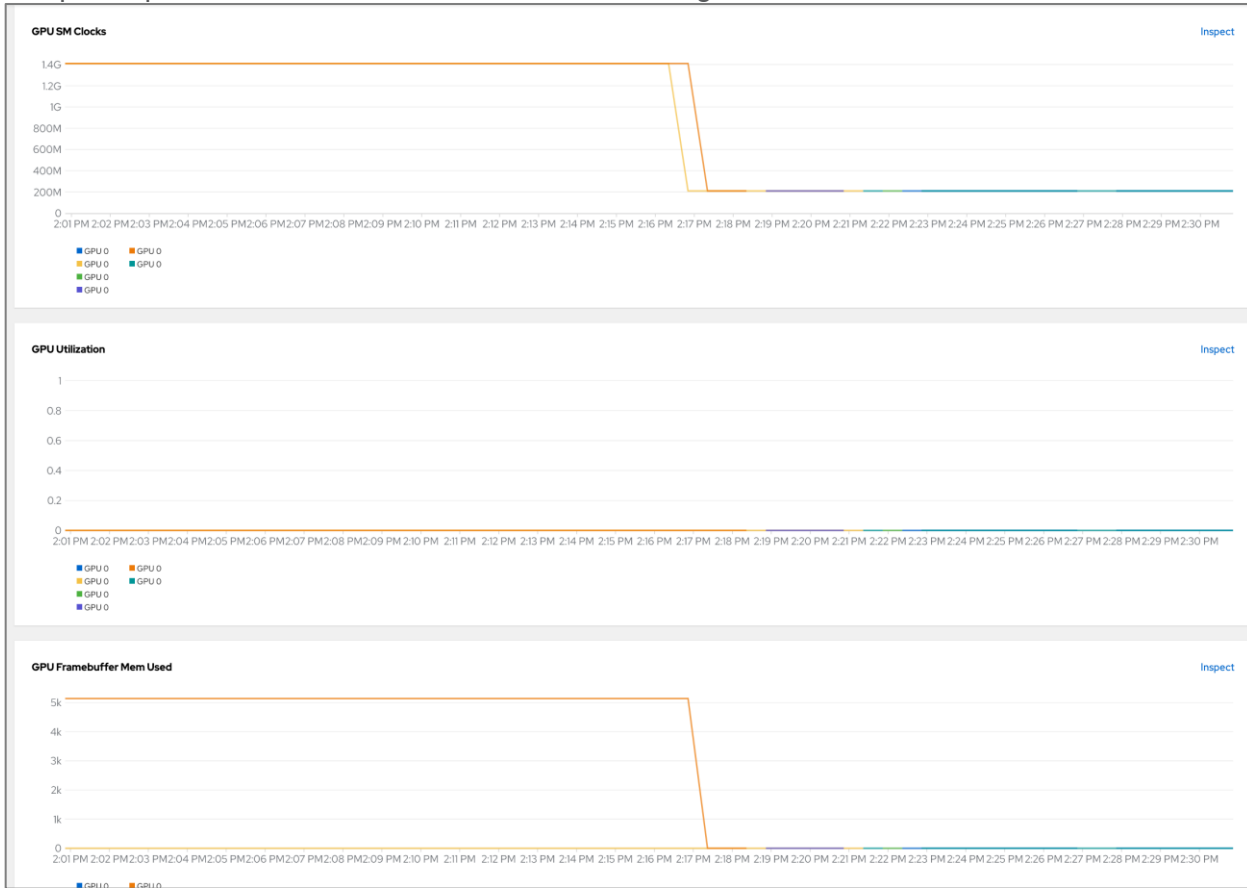
| GPU Metric | Description |
|--------------------------|--|
| GPU Utilization | GPU utilization, percent. |
| GPU Framebuffer Mem Used | Frame buffer memory used in MB. |
| Tensor Core Utilization | Ratio of cycles the tensor (HMMA) pipe is active, percent. |

In this solution, because we are using virtual GPUs, the metrics do not include power, temperature and other metrics related to the physical card. For the physical data, we must use the metrics available from VMware.

To verify that the vGPU is seen, you can execute the command: **dcgmi discovery -l** on the exporter pod as shown in the screenshot below:



Sample output from DCGM dashboard is shown in the figure below:



vGPU Monitoring using NVIDIA CLI tool for VMware vSphere and Red Hat OpenShift

NVIDIA provides the **nvidia-smi** CLI tool to collect GPU metrics and other details from the GPU in both OpenShift and VMware as outlined below:

In Red Hat OpenShift, execute the following commands from OpenShift installer workstation:

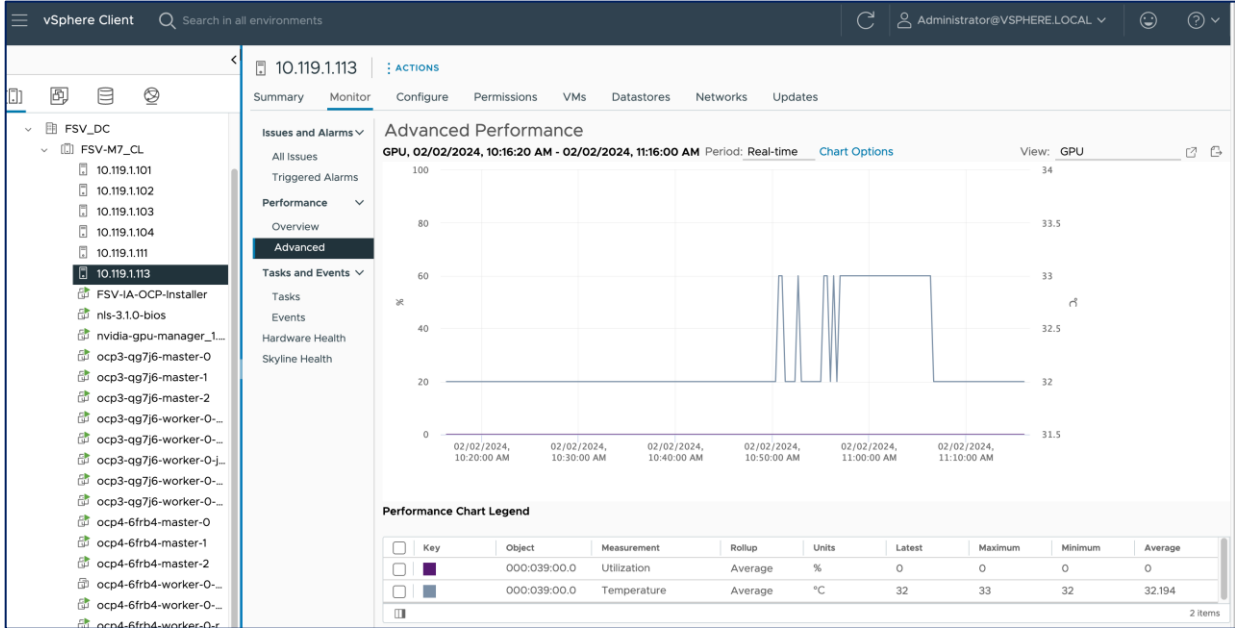
- `oc exec -it nvidia-driver-daemonset-<version> -- nvidia-smi`
- `oc exec -it nvidia-driver-daemonset-<version> -- nvidia-smi -q` (more info, provides licensing info, features enabled, and so on)


```

[root@FSV-ESXi-Host-13:~] nvidia-smi vgpu -l
Fri Feb  2 15:49:34 2024
+-----+
| NVIDIA-SMI 525.60.12                Driver Version: 525.60.12   |
+-----+-----+-----+-----+-----+
| GPU  Name          | Bus-Id          | GPU-Util  |
| vGPU ID  Name      | VM ID   VM Name  | vGPU-Util |
+-----+-----+-----+-----+-----+
|  0  NVIDIA A100 80GB PCIe | 00000000:39:00.0 | 0%        |
|      3251634281  GRID A100D-40C | 2105234  ocp3-qg7j6-worke... | 0%        |
|      3251634286  GRID A100D-40C | 2105254  ocp3-qg7j6-worke... | 0%        |
+-----+-----+-----+-----+-----+
Fri Feb  2 15:49:39 2024
+-----+
| NVIDIA-SMI 525.60.12                Driver Version: 525.60.12   |
+-----+-----+-----+-----+-----+
| GPU  Name          | Bus-Id          | GPU-Util  |
| vGPU ID  Name      | VM ID   VM Name  | vGPU-Util |
+-----+-----+-----+-----+-----+
|  0  NVIDIA A100 80GB PCIe | 00000000:39:00.0 | 0%        |
|      3251634281  GRID A100D-40C | 2105234  ocp3-qg7j6-worke... | 0%        |
|      3251634286  GRID A100D-40C | 2105254  ocp3-qg7j6-worke... | 0%        |
+-----+-----+-----+-----+-----+
[root@FSV-ESXi-Host-13:~]

```

Figure 28. GPU Monitoring from VMware vCenter



Validation

This chapter contains the following:

- [Hardware and Software Matrix](#)
- [GPU Functional/Load Tests](#)
- [Use Cases](#)
- [Interoperability Matrices](#)
- [Bill of Materials](#)

Hardware and Software Matrix

[Table 11](#) lists the hardware and software components that were used to validate the solution in Cisco labs.

Table 11. Hardware/Software Matrix

| Component (PID) | Software | Notes |
|---|---------------------|---|
| MLOps | | |
| Red Hat OpenShift AI (deployed as an operator) | 2.6.0 | Involves multiple pre-integrated & custom software components |
| GPU | | |
| NVIDIA AI Enterprise Software (NVAIE) | 3.0 | |
| NVIDIA GPU Operator | 23.9.1 | |
| NVAIE vGPU Driver (Guest) | 525.60.13 | Uses CUDA Driver 11.8 with support for CUDA applications up to 12.0 |
| Kubernetes (K8s) | | |
| Red Hat OpenShift | | |
| Red Hat OpenShift | 4.13.14 | This version was deployed by IPI installer when the cluster was first deployed. |
| Red Hat Node Feature Discovery Operator | 4.13.0-202401161111 | identifies and labels GPU |
| Red Hat OpenShift Pipelines | 1.13.1 | For ML Automation Pipelines |
| Pure Storage | | |
| Portworx (PX) Enterprise | 23.10.2 (Operator) | 3.0.3 (PX version) 23.8.0 (Stork Version) |
| Virtualization | | |
| VMware vCenter | 8.0 | |
| VMware vSphere | 8.0 | Supported VMware version on |

| Component (PID) | Software | Notes |
|---|--|--|
| | | HCL |
| Compute | | |
| Cisco UCS X-Series | | |
| Cisco UCS 6536 Fabric Interconnects (UCS-FI-6536) | 4.2(3e) | Intersight recommended version (not the latest) |
| Cisco UCS X9508 Chassis (UCSX-9508) | N/A | |
| Cisco UCS X9108-100G IFM (UCSX-I-9108-100G) | N/A | |
| Cisco UCS X210c M7 Compute Nodes (UCSX-210C-M7) | 5.2(0) | |
| Cisco UCS X440p PCIe Node () | N/A | |
| NVIDIA GPU (Cisco UCSC-GPU-A100-80) NVIDIA A100 PCIe FHFL DS 80GB HBM2e 300W (NON-CEC) | FW: 92.00.A0.00.05-1001.0230.00.03 Host Driver: 525.60.12 | NVIDIA NVAIE Driver Same FW & driver versions in UCS 4.3(2) & 5.2(0) bundles No GPU support listed in 5.1(0-1) |
| Cisco VIC 15231 MLOM (UCSX-ML-V5D200G) | FW: 5.3(2) | 2x100G mLOM |
| ESXi nenic (RDMA) driver | 2.0.11.0-1OEM.800.1.0.20143090 | |
| Storage | | |
| Pure Storage FlashArray//X50 | Purity 6.4.10 | |
| Pure Storage FlashBlade//S200 | Purity 4.1.12 | |
| Network | | |
| Cisco Nexus 93600CD-GX | 10.2(6) | Top-of-rack 100/400GbE switches |
| Other | | |
| Cisco Intersight | N/A | |
| Cisco Intersight Assist | 1.0.9-589 (may get upgraded) | Deployed OVA version: 1.0.9-588; Updated to the version shown |

GPU Functional/Load Tests

The following GPU focused validation was completed:

- GPU Functional Validation – Sample CUDA Application.
- GPU Stress/Load Test using GPU Burn Tests from: <https://github.com/wilicc/gpu-burn>. The test iterates up to max. GPU utilization to ensure that the GPU (vGPU in this case) is performing (Tflop/s) as it should before we add AI/ML workloads to Red Hat OpenShift.
- Same PyTorch script executed from Jupyter Notebooks on Red Hat OpenShift – see **Sample GPU Tests** folder in <https://github.com/ucs-compute-solutions/FlashStack-OpenShift-AI>

The next few figures show the results from the mentioned tests.

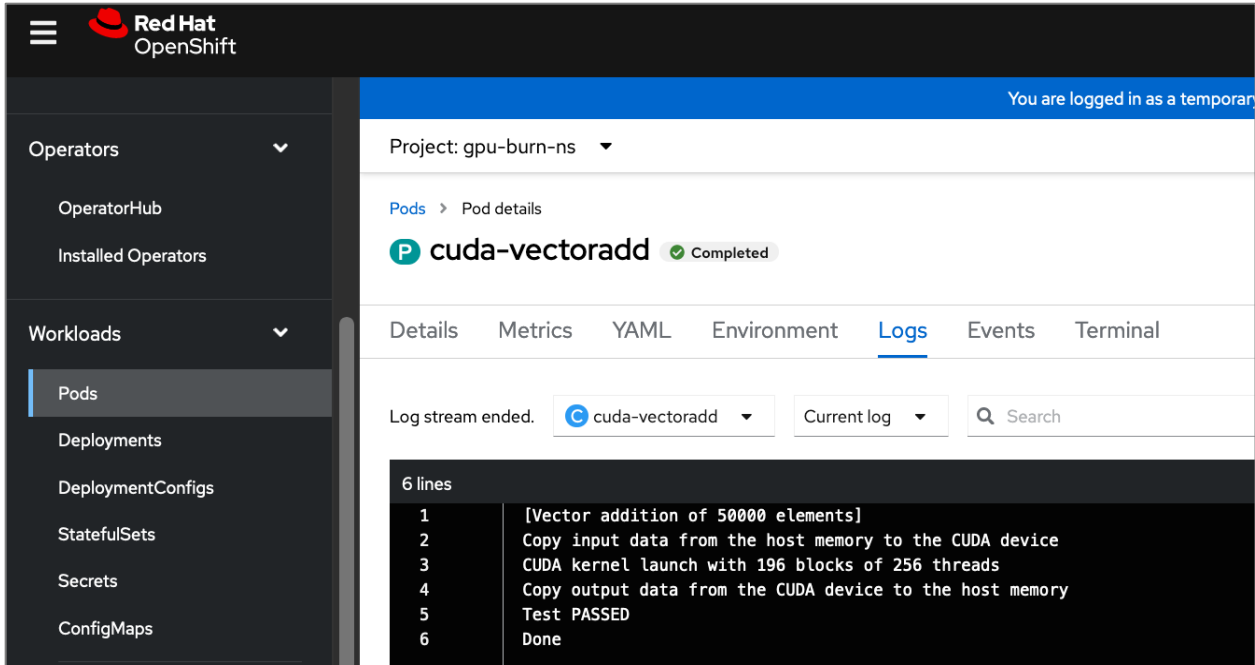
Sample CUDA Application Test

Configuration YAML file:

```
apiVersion: v1
kind: Pod
metadata:
  name: vectoradd
spec:
  restartPolicy: OnFailure
  containers:
  - name: vectoradd
    image: nvidia/samples:vectoradd-cuda11.6.0-ubi8
    resources:
      limits:
        nvidia.com/gpu: 1
    securityContext:
      capabilities:
        add: ["SYS_ADMIN"]
```

You can deploy the above configuration on the OpenShift cluster as outlined below:

```
[administrator@FSV-AI-OCP-Installer OCP3]$ vi cuda-vectoradd.yaml
[administrator@FSV-AI-OCP-Installer OCP3]$ oc project
Using project "nvidia-gpu-operator" on server "https://api.ocp3.fsv.local:6443".
[administrator@FSV-AI-OCP-Installer OCP3]$ oc apply -f cuda-vectoradd.yaml
pod/cuda-vectoradd created
[administrator@FSV-AI-OCP-Installer OCP3]$
```



Sample GPU Burn Test

The specifics of this test can be found in the GitHub repo provided earlier. Results of executing the test are provided in this section.

```
=====
==  CUDA  ==
=====
```

CUDA Version 12.0.0

Container image Copyright (c) 2016-2023, NVIDIA CORPORATION & AFFILIATES. All rights reserved.

This container image and its contents are governed by the NVIDIA Deep Learning Container License.
By pulling and using the container, you accept the terms and conditions of this license:
<https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license>

A copy of this license is made available in this container at /NGC-DL-CONTAINER-LICENSE for your convenience.

```
|
GPU 0: GRID A100D-40C (UUID: GPU-ef5a53d2-34d3-11b2-99cb-146bdf8cfaed)
Using compare file: compare.ptx
Burning for 60 seconds.
```

<REMOVED INTERMEDIATE RESULTS>

```
.
.
.
```

```
88.3% proc'd: 640 (18514 Gflop/s)  errors: 0  temps: --
88.3% proc'd: 640 (18514 Gflop/s)  errors: 0  temps: --
90.0% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
96.7% proc'd: 768 (18466 Gflop/s)  errors: 0  temps: --
100.0% proc'd: 896 (18449 Gflop/s)  errors: 0  temps: --
    Summary at:  Fri Dec 1 14:49:00 UTC 2023
```

Killing processes with SIGTERM (soft kill)

Using compare file: compare.ptx

Burning for 60 seconds.

Initialized device 0 with 40955 MB of memory (37077 MB available, using 33369 MB of it), using FLOATS

Results are 268435456 bytes each, thus performing 128 iterations

Freed memory for dev 0

Unitted cublas

done

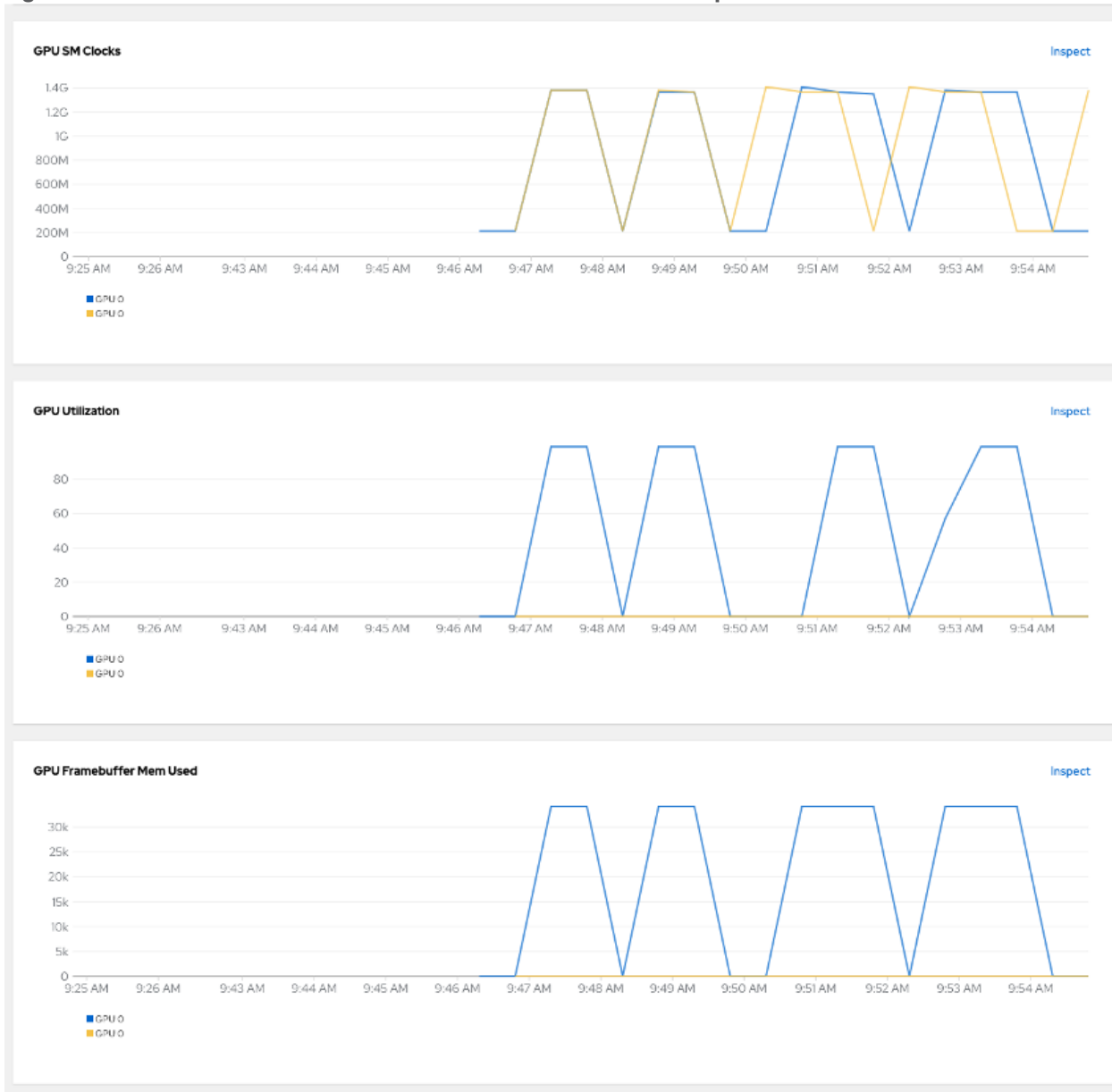
Tested 1 GPUs:

GPU 0: OK

Figure 29. GPU Burn Test Results - Output from nvidia-smi

```
[administrator@FSV-AI-OCP-Installer OCP3]$ oc exec -it nvidia-driver-daemonset-413.92.202309261804-0-zshvt -- nvidia-smi
Fri Dec 1 14:54:19 2023
+-----+
| NVIDIA-SMI 525.60.13    Driver Version: 525.60.13    CUDA Version: 12.0    |
+-----+-----+-----+-----+-----+-----+
| GPU  Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+
|  0  GRID A100D-40C    On      | 00000000:02:00.0 Off  |           0          |
| N/A  N/A    P0     N/A /  N/A | 34133MiB / 40960MiB |    99%      Default |
+-----+-----+-----+-----+-----+-----+
|
+-----+-----+-----+-----+-----+-----+
| Processes:
| GPU  GI  CI       PID  Type  Process name          GPU Memory
|   ID  ID             |          |      |                   |      Usage
+-----+-----+-----+-----+-----+-----+
|  0   N/A N/A     425634   C   ./gpu_burn            34069MiB
+-----+-----+-----+-----+-----+-----+
[administrator@FSV-AI-OCP-Installer OCP3]$
```

Figure 30. GPU Burn Test Results - DCGM Dashboard on Red Hat OpenShift



Use Cases

The AI/ML use cases and other testing that were validated for this effort are listed below. The code for the use cases can be found in the **Use Cases** folder on GitHub: <https://github.com/ucs-compute-solutions/FlashStack-OpenShift-AI>.

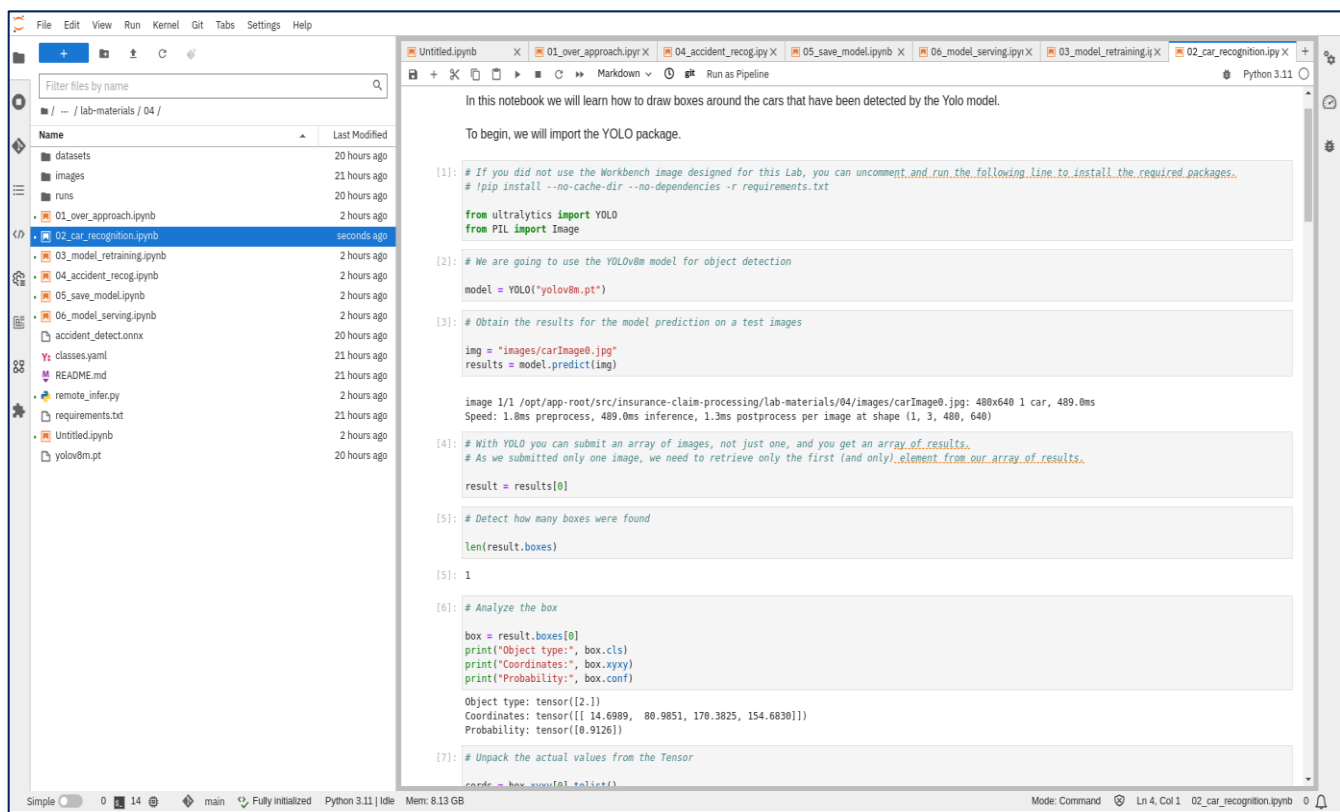
- **Fraud Detection** - Basic validation of a simple model using TensorFlow across the MLOps lifecycle. This model is an example in which transactions are analyzed using previous labeled data as either fraudulent or valid. This model would generally be called as part of real time transaction processing in financial institutions.
- **Object Detection** - Validation of a more advanced predictive AI Model. In this case we used PyTorch and YOLOv8 object detection. Starting from an open-source model, we retrain that model on new labeled data giving the ability for the model to detect car accidents. While not necessary, we see the benefits of using GPUs to reduce training time. As a service API, this

model could be consumed by applications using traffic cameras to detect accidents or other uses.

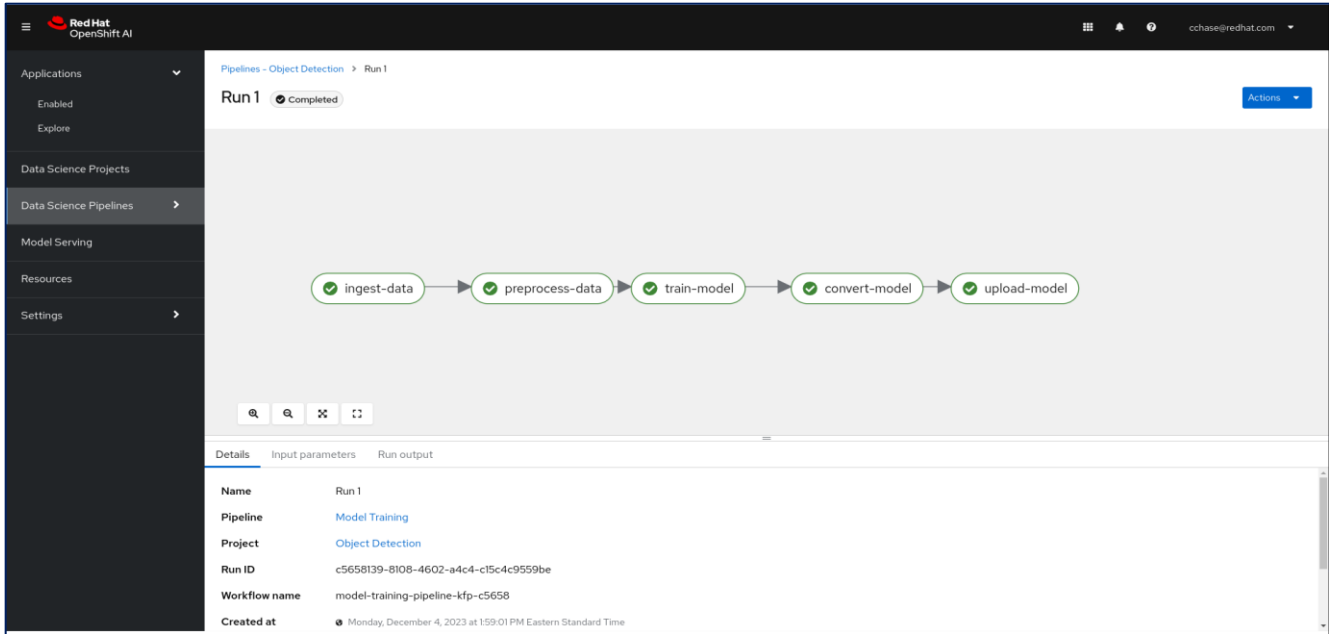
- **Text to Image** - Used to determine viability of Generative AI on the platform. Throughout the MLOps lifecycle, GPU accelerators are required. We started stable diffusion for image generation using PyTorch and CUDA 11.8. The demo involves fine-tuning the model using a small amount of custom data, exporting, and saving model in ONNX format to a model repo, and operationalizing the model into production using an inferencing server for use by application teams.

For each use case, the following three overarching tasks of the OpenShift AI MLOps workflow were tested:

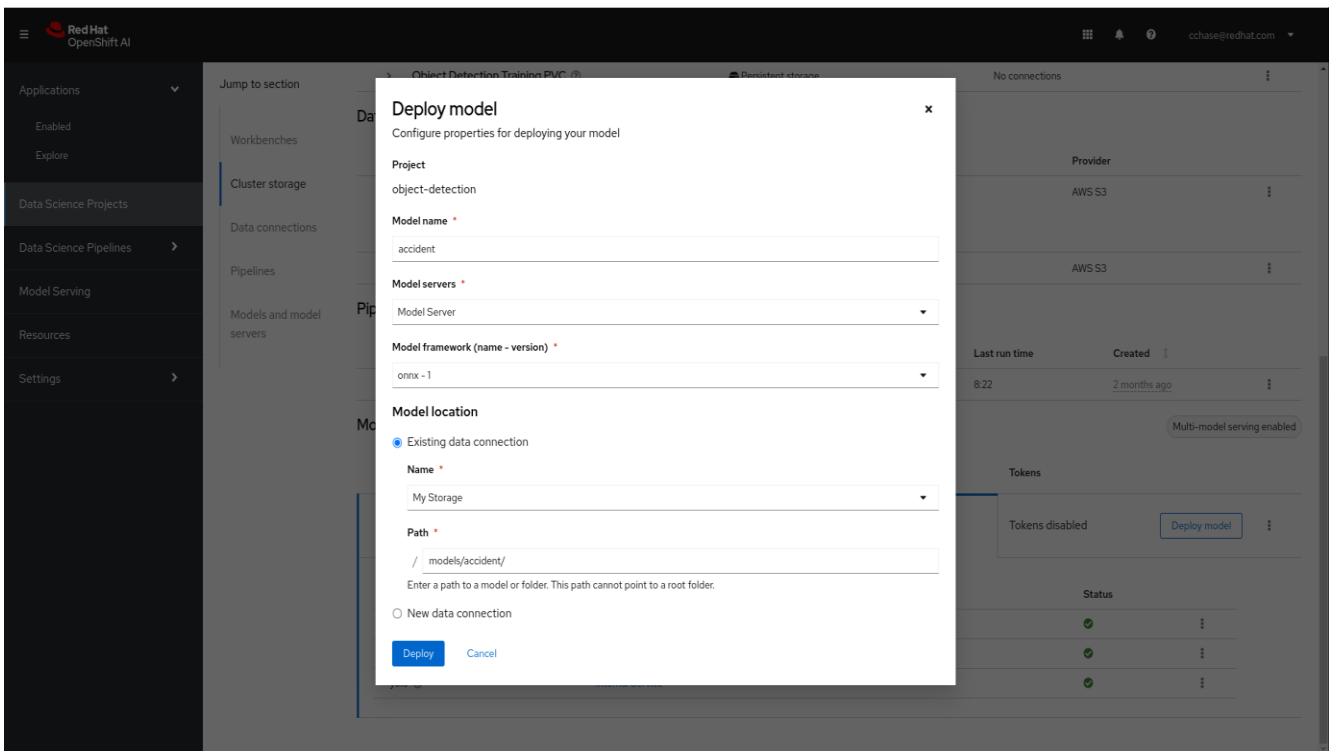
Model development - Conducting exploratory data science in a development environment with access to core AI/ML libraries and frameworks. Each use case starts with accessing the relevant model (for example, YOLO or You Only Look Once real time object detection model) from a public repository ([RoboFlow](#)). This includes creating the code for data extraction, validation, preparation, model training, model evaluation, and model testing. The Jupyter Notebook that is being used for this work, allows data scientists to describe key steps iteratively, such as describe the step, then write, and execute the code for that step before moving onto the next.



Lifecycle Management - Create repeatable data science pipelines for model training and validation. This includes everything necessary to be able to continuously train, save, and deploy models.



Model serving - Deploy models to be consumed from applications using protocols such as REST and gRPC. By saving models in a portable format, we leverage open-source model servers to create consistent APIs without requiring bespoke code to be developed for each model. When deployed, each model is monitored by OpenShift AI and consumed by authorized applications.



The served model is then consumed by an application that checks for the severity of a crash to generate an output such as the one shown below:



Interoperability Matrices

The interoperability information for the different components in the solution are summarized in [Table 12](#).

Table 12. Interoperability

| Component | Interoperability Matrix and Other Relevant Links |
|---|---|
| Cisco UCS Hardware Compatibility Matrix (HCL) | https://ucshcltool.cloudapps.cisco.com/public/ |
| VMware Compatibility Guide (VM Direct Path IO for GPU/PCIe Passthrough) | https://www.vmware.com/resources/compatibility/search.php?deviceCategory=vmdirect |
| VMware Product Interoperability Matrix | https://interopmatrix.vmware.com/Interoperability |
| NVIDIA AI Enterprise Qualification and Certification | https://www.nvidia.com/en-us/data-center/data-center-gpus/qualified-system-catalog/?&searchTerm=Cisco |
| NVIDIA Driver Lifecycle, Release and CUDA Support | https://docs.nvidia.com/datacenter/tesla/drivers/index.html#lifecycle |
| NVIDIA vGPU Certification (Not for NVAIE and NVIDIA H100/A100/A30 – for these, see NVAIE Certification above) | https://www.nvidia.com/en-us/data-center/resources/vgpu-certified-servers/ |
| NVAIE 4.1 Product Support Matrix | https://docs.nvidia.com/ai-enterprise/4.1/product-support-matrix/index.html - support-matrix vmware-vmware |
| Portworx Support Information | https://docs.portworx.com/portworx-enterprise/install-portworx/prerequisites - supported-kubernetes-versions |

Conclusion

Operationalizing AI is a daunting task for any Enterprise. Taking AI/ML efforts from proof-of-concept to production-ready is a significant challenge due to the complexity associated with streamlining and managing data and machine learning that deliver production-ready models. Data must be collected, cleaned, and curated before it can be used by a ML pipeline. The model must be continually updated to keep up with the changing data. The constantly changing data is as critical as the model itself to maintain the accuracy and reliability of the model and its predictions. This requires some level of integration between data and ML pipelines. When the model is ready for production, the data and ML pipelines supporting the model must integrate into existing enterprise application and software delivery pipelines that are managed by different teams with different but established practices and technologies. But if you want to go further and implement continuous integration and continuous delivery (CI/CD) at scale, considering the number of models and applications that require ongoing updates and maintenance, is anything but simple or easy.

A critical strategic decision that enterprises can make, regardless of the size of the effort, is to have a plan for operationalizing AI that brings consistency and efficiency to the process. Instead of ad-hoc ML efforts that add technical debt with each AI/ML effort, it is important to adopt processes, tools, and best-practices that can continually deliver and maintain models with speed and accuracy. An essential step towards this goal is to implement MLOps for your AI/ML efforts. MLOps brings successful DevOps practices from traditional application and software development into machine learning, to foster collaboration and other practices such as automation and CI/CD that accelerate model delivery. The FlashStack AI solution using Red Hat OpenShift AI delivers a complete platform for MLOps and AI/ML workloads, featuring pre-integrated tools and technologies to accelerate AI/ML efforts and operationalize AI in a repeatable manner, with consistency and efficiency.

Red Hat OpenShift AI in this solution, powered by FlashStack for AI, provides a foundational reference architecture for MLOps. Running on Red Hat OpenShift, OpenShift AI leverages FlashStack VSI to provide the foundational infrastructure for AI/ML workloads and MLOps. The FlashStack portfolio has a proven track record in enterprise data centers, delivering a high-performance and flexible architecture for a range of demanding enterprise applications including SAP, Oracle, HPC and graphics-accelerated VDI. Deploying FlashStack designs are easy with Infrastructure as Code (IaC) automation, available on Cisco and GitHub repositories for enterprise use. Cisco Intersight in the solution, continually delivers features that simplify enterprise IT operations, providing comprehensive management and visibility across all elements of the FlashStack datacenter, including GPUs and sustainability dashboards for monitoring power consumption.

FlashStack of AI extends the capabilities of FlashStack, simplifying infrastructure deployments for AI while accelerating AI/ML efforts by reducing complexity. As enterprise AI infrastructure needs grow, they can also scale and expand with ease and flexibility. The FlashStack for AI design uses Cisco UCS X-Series modular platform with the newest Cisco UCS M7 servers, Cisco UCS X440p PCIe nodes with NVIDIA GPUs, all centrally managed from the cloud using Cisco Intersight. Red Hat OpenShift provides Kubernetes container orchestration and management for AI/ML workloads and MLOps. The NVIDIA AI Enterprise software provides key capabilities in the solution, including virtual GPUs, GPU operator, GPU drivers and optimizations. Portworx Enterprise from Pure Storage, backed by Pure Storage FlashArray and FlashBlade meets all the storage requirements (image registry, model repository, pipeline artifacts, persistent storage for containers) for MLOps and the AI/ML workloads it manages.

This CVD provides a comprehensive solution for hosting AI/ML workloads in enterprise data centers. Coupled with OpenShift AI for MLOps, the CVD delivers a scalable solution that enterprises can use to accelerate AI/ML efforts and operationalizing AI quickly.

About the Authors

Archana Sharma, Technical Marketing, Cisco UCS Compute Solutions, Cisco Systems Inc.

Archana Sharma is a Technical Marketing Engineer with over 20 years of experience at Cisco on a variety of technologies that span Data Center, Desktop Virtualization, Collaboration, and other Layer2 and Layer3 technologies. Archana currently focusses on the design and deployment of Cisco UCS based solutions for Enterprise data centers, specifically Cisco Validated designs and evangelizing the solutions through demos and industry events such as Cisco Live. Archana holds a CCIE (#3080) in routing and switching and a bachelor's degree in electrical engineering from North Carolina State University.

Christopher Chase, Principal Marketing Manager, OpenShift AI, Red Hat

Chris Chase has 17 years of expertise across leading tech firms as software developer, software architect and technical marketing. His expertise includes AI and ML technologies, full-stack development, and software architecture. Known for translating user feedback into practical solutions and adept at prototyping, workshops, and strategic guidance. Proficient in OpenShift, Kubernetes, and MLOps, with a strong track record in deploying complex applications and integrating diverse technologies. He is responsible for various ML demos using OpenShift AI, most notably Generative AI Text-to-Image demo to generate images of his dog, Teddy.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the authors would like to thank:

- Chris O'Brien, Senior Director, Cisco Systems, Inc.
- John George, Technical Marketing Engineer, Cisco Systems, Inc.
- Rohit Mittal, Product Manager, Cisco Systems, Inc.
- Karl Eklund, Principal Architect, Red Hat
- Younes Ben Brahim, Senior Product Marketing Manager, Red Hat
- Vijay Bhaskar Kulari, Senior Solutions Architect, Pure Storage, Inc.
- Craig Waters, Technical Director, Pure Storage, Inc.

Appendices

This appendix contains the following:

- [Appendix A - References](#)

Appendix A - References

A complete list of the references used in this solution are provided below.

Red Hat OpenShift

- Red Hat OpenShift Operators: <https://www.redhat.com/en/technologies/cloud-computing/openshift/what-are-openshift-operators>
- Red Hat OpenShift Ecosystem catalog: https://catalog.redhat.com/software/search?deployed_as=Operator

FlashStack

- Cisco Design Zone for FlashStack CVDs: <https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/data-center-design-guides-all.html#FlashStack>
- FlashStack Compatibility Matrix: https://support.purestorage.com/FlashStack/Product_Information/FlashStack_Compatibility_Matrix

Automation

- GitHub repository for Cisco UCS solutions: <https://github.com/ucs-compute-solutions/>

Compute

- Cisco UCS Hardware Compatibility Matrix: <https://ucshcltool.cloudapps.cisco.com/public/>
- Cisco Intersight: <https://www.intersight.com>
- Cisco Intersight Managed Mode: https://www.cisco.com/c/en/us/td/docs/unified_computing/Intersight/b_Intersight_Managed_Mode_Configuration_Guide.html
- Cisco Unified Computing System: <http://www.cisco.com/en/US/products/ps10265/index.html>
- Cisco UCS 6536 Fabric Interconnects: <https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs6536-fabric-interconnect-ds.html>

Network

- Cisco Nexus 9000 Series Switches: <http://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html>
- Cisco MDS 9132T Switches: <https://www.cisco.com/c/en/us/products/collateral/storage-networking/mds-9100-series-multilayer-fabric-switches/datasheet-c78-739613.html>

Storage

- Portworx Enterprise Documentation: <https://docs.portworx.com/portworx-enterprise>
- Pure Storage FlashArray//X: <https://www.purestorage.com/products/nvme/flasharray-x.html>

-
- Pure Storage FlashBlade//S: <https://www.purestorage.com/products/unstructured-data-storage.html>
 - Pure Storage FlashArray Compatibility Matrix: https://support.purestorage.com/FlashArray/Getting_Started_with_FlashArray/FlashArray_Compatibility_Matrix
 - Pure Storage FlashBlade Compatibility Matrix: https://support.purestorage.com/FlashBlade/Getting_Started_with_FlashBlade/FlashBlade_Compatibility_Matrix

Virtualization

- VMware and Cisco Unified Computing System: <http://www.vmware.com/resources/compatibility>
- VMware vCenter Server: <http://www.vmware.com/products/vcenter-server/overview.html>
- VMware vSphere: <https://www.vmware.com/products/vsphere>

Feedback

For comments and suggestions about this guide and related guides, join the discussion on [Cisco Community](https://cs.co/en-cvds) at <https://cs.co/en-cvds>.

CVD Program

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unified Computing System (Cisco UCS), Cisco UCS B-Series Blade Servers, Cisco UCS C-Series Rack Servers, Cisco UCS S-Series Storage Servers, Cisco UCS X-Series, Cisco UCS Manager, Cisco UCS Management Software, Cisco Unified Fabric, Cisco Application Centric Infrastructure, Cisco Nexus 9000 Series, Cisco Nexus 7000 Series, Cisco Prime Data Center Network Manager, Cisco NX-OS Software, Cisco MDS Series, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trade-marks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. (LDW_P2)

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0809R)

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)