# Cisco UCS Infrastructure for AI and Machine Learning with Red Hat OpenShift Container Platform 3.11

Use a Cisco Validated Design to deploy OpenShift on Cisco UCS infrastructure.

**Last Updated**: May 21, 2019

# Contents

## Executive summary

Cisco® Validated Designs are the foundation for designing systems that use Cisco infrastructure and for implementing complex customer deployments. Validated designs incorporate products and technologies from a broad portfolio of enterprise, service provider, and commercial systems into solutions that are designed, tested, and fully documented to help ensure fast, reliable, consistent, and predictable customer deployments.

Cisco provides converged infrastructure that uses the Cisco Unified Computing System™ (Cisco UCS®) to integrate systems that help make IT operations more cost effective, efficient, and agile. These validated converged architecture designs promote customer success and reduce risk through strategic guidance and expert advice.

Cisco is a leader in integrated systems. We offer a diversified portfolio of converged infrastructure solutions that integrate a wide range of technologies and products into cohesive validated and supported solutions to address our customers' business needs.

The design described in this document is intended to provide a best-in-class solution for graphics processing unit (GPU)–enabled container workloads using Red Hat OpenShift Container Platform 3.11. The reference architecture discussed in this design guide provides a methodology for deploying a highly available Red Hat OpenShift Container Platform solution for AI/ML workloads on Cisco UCS infrastructure.

## Solution overview

This section introduces the Red Hat OpenShift and Cisco UCS solution.

### Introduction

Many organizations today are seeking to implement DevOps initiatives and application platforms that smooth the deployment process to facilitate their digital transformation journey. Though still in the early stages of development, Docker container packaging and Kubernetes container orchestration are emerging as important technologies to accelerate the digital transformation process. Containers offer considerable value to IT by providing predictability when applications move from the build stage to actual deployment. Developers can be confident that an application will perform the same when it is run in a production environment as it did when it was built, and that operations teams and administrators will have the precise tools they need to operate and maintain the application.

Red Hat OpenShift Container Platform provides a set of container-based open-source tools that support digital transformation. These tools accelerate application development while making optimal use of infrastructure. The availability of highly secure operating systems helps organizations deploy an environment that can withstand continuously changing security threats, helping keep applications highly secure.

With OpenShift, organizations can use a cloud delivery model, simplifying continuous delivery of applications and services through a cloud-native platform. Built on proven open-source technologies, OpenShift also gives development teams multiple modernization options to enable a smooth transition to a microservices architecture and to the cloud for existing traditional applications.

Cisco UCS servers are designed to adapt to meet rapidly changing business needs, supporting just-in-time deployment of new computing resources to meet requirements and improve business outcomes. With Cisco UCS, you can tune your environment to support the unique needs of each application while powering all your server workloads on a centrally managed, highly scalable system. Cisco UCS brings the flexibility of nonvirtualized and virtualized systems in ways that no other server architecture does, lowering costs and improving your return on investment (ROI).

The Cisco UCS M5 servers used in this solution are built on powerful Intel® Xeon® Scalable processors. These unified yet modular, scalable, high-performing servers are built on an infrastructure-as-code model that enables strong integration and continuous delivery of distributed applications.

NVIDIA provides a number of products that address emerging artificial intelligence (AI) and machine-learning workload use cases with GPU-based hardware acceleration and complementary software. The Cisco UCS C480 ML M5 Rack Server can integrate up to eight NVIDIA Tesla V100 GPUs in a standard form factor, providing improved performance in a relatively small footprint well suited to the data center, and Red Hat Enterprise Linux and OpenShift, together with NVIDIA technology, can serve as primary platforms for AI and other GPU-accelerated workloads.

In addition, NVIDIA GPU Cloud (NGC) containers can be deployed on OpenShift. NVIDIA uses NGC containers to deliver integrated software stacks and drivers to run GPU-optimized machine-learning frameworks such as TensorFlow, Caffe2, PyTorch, and MXNet, and many others.

Cisco, NVIDIA, and Red Hat have worked together to develop a best-in-class solution for delivering AI and machine-learning capabilities to enterprises. This solution enables organizations to develop, deploy, and manage containers in on-premises and private and public cloud environments through automation using a robust platform such as OpenShift.

### Implementation overview

This document provides guidance for preparing, provisioning, deploying, and managing a Cisco UCS and Red Hat OpenShift Container Platform solution to support AI and machine-learning workloads on GPU-enabled containers. The OpenShift platform uses an on-premises environment with additional container-native storage components on Cisco UCS C-Series Rack Servers to address the persistent storage needs of stateful applications.

### Solution benefits

The main benefits of this solution include the following:

- Cisco UCS
  - Reduced data center complexity through Cisco UCS infrastructure, which provides a single management control plane for hardware lifecycle management
  - Easy deployment and scaling
  - Superior scalability and high availability
  - Computing form-factor independence
  - Optimized hardware footprint for production and development and test deployments
- Red Hat OpenShift Container Platform
  - Strong, role-based access controls, with integration with enterprise authentication systems
  - Powerful, web-scale container orchestration and management with Kubernetes
  - Integrated Red Hat Enterprise Linux Atomic Host, optimized for running containers at scale with Security-Enhanced Linux (SELinux) enabled for strong isolation
  - Integration with public and private registries
  - Integrated continuous integration and continuous development (CI/CD) tools for secure DevOps practices
  - New model for container networking
  - Modern application architecture that supports microservices

- Consistent application platform for hybrid cloud deployments
- Support for remote storage volumes
- Persistent storage for stateful cloud-native containerized applications

- NVIDIA GPU Cloud containers
  - Powerful and easy-to-deploy GPU-accelerated software for deep learning, machine learning, and high-performance computing (HPC), delivering fast results
  - Quick deployment of containers, with less complexity than is typically associated with software setup
  - Container registry that provides researchers, data scientists, and developers with easy access to a comprehensive catalog of GPU-accelerated software for AI, machine learning, and HPC that takes full advantage of NVIDIA GPUs on the organization's premises and in the cloud

### Audience

The audience for this document includes sales engineers, field consultants, professional services, IT managers, partner engineers, IT architects, and customers who want to take advantage of infrastructure that is built to deliver IT efficiency and enable IT innovation. The reader of this document is expected to have the training and background necessary to install and configure Red Hat Enterprise Linux, Cisco Unified Computing System, and Cisco Nexus® switches. The reader should also have an understanding of AI and machine-learning workloads. In addition, knowledge of container platforms, preferably Red Hat OpenShift Container Platform, is required. External references are provided where applicable, and familiarity with these documents is highly recommended.

## Technology overview

This section provides a brief introduction of the various hardware and software components used in this solution.

### Cisco Unified Computing System

Cisco UCS is a state-of-the-art data center platform that unites computing, network, storage access, and virtualization into a single cohesive system.

The main components of Cisco UCS are:

- **Computing:** The system is based on an entirely new class of computing system that incorporates rack-mount and blade servers based on Intel Xeon processor E5 and E7 CPUs. Cisco UCS servers incorporate patented Cisco Extended Memory Technology to support applications with large data sets and more virtual machines per server.
- **Network:** The system is integrated on a low-latency, lossless, 40-Gbps unified network fabric. This network foundation consolidates LANs, SANs, and HPC networks, which are separate networks today. The unified fabric lowers costs by reducing the number of network adapters, switches, and cables and by decreasing the power and cooling requirements.
- **Virtualization:** The system unleashes the full potential of virtualization by enhancing the scalability, performance, and operational control of virtual environments. Cisco security, policy enforcement, and diagnostic features now extend into virtualized environments to better support changing business and IT requirements.
- **Storage access:** The system provides consolidated access to both SAN storage and network-attached storage (NAS) over the unified fabric. By unifying storage access, Cisco UCS can access storage over Ethernet (Network File System [NFS] or Small Computer System Interface over IP [iSCSI]), Fibre Channel, and Fibre Channel over Ethernet (FCoE). This capability allows customers to choose among storage access options and helps provide investment protection. In addition, server administrators can pre-assign storage-access policies for system connection to storage resources, simplifying storage connectivity and management for increased productivity.

Cisco UCS is designed to deliver:

- Reduced total cost of ownership (TCO) and increased business agility
- Increased IT staff productivity through just-in-time provisioning and mobility support
- A cohesive, integrated system that unifies the technology in the data center
- Industry standards supported by a partner ecosystem of industry leaders

### Cisco UCS Manager

Cisco UCS Manager provides unified, embedded management for all software and hardware components in Cisco UCS. Using Cisco Single Connect technology, it manages, controls, and administers multiple chassis for thousands of virtual machines. Administrators use the software to manage the entire Cisco UCS platform as a single logical entity through an intuitive GUI, a command-line interface (CLI), or an XML API. Cisco UCS Manager resides on a pair of Cisco UCS 6300 Series Fabric Interconnects using a clustered, active-standby configuration for high availability.

Cisco UCS Manager offers a unified embedded management interface that integrates server, network, and storage resources. It performs autodiscovery to detect inventory, manage, and provision system components that are added or changed. It includes a comprehensive set of XML APIs for third-party integration and exposes 9000 points of integration, facilitating custom development for automation and orchestration and providing organizations with outstanding system visibility and control.

Service profiles benefit both virtualized and nonvirtualized environments and increase the mobility of nonvirtualized servers: for instance, when moving workloads from server to server or taking a server offline to service or upgrade it. Profiles can also be used in conjunction with virtualization clusters to bring new resources online easily, complementing existing virtual machine mobility.

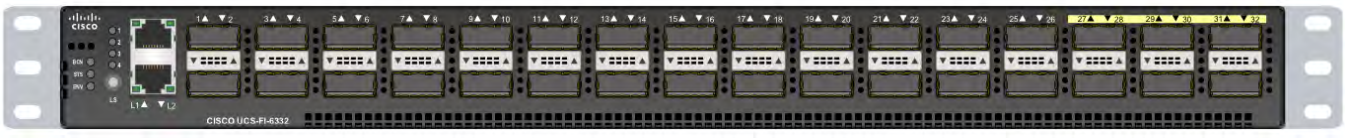
For more information about Cisco UCS Manager, see http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-manager/index.html.

### Cisco UCS 6300 Series Fabric Interconnects

Fabric interconnects provide a single point for connectivity and management for the entire system. Typically deployed as an active-active pair, the system's fabric interconnects integrate all components into a single, highly available management domain controlled by Cisco UCS Manager. The fabric interconnects manage all I/O operations efficiently and securely at a single point, resulting in deterministic I/O latency regardless of a server's or virtual machine's topological location in the system.

Cisco UCS 6300 Series Fabric Interconnects support bandwidth of up to 2.43 terabits per second (Tbps) for unified fabric with low-latency, lossless, cut-through switching that supports IP, storage, and management traffic using a single set of cables. The fabric interconnects provide virtual interfaces that terminate both physical and virtual connections equivalently, establishing a virtualization-aware environment in which blade servers, rack servers, and virtual machines are interconnected using the same mechanisms. The Cisco UCS 6332-16UP Fabric Interconnect (Figure 1) is a one-rack-unit (1RU) interconnect that offers up to 40 universal ports that can support up to twenty-four 40 Gigabit Ethernet, FCoE, or native Fibre Channel connections. In addition, it supports up to sixteen 1- and 10-Gbps FCoE or 4-, 8-, and 16-Gbps Fibre Channel unified ports.

**Figure 1.** Cisco UCS 6332-16UP Fabric Interconnect



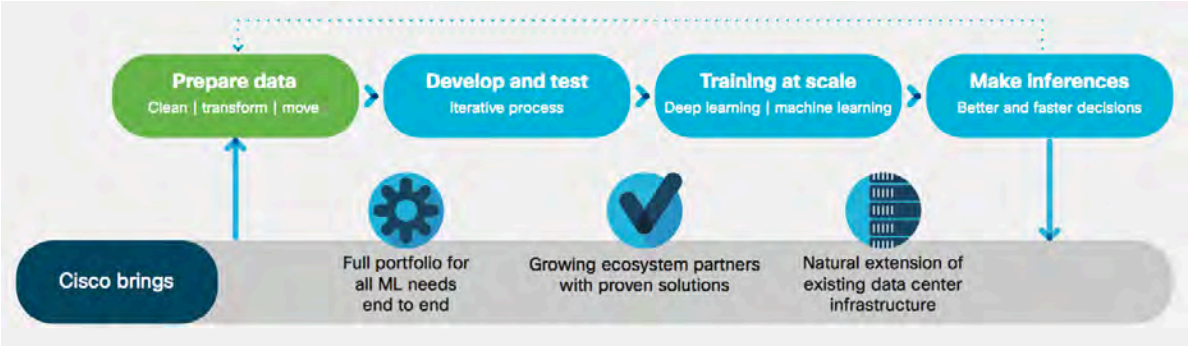

For more information, see https://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-6332-16up-fabric-interconnect/index.html.

### Cisco UCS C480 ML M5 Rack Server AI platform

The Cisco UCS C480 ML M5 Rack Server is the latest addition to the Cisco UCS server portfolio and its first server built from the start for AI and machine-learning workloads. With this addition, organizations have a complete range of computing options designed for each stage of the AI and machine-learning lifecycles, enabling organizations to extract more intelligence from their data and use it to make better, faster decisions (Figure 2).

The Cisco UCS C480 ML M5 server has a 4RU form factor and is specifically built for deep learning. It is storage and I/O optimized to deliver industry-leading performance for training models. It is designed for the most computation-intensive phase of the AI and machine-learning lifecycles: deep learning. This server integrates GPUs and high-speed interconnect technology with a large storage capacity and up to 100-Gbps network connectivity.

**Figure 2.** Support your data scientists with a complete portfolio of AI and machine-learning servers



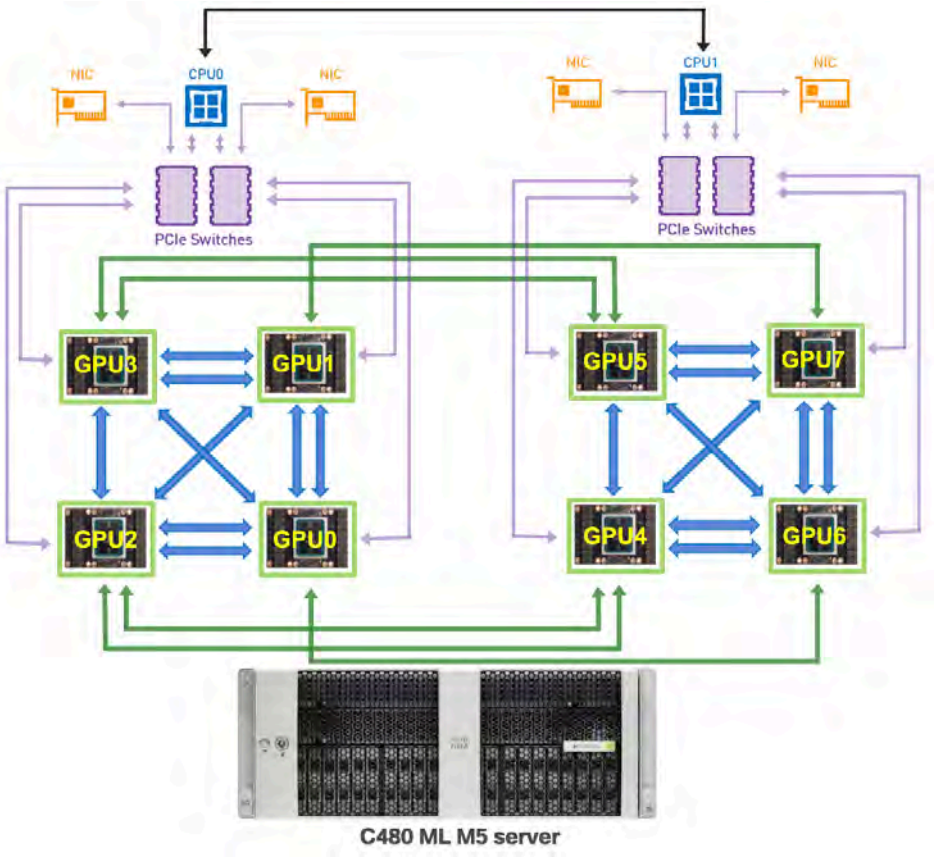

The Cisco UCS C480 ML M5 offers these features and benefits:

- **GPU acceleration:** Eight NVIDIA Tesla V100 SXM2 32-GB modules are interconnected with NVIDIA NVLink technology for fast communication across GPUs to accelerate computing. NVIDIA specifies TensorFlow performance of up to 125 teraflops per module, for a total of up to 1 petaflop of processing capability per server.
- **Internal NVLink topology:** NVLink is a high-speed GPU interconnect. Eight GPUs are connected through an NVLink cube mesh. Each NVLink is capable of 25 GBps of send and receive processing, for a total bandwidth of about 300 GBps among the eight GPUs.

Note that each GPU has six NVLinks. Thus, not all GPUs are directly connected through NVLink. Therefore, performance may be negatively affected if a particular training model has a heavy load of GPU-to-GPU communication.

The eight-GPU hybrid cube-mesh NVLink topology provides the highest bandwidth for multiple collective communication primitives, including broadcast, gather, all-reduce, and all-gather primitives, which are important to deep learning. Figure 3 shows the NVLink topology of UCS C480 ML M5.

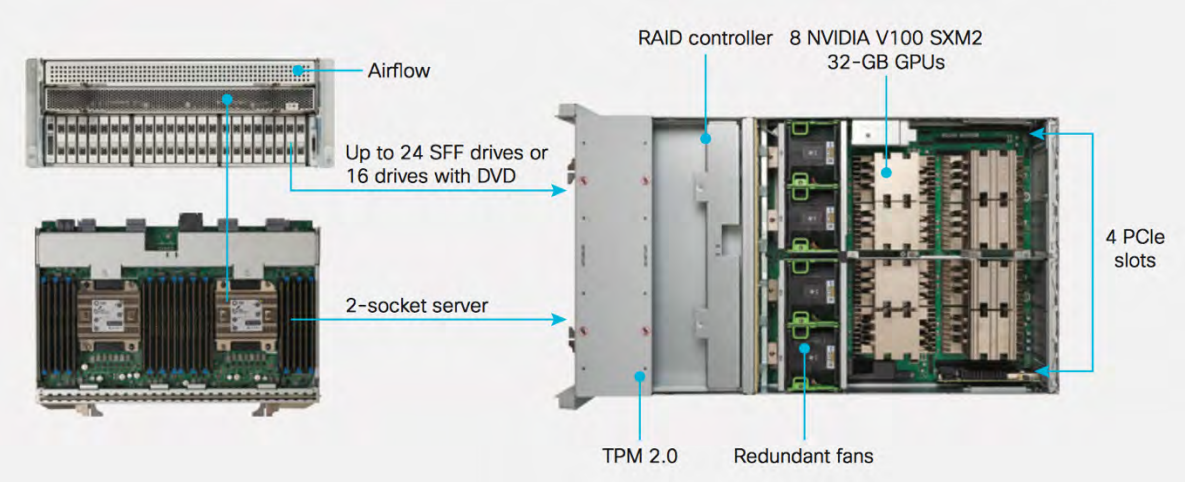
**Figure 3.** NVIDIA NVLink topology of UCS C480 ML M5




- **The latest Intel Xeon Scalable CPUs:** Two CPUs with up to 28 cores each manage the machine-learning process and send calculations to the GPUs.
- **Storage capacity and performance:** Data locality can be important for deep-learning applications. Up to 24 hard-disk drives (HDDs) or solid-state disks (SSDs) store data close to where it is used and are accessed through a midplane-resident RAID controller. Up to six disk-drive slots can be used for NVMe drives, providing best-in-class performance.
- **Up to 3 TB of main memory:** The system uses fast 2666-MHz DDR4 DIMMs.
- **High-speed networking:** Two built-in 10 Gigabit Ethernet interfaces accelerate the flow of data to and from the server.
- **PCI Express (PCIe) expandability:** Four PCIe switches feed four x16 PCIe slots for high-performance networking. Options include Cisco UCS virtual interface cards (VICs) and third-party network interface cards (NICs), for up to 100-Gbps connectivity.
- **Unified management:** By expanding the Cisco UCS portfolio with the new Cisco UCS C480 ML M5 server, we continue to support any workload without adding management complexity.

Figure 4 shows the physical design of the server.

**Figure 4.** Cisco UCS C480 ML M5 Rack Server physical design




## NVIDIA Tesla V100 SXM2 with 32 GB of memory

The NVIDIA Tesla V100 (Figure 5) is the world's most advanced data center GPU ever built to accelerate AI, HPC, and graphics processing. Powered by NVIDIA Volta, the latest GPU architecture, the Tesla V100 offers the performance of up to 100 CPUs in a single GPU, enabling data scientists, researchers, and engineers to overcome challenges formerly considered insurmountable.

**Figure 5.** NVIDIA Tesla V100 SXM2 GPU

- **Volta architecture:** By pairing CUDA cores and Tensor cores within a unified architecture, a single server with Tesla V100 GPUs can outperform hundreds of commodity CPU servers for certain deep-learning applications.
- **Tensor core:** Equipped with 640 Tensor cores, the Tesla V100 delivers 125 teraflops of deep-learning performance. Thus, Tensor offers 12 times more floating-point operations per second (FLOPS) for deep-learning training and 6 times more FLOPS for deep-learning inference than NVIDIA Pascal GPUs.
- **Next-generation NVIDIA NVLink:** NVLink in the Tesla V100 delivers twice the throughput of the previous generation of technology. Up to eight Tesla V100 accelerators can be interconnected at up to 300 GBps to achieve the highest application performance possible on a single server.

- **Maximum-efficiency mode:** The new maximum-efficiency mode allows data centers to achieve up to 40 percent greater computing capacity per rack within the existing power budget. In this mode, the Tesla V100 runs at peak processing efficiency, providing up to 80 percent of the performance at half the power consumption.
- **Programmability:** The Tesla V100 is designed from the foundation to simplify programmability. Its new independent thread scheduling enables synchronization and improves GPU utilization by sharing resources among small jobs.

Every major deep-learning framework is optimized for NVIDIA GPUs, enabling data scientists and researchers to use AI for their work. When running deep-learning training and inference frameworks, a data center with Tesla V100 GPUs can save over 90 percent in server and infrastructure acquisition costs.

The Tesla V100 platform for deep-learning training offers these main features:

- Caffe, TensorFlow, and the Microsoft Cognitive Toolkit (previously called CNTK) are up to three times faster with the Tesla V100 than with the NVIDIA P100 GPU.
- 100 percent of the top deep-learning frameworks are GPU accelerated.
- The platform offers up to 125 teraflops of TensorFlow operations per GPU.
- The platform offers up to 32 GB of memory capacity per GPU.
- The platform offers memory bandwidth of up to 900 GBps per GPU.

### Cisco UCS C240 M5 Rack Server

The Cisco UCS C240 M5 Rack Server (Figure 6) is a two-socket, 2RU rack server offering industry-leading performance and expandability. It supports a wide range of storage and I/O-intensive infrastructure workloads, from big data and analytics to collaboration. Cisco UCS C-Series Rack Servers can be deployed as standalone servers or as part of a Cisco UCS managed environment to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' TCO and increase their business agility.

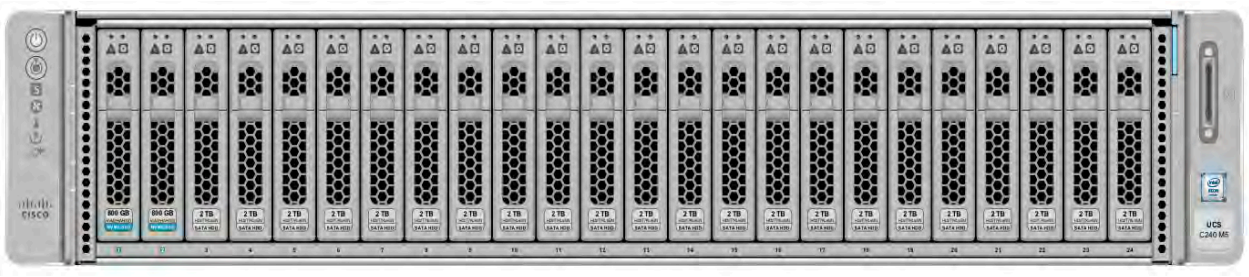
In response to ever-increasing computing and data-intensive real-time workloads, the enterprise-class Cisco UCS C240 M5 server extends the capabilities of the Cisco UCS portfolio in a 2RU form factor. It incorporates the Intel Xeon Scalable processors, supporting up to 20 percent more cores per socket, twice the memory capacity, and five times more Non-Volatile Memory Express (NVMe) PCI Express (PCIe) SSDs than the previous generation of servers. These improvements deliver significant performance and efficiency gains that will improve your application performance.

The Cisco UCS C240 M5 delivers outstanding levels of storage expandability with exceptional performance, with:

- The latest Intel Xeon Scalable CPUs, with up to 28 cores per socket
- Up to 24 DDR4 DIMMs for improved performance
- Up to 26 hot-swappable small-form-factor (SFF) 2.5-inch drives, including 2 rear hot-swappable SFF drives (up to 10 support NVMe PCIe SSDs on the NVMe-optimized chassis version) or 12 large-form-factor (LFF) 3.5-inch drives plus 2 rear hot-swappable SFF drives
- Support for a Cisco 12-Gbps SAS modular RAID controller in a dedicated slot, leaving the remaining PCIe Generation 3.0 slots available for other expansion cards
- Modular LAN-on-motherboard (mLOM) slot that can be used to install a Cisco UCS VIC without consuming a PCIe slot, supporting dual 10- or 40-Gbps network connectivity

- Dual embedded Intel x550 10GBASE-T LAN-On-motherboard (LOM) ports
- Modular M.2 or Secure Digital (SD) cards that can be used for boot

**Figure 6.** Cisco UCS C240 M5 Rack Server



For more information, see https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/datasheet-c78-739279.html.

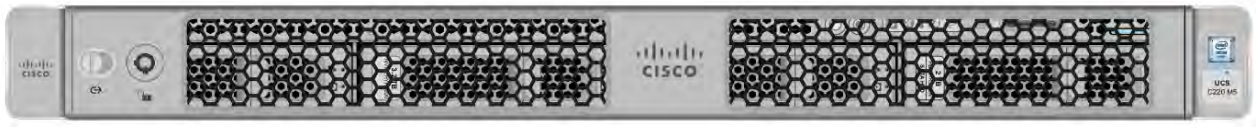
### Cisco UCS C220 M5 Rack Server

The Cisco UCS C220 M5 Rack Server (Figure 7) is among the most versatile general-purpose enterprise infrastructure and application servers. It is a high-density two-socket rack server that delivers industry-leading performance and efficiency for a wide range of workloads, including virtualization, collaboration, and bare-metal applications. The Cisco UCS C-Series Rack Servers can be deployed as standalone servers or as part of a Cisco UCS environment to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' TCO and increase their business agility. The Cisco UCS C220 M5 server extends the capabilities of the Cisco UCS portfolio in a 1RU form factor. It incorporates the Intel Xeon Scalable processors, supporting up to 20 percent more cores per socket, twice the memory capacity, 20 percent greater storage density, and five times more PCIe NVMe SSDs compared to the previous generation of servers. These improvements deliver significant performance and efficiency gains that will improve your application performance.

The Cisco UCS C220 M5 delivers outstanding levels of expandability and performance in a compact package, with:

- The latest Intel Xeon Scalable CPUs, with up to 28 cores per socket
- Up to 24 DDR4 DIMMs for improved performance
- Up to 10 SFF2.5-inch drives or 4 LFF 3.5-inch drives (77 TB of storage capacity with all-NVMe PCIe SSDs)
- Support for a Cisco 12-Gbps SAS modular RAID controller in a dedicated slot, leaving the remaining PCIe Generation 3.0 slots available for other expansion cards
- mLOM slot that can be used to install a Cisco UCS VIC without consuming a PCIe slot
- Dual embedded Intel x550 10GBASE-T LOM ports

**Figure 7.** Cisco UCS C220 M5 Rack Server



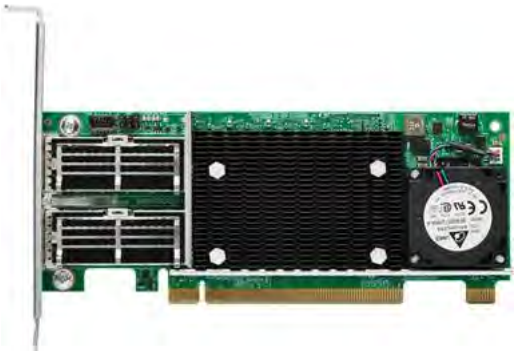
For more information, see https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/datasheet-c78-739281.html.

### Cisco UCS Virtual Interface Card 1385

The Cisco UCS VIC 1385 (Figure 8) is a Cisco innovation. It provides a policy-based, stateless, agile server infrastructure for your data center. This dual-port Enhanced Quad Small Form-Factor Pluggable (QSFP) half-height PCIe card is designed exclusively for Cisco UCS C-Series Rack Servers.

The card supports 40 Gigabit Ethernet and FCoE. It incorporates Cisco's next-generation converged network adapter (CNA) technology and offers a comprehensive feature set, providing investment protection for future software feature releases. The card can present more than 256 PCIe standards-compliant interfaces to the host, and these can be dynamically configured as either NICs or host bus adapters (HBAs). In addition, the VIC supports Cisco Data Center Virtual Machine Fabric Extender (VM-FEX) technology. This technology extends the Cisco UCS fabric interconnect ports to virtual machines, simplifying server virtualization deployment.

**Figure 8.** Cisco UCS VIC 1385 network adapter



The Cisco UCS VIC 1385 provides the following features and benefits:

- **Stateless and agile platform:** The personality of the card is determined dynamically at boot time using the service profile associated with the server. The number, type (NIC or HBA), identity (MAC address and World Wide Name [WWN]), failover policy, bandwidth, and quality-of-service (QoS) policies of the PCIe interfaces are all determined using the service profile. The capability to define, create, and use interfaces on demand provides a stateless and agile server infrastructure.
- **Network interface virtualization:** Each PCIe interface created on the VIC is associated with an interface on the Cisco UCS fabric interconnect, providing complete network separation for each virtual cable between a PCIe device on the VIC and the interface on the fabric interconnect.

### Cisco Nexus 9332PQ Switch

The Cisco Nexus® 9000 Series Switches include both modular and fixed-port switches that offer a flexible, agile, low-cost, application-centric infrastructure.

The Cisco Nexus 9300 platform consists of fixed-port switches designed for top-of-rack (ToR) and middle-of-row (MoR) deployment in data centers that support enterprise applications, service provider hosting, and cloud computing environments. They are Layer 2 and 3 nonblocking 10 and 40 Gigabit Ethernet switches with up to 2.56 Tbps of internal bandwidth.

The Cisco Nexus 9332PQ Switch (Figure 9) is a 1RU switch that supports 2.56 Tbps of bandwidth and over 720 million packets per second (mpps) across thirty-two 40-Gbps Enhanced QSFP+ ports.

**Figure 9.** Cisco Nexus 9332PQ Switch

All the Cisco Nexus 9300 platform switches use dual-core 2.5-GHz x86 CPUs with 64-GB SSD drives and 16 GB of memory for enhanced network performance.

With the Cisco Nexus 9000 Series, organizations can quickly and easily upgrade existing data centers to carry 40 Gigabit Ethernet to the aggregation layer or to the spine (in a leaf-and-spine configuration) through advanced and cost-effective optics that enable the use of existing 10 Gigabit Ethernet fiber (a pair of multimode fiber strands).

Cisco provides two modes of operation for the Cisco Nexus 9000 Series. Organizations can use Cisco NX-OS Software to deploy the Cisco Nexus 9000 Series in standard Cisco Nexus switch environments. Organizations also can use a hardware infrastructure that is ready to support Cisco Application Centric Infrastructure (Cisco ACI™) to take full advantage of an automated, policy-based, systems management approach.

### Red Hat Enterprise Linux 7.5

Red Hat Enterprise Linux (RHEL) is a high-performing operating system that has delivered outstanding value to IT environments for more than a decade. More than 90 percent of Fortune Global 500 companies use Red Hat products and solutions, including RHEL. As the world's most trusted IT platform, RHEL has been deployed in mission-critical applications at global stock exchanges, financial institutions, leading telecommunications companies, and animation studios. It also powers the websites of some of the most recognizable global retail brands.

RHEL offers these main features:

- It delivers high performance, reliability, and security.
- It is certified by the leading hardware and software vendors.
- It scales from workstations, to servers, to mainframes.
- It provides a consistent application environment across physical, virtual, and cloud deployments.

Designed to help organizations make a seamless transition to emerging data center models that include virtualization and cloud computing, RHEL includes support for major hardware architectures, hypervisors, and cloud providers, making deployment across different physical and virtual environments predictable and secure. Enhanced tools and new capabilities in the current release enable administrators to tailor the application environment to efficiently monitor and manage computing resources and security.

### Red Hat OpenShift Container Platform

OpenShift Container Platform is Red Hat's container application platform that brings together Docker and Kubernetes and provides an API to manage these services. OpenShift allows you to create and manage containers. Containers are standalone processes that run within their own environment, independent of operating system and the underlying infrastructure.

OpenShift helps organizations develop, deploy, and manage container-based applications. It provides a self-service platform for creating, modifying, and deploying applications on demand, thus enabling faster development and release lifecycles. OpenShift has

a microservices-based architecture of smaller, decoupled units that work together. It runs on top of a Kubernetes cluster, with data about the objects stored in etcd, a reliable clustered key-value store.

### Kubernetes infrastructure

Within OpenShift, Kubernetes manages containerized applications across a set of Docker runtime hosts and provides mechanisms for deployment, maintenance, and application-scaling. The Docker service packages, instantiates, and runs containerized applications.

A Kubernetes cluster consists of one or more master nodes and a set of other nodes. This solution design includes high-availability capabilities in the hardware as well as in the software stack. Kubernetes clusters are designed to run in high-availability mode, with three master nodes and two infrastructure nodes to help ensure that the cluster has no single point of failure.

### Red Hat OpenShift integrated container registry

OpenShift provides an integrated container registry called OpenShift Container Registry (OCR) that can automatically provision new image repositories on demand. This feature provides users with a built-in location for their application builds to push the resulting images. Whenever a new image is pushed to the OCR, the registry notifies OpenShift about the new image, passing along all the information about it, such as the name space, name, and image metadata. Different pieces of OpenShift respond to the presence of new images, creating new builds and deployments.

### Container-native storage from Red Hat

Container-native storage from Red Hat makes OpenShift a fully hyperconverged infrastructure in which storage containers reside together with computing containers. The storage plane is based on containerized Red Hat Gluster Storage, which controls storage devices on every storage server.

Heketi is part of the container-native storage architecture and controls all the nodes that are members of the storage cluster. Heketi also provides an API through which storage space for containers can easily be requested. Heketi provides an endpoint for the storage cluster, and the object that makes calls to its API from OpenShift clients is called a storage class. It is a Kubernetes and OpenShift object that describes the type of storage available for the cluster and can dynamically send storage requests when a persistent volume claim is generated.

Container-native storage for OpenShift is built on three main technologies:

- OpenShift provides platform-as-a-service (PaaS) infrastructure based on Kubernetes container management. Basic OpenShift architecture uses multiple master systems, with each system containing a set of nodes.
- Red Hat Gluster Storage provides containerized distributed storage using Red Hat Gluster Storage containers. Each Red Hat Gluster Storage volume is composed of a collection of bricks, and each brick consists of a node and an export directory.
- Heketi provides Red Hat Gluster Storage volume lifecycle management. It creates Red Hat Gluster Storage volumes dynamically and supports multiple Red Hat Gluster Storage clusters.

### Docker

OpenShift uses the Docker runtime engine for containers.

### Kubernetes

OpenShift is a complete container application platform that natively integrates technologies such as Docker and Kubernetes to provide powerful container cluster management and orchestration.

### Etcd

Etcd is a key-value store used in OpenShift clusters. The etcd data store provides complete cluster and endpoint states to the OpenShift API servers. The etcd data store furnishes information to API servers about node status, network configurations, secrets, etc.

### Open vSwitch

Open vSwitch is an open-source implementation of a distributed virtual multilayer switch. It is designed to enable effective network automation through programmatic extensions, while supporting standard management interfaces and protocols such as IEEE 802.1ag, Cisco Switched Port Analyzer (SPAN), Link-Access Control Protocol (LACP), and Cisco NetFlow. Open vSwitch provides software-defined networking (SDN)–specific functions in the OpenShift environment.

### HAProxy

HAProxy is open-source software that provides a high-availability load balancer and proxy server for TCP- and HTTP-based applications that spreads requests across multiple servers. In this solution, HAProxy is deployed in an infrastructure node and provides routing and load-balancing functions for OpenShift applications. Another instance of HAProxy acts as an ingress router for all applications deployed in the OpenShift cluster.

### Red Hat Ansible Automation

Red Hat Ansible Automation is a powerful IT automation tool. It can provision numerous types of resources and deploy applications. It can configure and manage devices and operating system components. Because of Ansible's simplicity, extensibility, and portability, this OpenShift solution is based largely on Ansible playbooks.

Ansible is used mainly for installing and managing the OpenShift deployment.

## NVIDIA deep-learning neural framework and tools

The NVIDIA deep-learning SDK accelerates widely used deep-learning frameworks such as NVCaffe, Caffe2, Microsoft Cognitive Toolkit, MXNet, TensorFlow, PyTorch, Torch, and TensorRT. NVIDIA GPU Cloud, or NGC, provides containerized versions of these frameworks optimized for the Cisco AI server platform. These frameworks, including all necessary dependencies, are prebuilt, tested, and ready to run. For users who need more flexibility to build custom deep-learning solutions, each framework container image also includes the framework source code to enable custom modifications and enhancements, along with the complete software development stack.

Most deep-learning frameworks have begun to merge support for half-precision training techniques that exploit Tensor core calculations in Volta. Some frameworks include support for FP16 storage and Tensor core math. To achieve increased training throughput, you can train a model using Tensor core math and FP16 mode on some frameworks.

### ImageNet

Deep learning attempts to model data through multiple processing layers containing nonlinearities. It has proven to be efficient at classifying images, as shown by the impressive results of deep neural networks in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition, for example. However, training these models requires large data sets and is time consuming.

ImageNet, developed at Stanford and MIT, is a large visual database designed for use in visual object recognition software research. It is the data set most commonly used by major training models (ResNet, Inception, VGG16, etc.) for performance benchmarking. It provides a common foundation for comparing architectures. Networks trained on ImageNet can be starting points for other computer visioning tasks.

### TensorFlow

The TensorFlow deep-learning framework can be used to test popular convolutional neural network (CNN) training models such as ResNet, Inception, VGG, AlexNet, and GoogleNet. TensorFlow is a software library, developed by the Google Brain Team within the Google Machine Learning Intelligence research organization.

The main features of TensorFlow include the following:

- Capability to define, optimize, and efficiently calculate mathematical expressions involving multidimensional arrays (tensors)
- Programming support for deep neural networks and machine-learning techniques
- Transparent use of GPU computing, automating management and optimization of the same memory and the data used; you can write the same code and run it on either CPUs or GPUs
- High scalability of computation across machines and huge data sets

The TensorFlow CNN benchmarks contain implementations of several popular convolutional models and can be run on a single machine or in distributed mode across multiple hosts. TensorFlow is used in the tests reported in this document for the Cisco UCS and OpenShift solution.

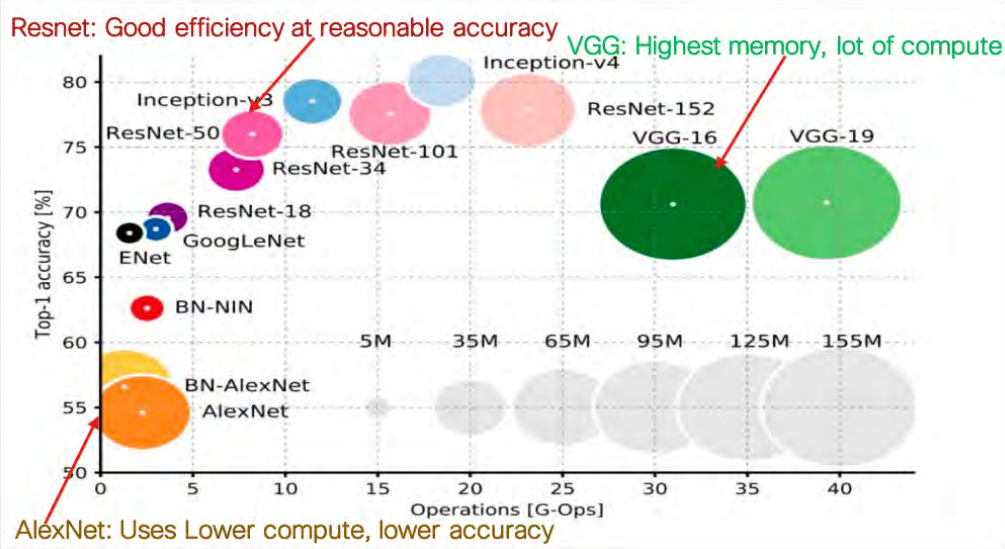
### Choice of model

The first step is to choose the model to use. ILSVRC requires models to classify 1000 categories of images, and some suggested models cannot provide this level of performance. However, if you choose to classify 10 to 100 categories of images, the models can fit the architecture discussed here.

CNN models have evolved, and some of them have complicated architecture. If you want to modify certain parts of an entire layer, or if you want to troubleshoot to find the part that is the bottleneck in a problem situation, you must understand how the models works behind the scenes.

CNN models have evolved to support a range of trade-offs between time to accuracy and number of operations. For example, ResNet50 requires relatively few operations to achieve relatively high accuracy. VGG models require the greatest amount of memory and have high computing requirements with moderate accuracy for a single forward pass. Figure.10, shows the evolution of the training models.

**Figure 10.** Operations versus accuracy among the training models



## Solution design

This section provides an overview of the hardware and software components used in this solution, as well as the design factors you should consider to make the system work as a single, highly available solution.

### Hardware and software revisions

Note: The following hardware and software revisions have been validated in Red Hat OpenShift Container Platform 3.11.

Table 1 lists the firmware versions validated in this OpenShift solution.

**Table 1.** Hardware versions

| Hardware | Firmware Version |
|---|---|
| **Cisco UCS Manager** | Release 4.0.(2b)A |
| **Cisco UCS 6332 16-UP Fabric Interconnect** | Release 4.0.(2b)A |
| **Cisco UCS C480 ML M5 Rack Server** | Release 4.0.(2b)C |
| **Cisco UCS C240 M5 Rack Server (SFF)** | Release 4.0.(2b)C |
| **Cisco UCS C220 M5 Rack Server (SFF)** | Release 4.0.(2b)C |

Note: For information about the OS version and system type, see Cisco Hardware and Software Compatibility.

Table 2 lists the software versions validated in this OpenShift solution.

**Table 2.** Software versions

| Software | Version |
|---|---|
| **Red Hat Enterprise Linux** | Release 7.6 |
| **Red Hat OpenShift Container Platform** | Release 3.11.69-1 |
| **Kubernetes** | Release 1.11 |
| **Docker** | Release 1.13.1 |
| **Red Hat Ansible Engine** | Release 2.7.4 |
| **Etcd** | Release 3.2.22 |
| **Open vSwitch** | Release 2.9.0 |
| **Red Hat Gluster Storage** | Release 3.3.0 |
| **GlusterFS** | Release 3.12.2 |
| **NVIDIA driver** | 410.79 |
| **CUDA** | 10.0 |
| **NVIDIA-Container-runtime-hook** | 1.4.0-2 |

## Solution components

The solution was validated using the components listed in Table 3.

**Table 3.** Solution components

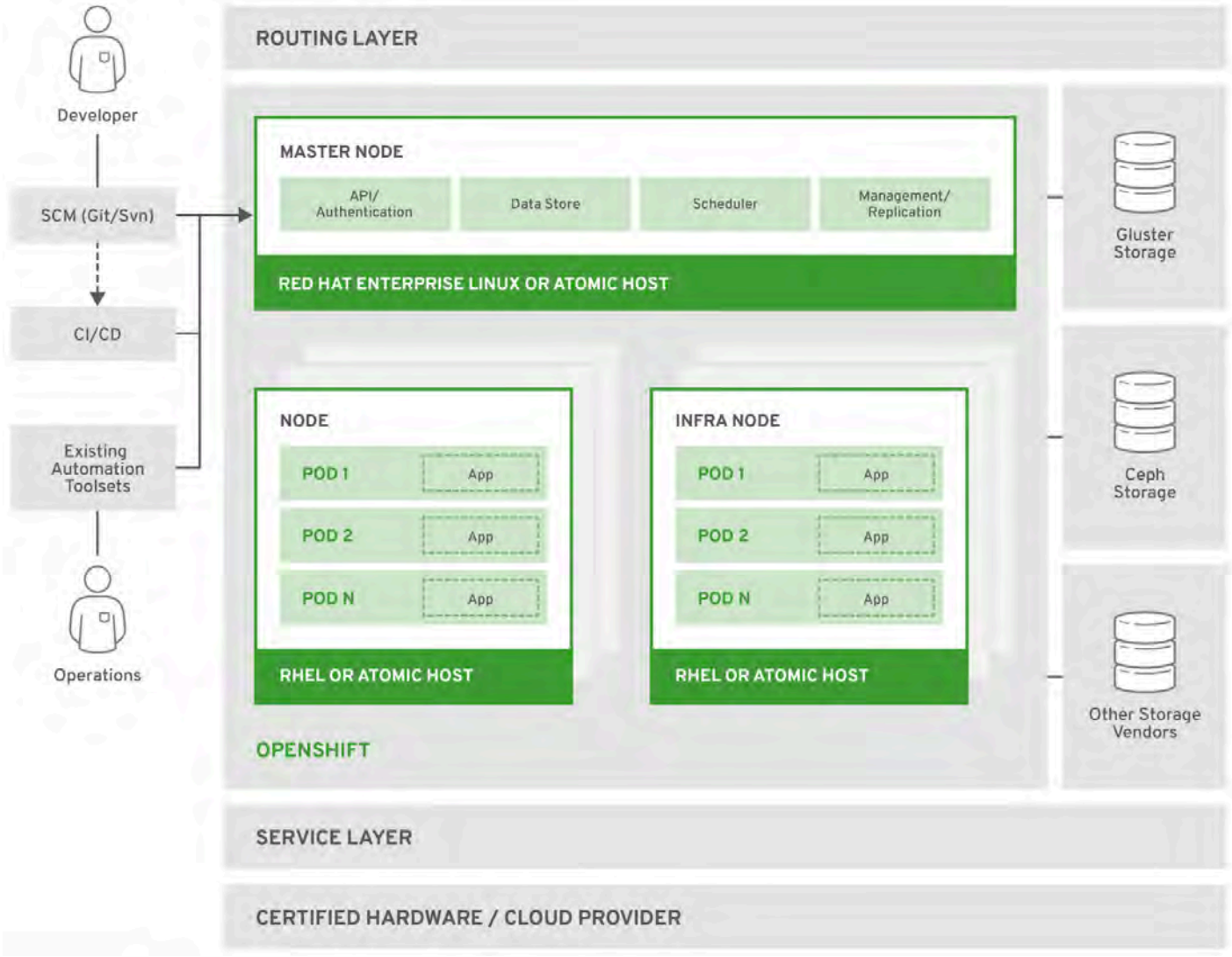
| Component | Model | Quantity |
|---|---|---|
| **Bastion node** | Cisco UCS C220 M5SX | 1 |
| **Master nodes** | Cisco UCS C220 M5SX | 3 |
| **Infrastructure nodes** | Cisco UCS C220 M5SX | 2 |
| **Computing nodes (applications)** | Cisco UCS C240 M5SX | 2 |
| **Computing node (application)** | Cisco UCS C480 ML M5 | 1 |
| **Storage nodes** | Cisco UCS C240 M5SX | 3 |
| **Fabric interconnects** | Cisco UCS 6332 16-UP | 2 |
| **Cisco Nexus switches** | Cisco Nexus 9332PQ | 2 |

## Architecture overview

The Red Hat OpenShift Container Platform is managed by the Kubernetes container orchestrator, which manages containerized applications across a cluster of systems running the Docker container runtime. The physical configuration of OpenShift is based on the Kubernetes cluster architecture. OpenShift is a layered system designed to expose underlying Docker-formatted container images and Kubernetes concepts as accurately as possible, with a focus on easy composition of applications by a developer: for example, install Ruby, push code, and add MySQL. The concept of an application as a separate object is replaced with the more flexible concept of a composition of services, allowing two web containers to reuse a database or expose a database directly to the edge of the network.

Figure 11 illustrates the architecture.

**Figure 11.** Architectural overview




This OpenShift reference architecture contains five types of nodes: bastion, master, infrastructure, storage, and application nodes. The solution discussed here uses Gluster Storage to run containerized GlusterFS services, which configure persistent volumes for application containers that require data persistence.

Table 4 summarizes the solution architecture.

**Table 4.** Solution architecture

| Component | Model | Quantity | Description |
|---|---|---|---|
| **Application nodes** | Cisco UCS C480 ML M5 | 1 | • CPU: 2 x Intel Xeon processor 8180 CPUs at 2.1 GHz, with 16 cores each<br>• Memory: 24 x 32-GB 2666 DIMMs, for a total of 768 GB<br>• Local storage: 2 x 300-GB SAS 10,000-rpm SFF drives<br>• Network card: 1 x VIC 1385 PCIe cards<br>• RAID controller: Cisco MegaRAID 12-Gbps SAS controller<br>• GPU: 8 x 32-GB NVIDIA Tesla V100 SMX2 card |
| **Application nodes** | Cisco UCS C240 M5S | 2 | • CPU: 2 x Intel Xeon processor 8180 CPUs at 2.1 GHz, with 16 cores each<br>• Memory: 12 x 32-GB 2666 DIMMs, for a total of 384 GB<br>• Local storage: 2 x 300-GB SAS 10,000-rpm SFF drives<br>• Network card: 1 x VIC 1385 PCIe card<br>• RAID controller: Cisco MegaRAID 12-Gbps SAS controller<br>• GPU: 2 x 32-GB NVIDIA Tesla V100 PCIe card |
| **Master nodes** | Cisco UCS C220 M5SX | 3 | • CPU: 2 x Intel Xeon processor 4114 CPUs at 2.2 GHz, with 10 cores each<br>• Memory: 12 x 16-GB 2400 DIMMs, for a total of 192 GB<br>• Local storage: 2 x 300-GB SAS 10,000-rpm SFF drives<br>• Network card: 1 x VIC 1340 card<br>• RAID controller: Cisco MegaRAID 12-Gbps SAS controller |
| **Infrastructure nodes** | Cisco UCS C220 M5SX | 2 | • CPU: 2 x Intel Xeon processor 4114 at 2.2 GHz, with 10 cores each<br>• Memory: 12 x 16-GB 2400 DIMMs, for a total of 192 GB<br>• Local storage: 2 x 300-GB SAS 10,000-rpm SFF drives<br>• Network card: 1 x VIC 1340 card<br>• RAID controller: Cisco MegaRAID 12-Gbps SAS controller |
| **Bastion node** | Cisco UCS C220 M5SX | 1 | • CPU: 2 x Intel Xeon processor 4114 CPUs at 2.2 GHz, with 10 cores each<br>• Memory: 12 x 16-GB 2400 DIMMs, for a total of 192 GB<br>• Local storage: 2 x 300-GB SAS 10,000-rpm SFF drives<br>• Network card: 1 x VIC 1387 mLOM<br>• RAID controller: Cisco MegaRAID 12-Gbps SAS controlle |
| **Storage nodes** | Cisco UCS C240 M5SX | 3 | • CPU: 2 x Intel Xeon 6130 CPUs at 2.1 GHz, with 16 cores each<br>• Memory: 12 x 16-GB 2666 DIMMs, for a total of 192 GB<br>• SSDs: 2 x 300-GB SAS 15,000-rpm SFF drives<br>• SSDs: 20 x 3.8-TB SATA drives<br>• Network card: 1 x VIC 1387 mLOM<br>• RAID controller: Cisco MegaRAID 12-Gbps SAS controller |
| **Fabric interconnects** | Cisco UCS 6332 16-UP | 2 | |
| **Switches** | Cisco Nexus 9332PQ | 2 | |

Table 5 lists the functions and roles for each class of node in this solution for OpenShift.

**Table 5.** OpenShift cluster node types and roles

| Node | Role |
|---|---|
| **Bastion node** | • System deployment and management operations<br>• Running Ansible playbooks<br>• Can also be configured as an IP router that routes traffic across all the nodes through the control network |
| **Master nodes** | • Kubernetes services<br>• Etcd data store<br>• Controller manager and scheduler<br>• API services |
| **Infrastructure nodes** | • Container registry<br>• Heketi<br>• High-availability proxy router |
| **Application nodes** | • Application containers pods<br>• Docker runtime |
| **Storage nodes** | • Red Hat Gluster Storage<br>• Container-native storage services<br>• Labeled as computing nodes, so workload scheduling is enabled by default |

### Bastion node

The bastion node is a dedicated node that serves as the main deployment and management server for the OpenShift cluster. It is used as the logon node for cluster administrators to perform system deployment and management operations, such as running Ansible OpenShift deployment playbooks and performing scale-out operations. The bastion node also runs Domain Name System (DNS) services for the OpenShift cluster nodes. The bastion node runs RHEL 7.5.

### OpenShift master nodes

The OpenShift master node is a server that performs control functions for the whole cluster environment. It is responsible for the creation, scheduling, and management of all objects specific to OpenShift. It includes API, controller manager, and scheduler capabilities in one OpenShift binary. It is also a common practice to install an etcd key-value store on OpenShift masters to achieve a low-latency link between etcd and OpenShift master nodes. You should run both OpenShift masters and etcd in highly available environments. You can achieve this by running multiple OpenShift masters in conjunction with an external active-passive load balancer and the clustering functions of etcd. OpenShift master nodes run RHEL Atomic Host 7.5.

### OpenShift infrastructure nodes

The OpenShift infrastructure node runs infrastructure-specific services: the Docker registry, the HAProxy router, and Heketi. The Docker registry stores application images in the form of containers. The HAProxy router provides routing functions for OpenShift applications. It currently supports HTTP(S) traffic and TLS-enabled traffic through Server Name Indication (SNI). Heketi provides a management API for configuring GlusterFS persistent storage. Additional applications and services can be deployed on OpenShift infrastructure nodes. OpenShift infrastructure nodes run RHEL Atomic Host 7.5.

### OpenShift application nodes

The OpenShift application nodes run containerized applications created and deployed by developers. An OpenShift application node contains the OpenShift node components combined into a single binary, which can be used by OpenShift masters to schedule and control containers. OpenShift application nodes run RHEL Atomic Host 7.5.

### OpenShift storage nodes

The OpenShift storage nodes run containerized GlusterFS services, which configure persistent volumes for application containers that require data persistence. Persistent volumes can be created manually by a cluster administrator or automatically by storage-class objects. OpenShift storage nodes can also run containerized applications. OpenShift storage nodes run RHEL Atomic Host 7.5.

### HAProxy load balancer

The reference architecture uses the HAProxy load balancer. However, an existing on-premises load balancer can also be used. HAProxy is the entry point for many OpenShift components. The OpenShift console is accessible through the master nodes, which are spread across multiple instances to provide load balancing as well as high availability.

Application traffic passes through the OpenShift router on its way to the container processes. The OpenShift router is a reverse proxy service container that multiplexes traffic to multiple containers that make up a scaled application running within OpenShift. The load balancer used by the infrastructure nodes serves as the public view for OpenShift applications.

The destination for master and application traffic must be set in the load balancer configuration after each instance is created and the floating IP address is assigned and before installation. A single HAProxy load balancer can forward both sets of traffic to different destinations.

In this solution, HAProxy provides routing and load-balancing functions for OpenShift applications. Another instance of HAProxy acts as an ingress router for all applications deployed in the OpenShift cluster. Both instances are replicated to every infrastructure node and managed by additional components (Keepalived and OpenShift services) to provide redundancy and high availability.

### Keepalived

Keepalived is routing software that provides simple and robust facilities for load balancing and high availability on Linux-based infrastructure. In this solution, Keepalived is used to provide virtual IP management for HAProxy instances to help ensure that the OpenShift cluster is highly available. Keepalived is deployed on infrastructure nodes, which also act as HAProxy load balancers. If one infrastructure node fails, Keepalived automatically moves the virtual IP addresses to the second node, which acts as a backup. With Keepalived, OpenShift infrastructure becomes highly available and resistant to failures.

### OpenShift networking

OpenShift supports the Kubernetes Container Network Interface (CNI) as the interface between OpenShift and Kubernetes. SDN plug-ins provide a powerful and flexible way to match network capabilities to networking needs.

This solution design relies on Kubernetes to help ensure that pods can network with each other and to obtain an IP address from an internal network for each pod. This approach helps ensure that all containers in the pod behave as if they were on the same host. Giving each pod its own IP address allows pods to be treated like physical hosts or virtual machines for the purposes of port allocation, networking, naming, service discovery, load balancing, application configuration, and migration. Creation of links between pods is not necessary, nor is it recommended because pods communicate with each other directly using the IP address. For interaction with the pods, services are created, and communication patterns between the pods can be defined.

### OpenShift software-defined networking

OpenShift deploys an SDN approach for connecting pods in an OpenShift cluster. The OpenShift SDN connects all pods across all node hosts, providing a unified cluster network. OpenShift SDN is installed and configured by default as part of the Ansible-based installation procedure.

OpenShift uses SDN to provide a unified cluster network that enables communication between pods across the OpenShift cluster. This pod network is established and maintained by the OpenShift SDN, which configures an overlay network using Open vSwitch (OVS).

### Network isolation

The OVS multitenant plug-in provides network isolation. When a packet exits a pod assigned to a nondefault project, the OVS bridge br0 tags that packet with the project's assigned virtual network identifier (VNID). If the packet is directed to another IP address in the node cluster subnet, the OVS bridge allows the packet to be delivered to the destination pod only if the VNIDs match.

### OpenShift DNS

OpenShift has a built-in DNS tool so that services can be reached through the service's DNS as well as by the service's IP address and port. OpenShift supports split DNS through the use of SkyDNS on the master node, with SkyDNS responding to DNS queries for services. The master node listens to port 53 by default.

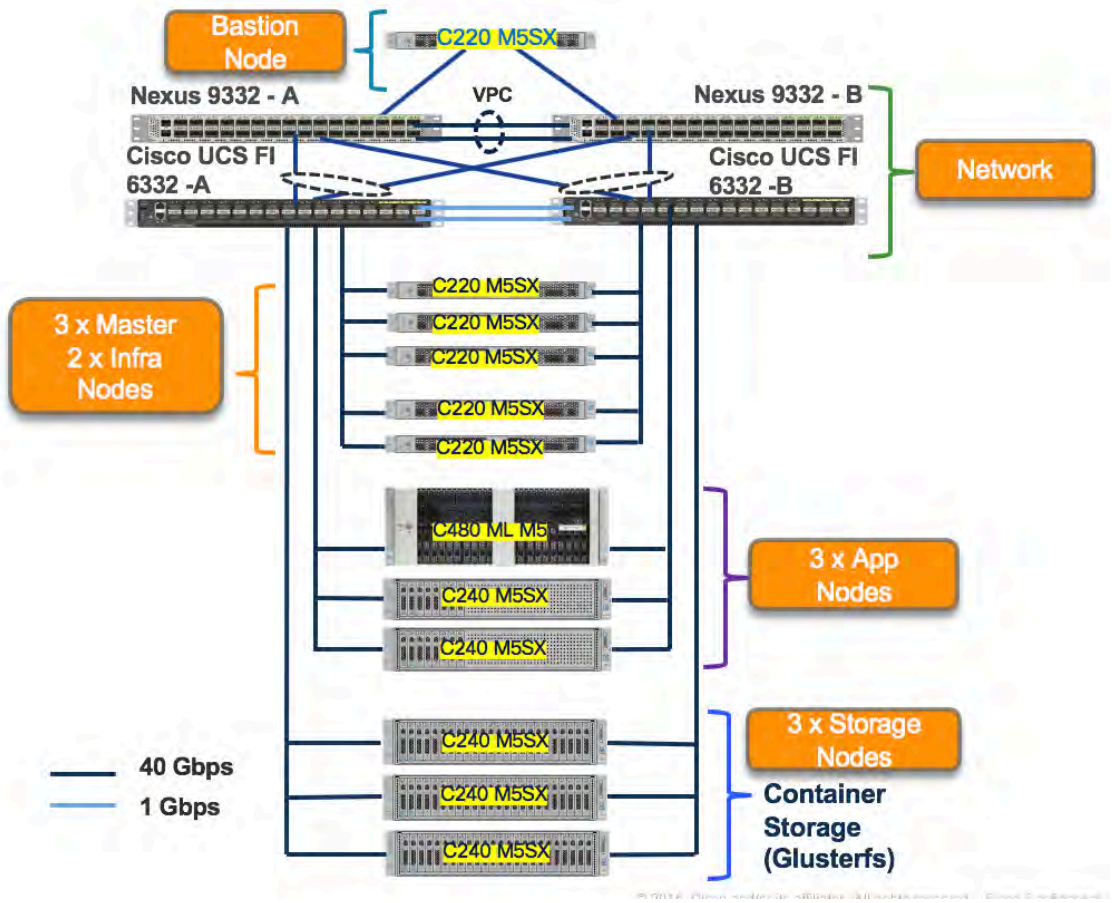
DNS services are required for the following use cases:

- DNS services are required when multiple services such as front-end and back-end services are running on multiple pods, and front-end pods need to communicate with back-end pods.
- When services are deleted and re-created again, you may assign a new set of IP addresses but want service names to remain unchanged to enable communication between service types, without having to change the code. By using DNS services, interservice communication can take place using names instead of IP addresses.

### Physical topology

Figure 12 shows the physical architecture used in this reference design.

**Figure 12.** Red Hat OpenShift Container Platform architecture: Physical topology

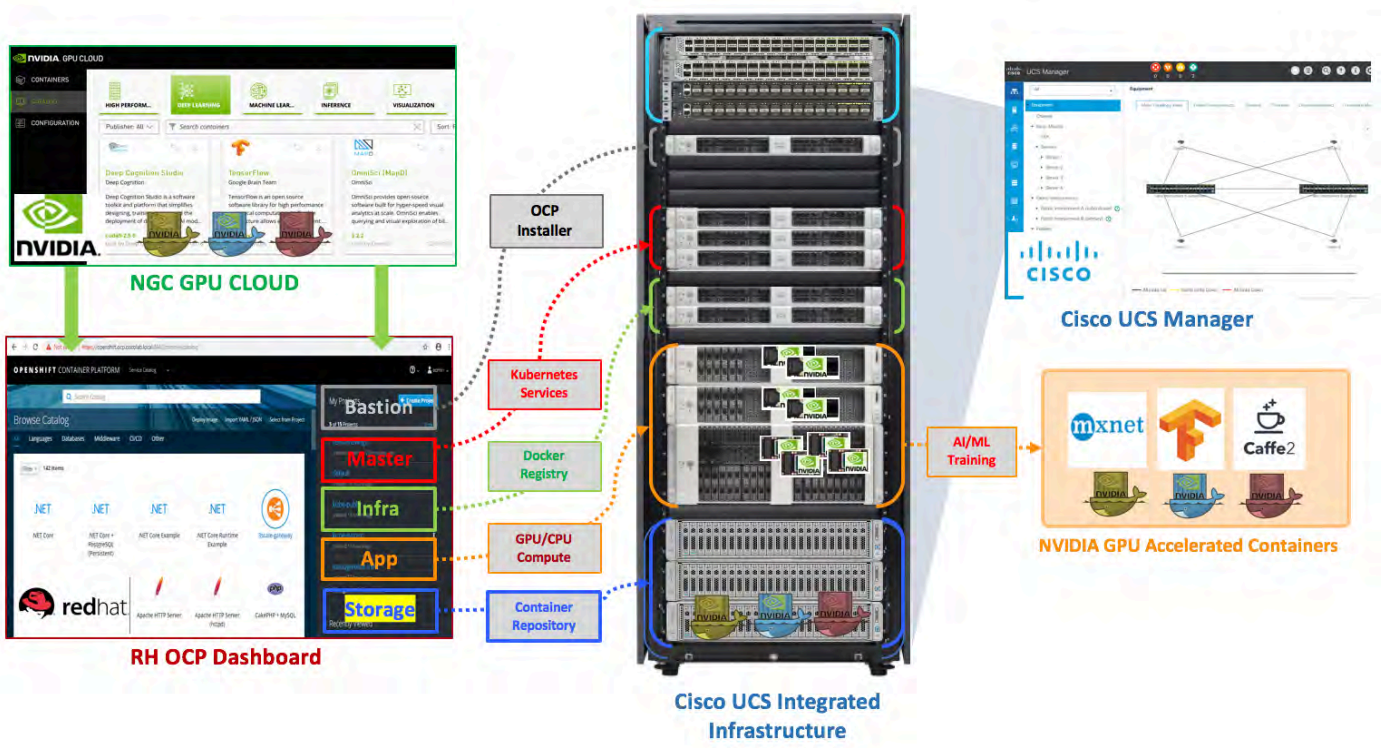



Configure all the Cisco UCS servers to boot from the local disk by configuring storage profiles in Cisco UCS Manager with RAID 1 mirrored.

## Logical topology

Figure 13 shows the logical topology for the OpenShift platform managed by Cisco UCS infrastructure.

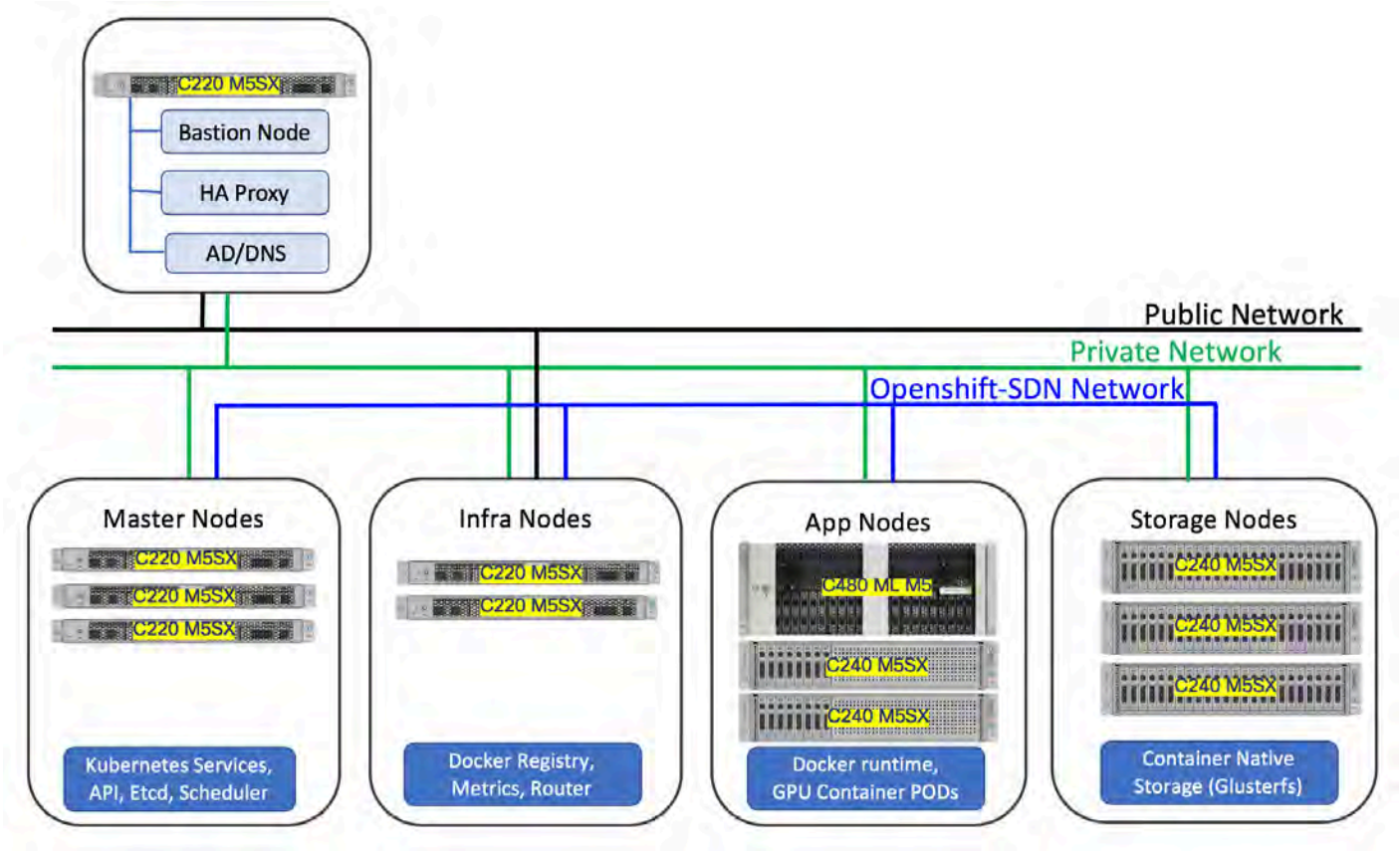
**Figure 13.** Logical topology



### Network layout

Figure 14 shows the network layout.

**Figure 14.** Network layout




**Notes:**

- **Private network:** This network is common to all nodes. This is an internal network and is used mainly by the bastion node to perform Secure Shell (SSH) access for the Ansible playbook. Hence, the bastion node acts as a jump host for SSH. Outbound Network Address Translation (NAT) is configured so that nodes can access the Red Hat content delivery network (CDN) to enable the required repositories.
- **Bastion node:** The bastion node is part of the public subnet. It can also be configured as an IP router that routes traffic across all the nodes through the control network. In this case, the default gateway for all nodes will be the bastion node's private IP address. This scenario would limit outside access for all nodes if the bastion node were powered down.
- **OpenShift SDN:** OpenShift uses an SDN approach to provide a unified cluster network that enables communication between pods across the OpenShift cluster. This pod network is established and maintained by the OpenShift SDN, which configures an overlay network using OVS.
- **Public network:** This network provides Internet access.

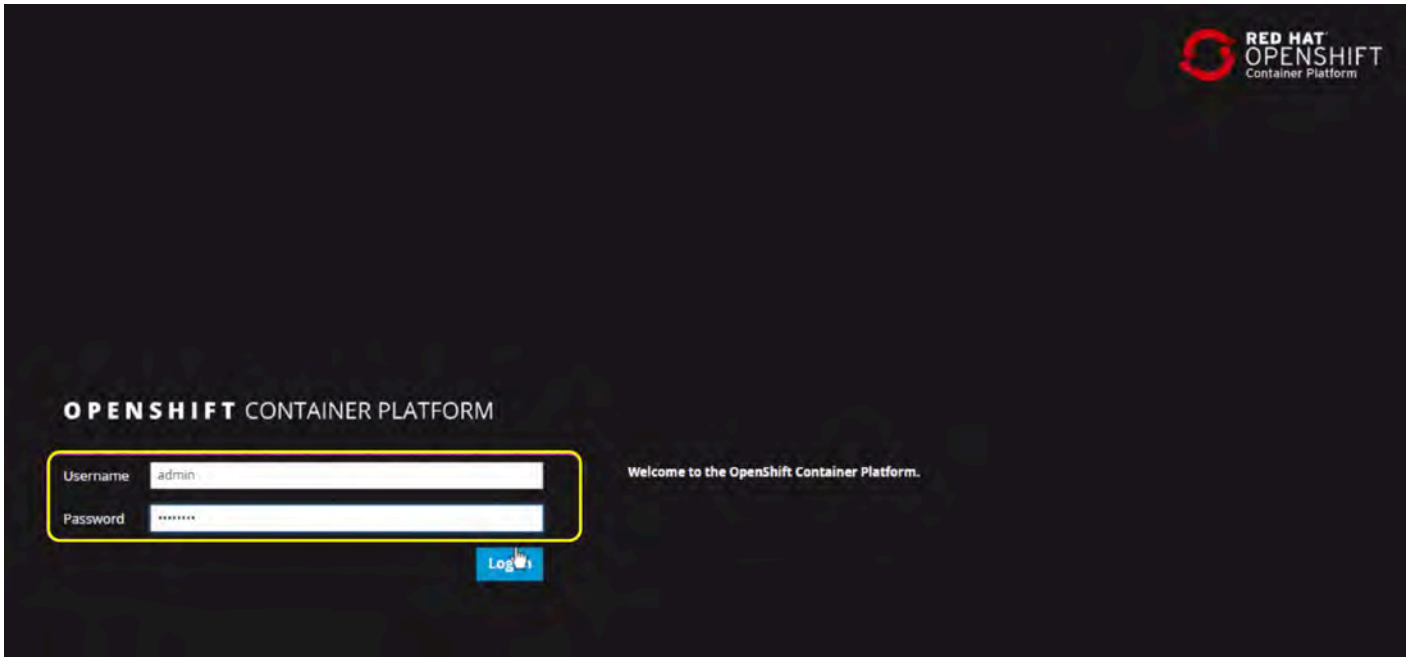
### Solution Validation

After the OCP installation completed successfully, we run the following steps to validate the deployment:

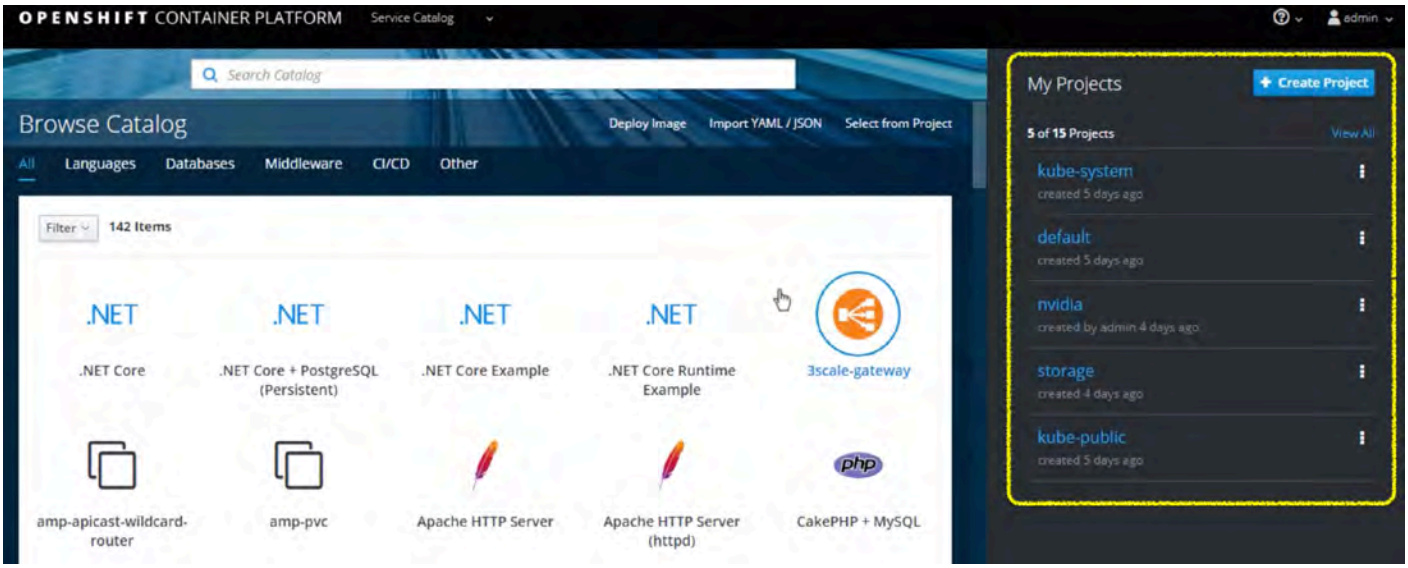
1. Verify all the OCP nodes are registered and reporting `Ready` status.

```
[root@ocp-master-node1 ~]# oc get nodes
NAME                          STATUS    ROLES     AGE       VERSION
ocp-master-node1.ocp.ciscolab.local        Ready     master    1d        v1.11.0+d4cacc0
ocp-master-node2.ocp.ciscolab.local        Ready     master    1d        v1.11.0+d4cacc0
ocp-master-node3.ocp.ciscolab.local        Ready     master    4d        v1.11.0+d4cacc0
ocp-infra-node1.ocp.ciscolab.local        Ready     infra     4d        v1.11.0+d4cacc0
ocp-infra-node2.ocp.ciscolab.local        Ready     infra     4d        v1.11.0+d4cacc0
ocp-app-node1.ocp.ciscolab.local    Ready     compute   4d        v1.11.0+d4cacc0
ocp-app-node2.ocp.ciscolab.local    Ready     compute   4d        v1.11.0+d4cacc0
ocp-app-node3.ocp.ciscolab.local    Ready     compute   4d        v1.11.0+d4cacc0
ocp-storage-node1.ocp.ciscolab.local   Ready     compute   4d        v1.11.0+d4cacc0
ocp-storage-node2.ocp.ciscolab.local   Ready     compute   4d        v1.11.0+d4cacc0
ocp-storage-node3.ocp.ciscolab.local   Ready     compute   4d        v1.11.0+d4cacc0
```

2. Launch the web browser and navigate to openshift web console. https://ocp.ciscolab.local:8443 and we make sure the authentication is successful to log into Redhat Openshift Container platform.

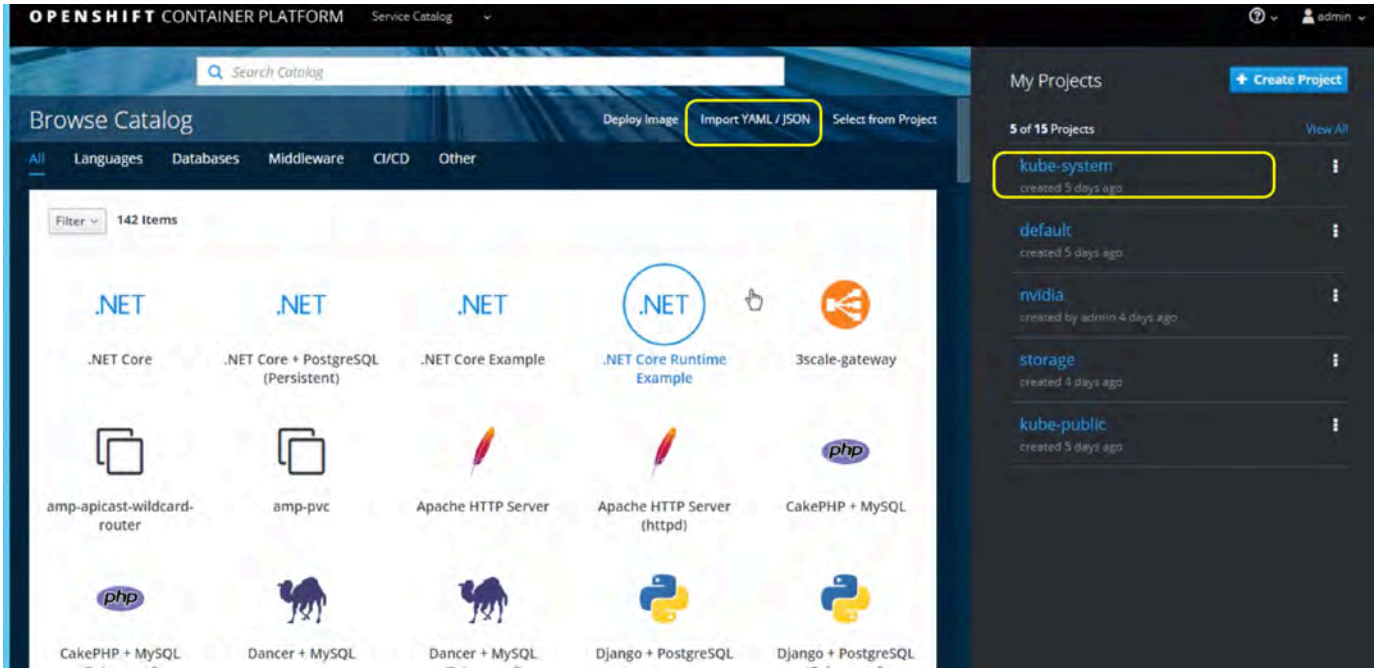


3. Verify OpenShift Container platform web console is listing all the openshift projects.



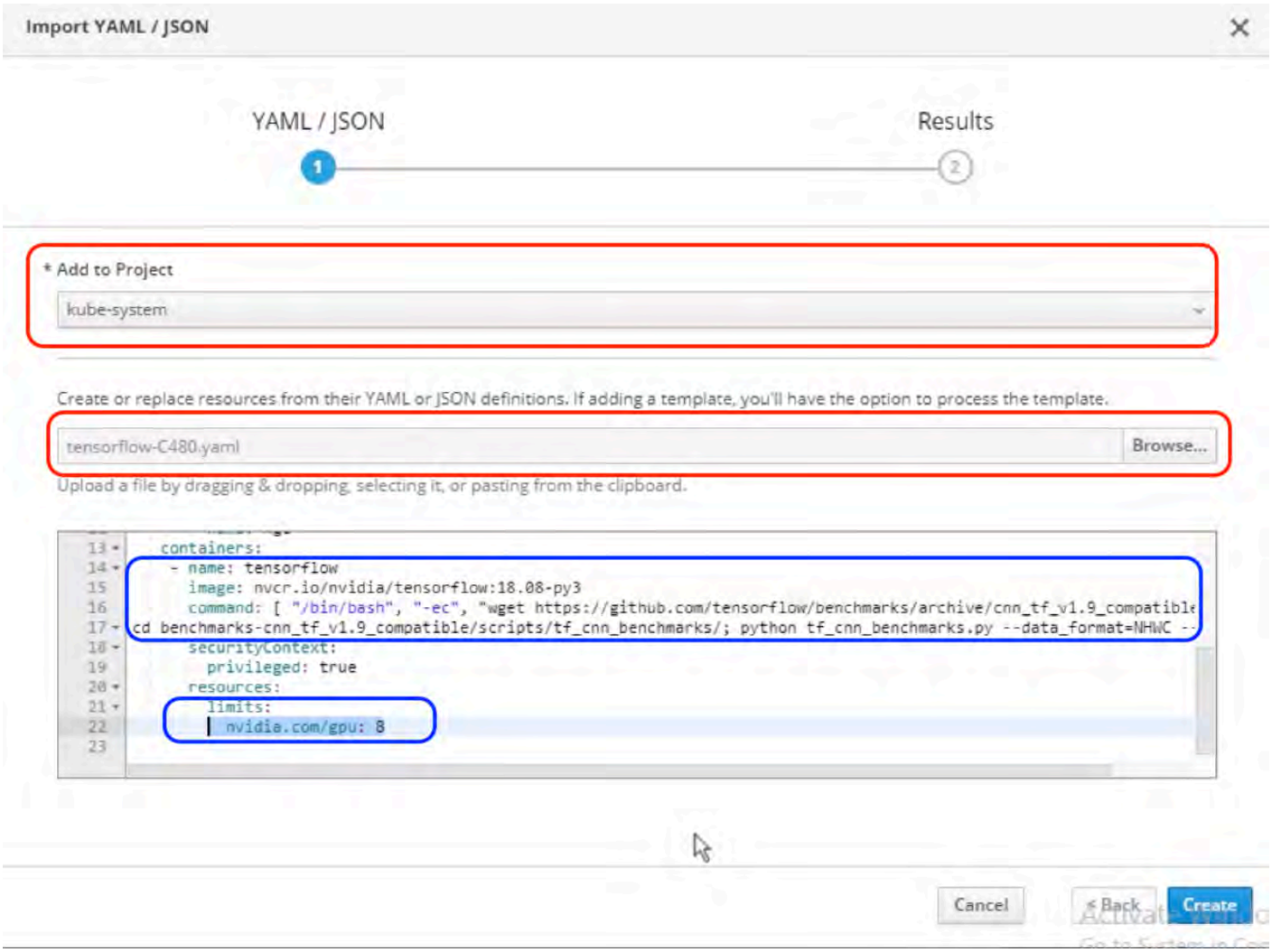

Below section summarizes AI/ML test scenarios during solution validation exercise.

1. To Spawn NGC GPU enabled containers using Openshift container webconsole. Select the appropriate project “Kube-system” which is already configured with GPU enabled Compute/App nodes. Then Click on “Import YAML/JSON” as shown in the picture.

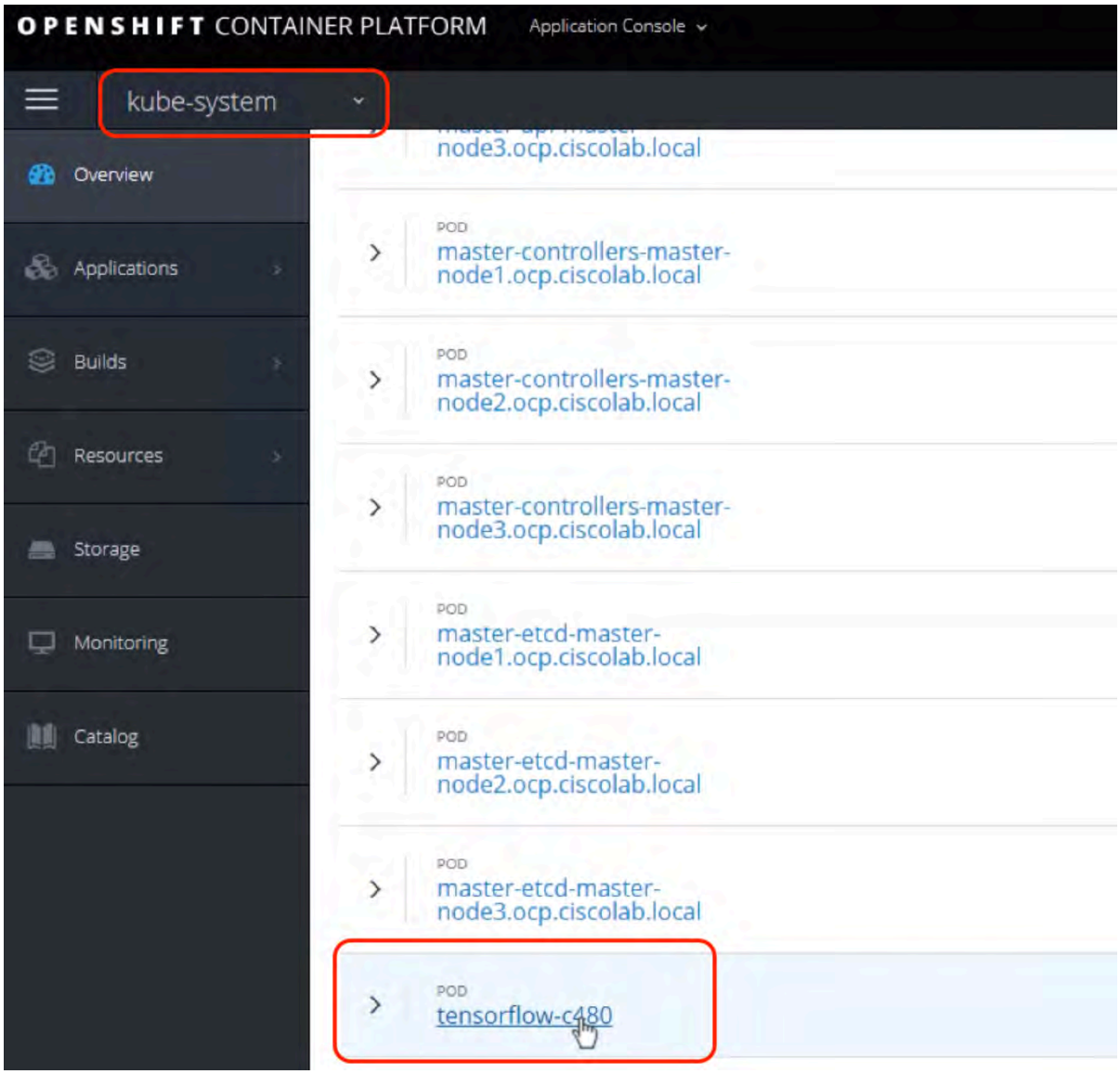


2. Import the Yaml or JSON resource file created for running Tensorflow benchmark (for DL model “Resnet50”) on the App node (C480 ML M5) by testing all 8 x NVIDIA V100 GPUs.

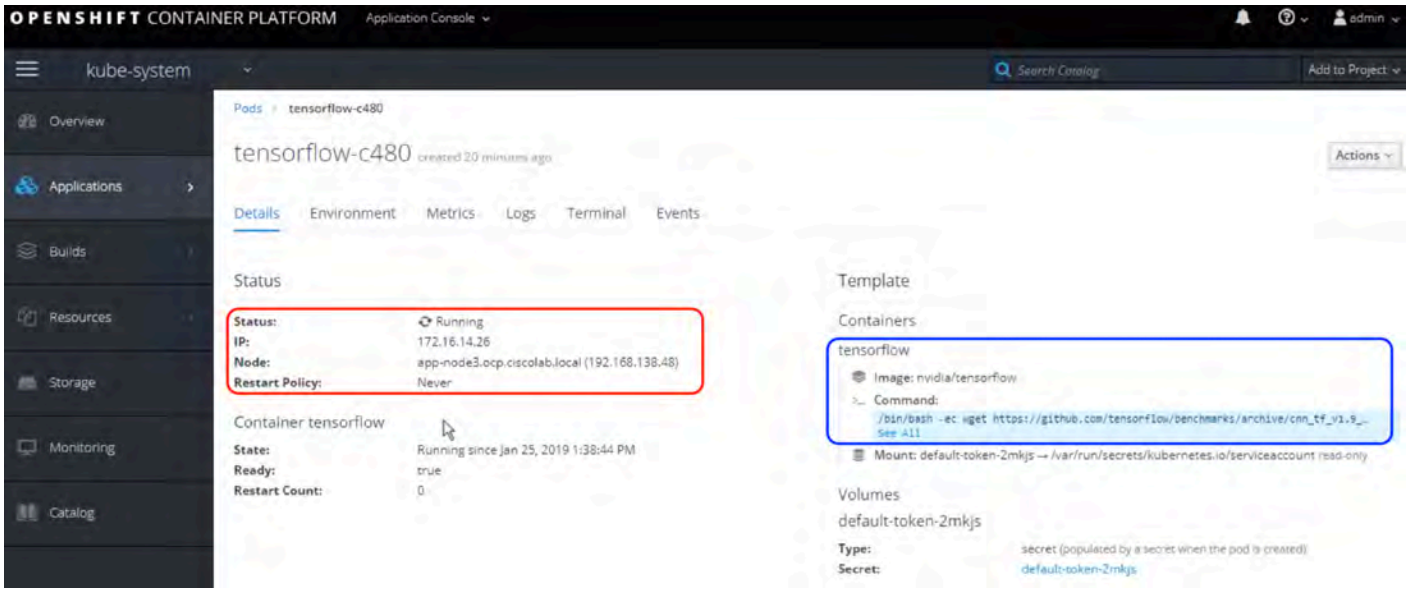
**Note:** Refer Appendix A. for creating .yaml file for running Tensorflow on C480 ML M5 server.



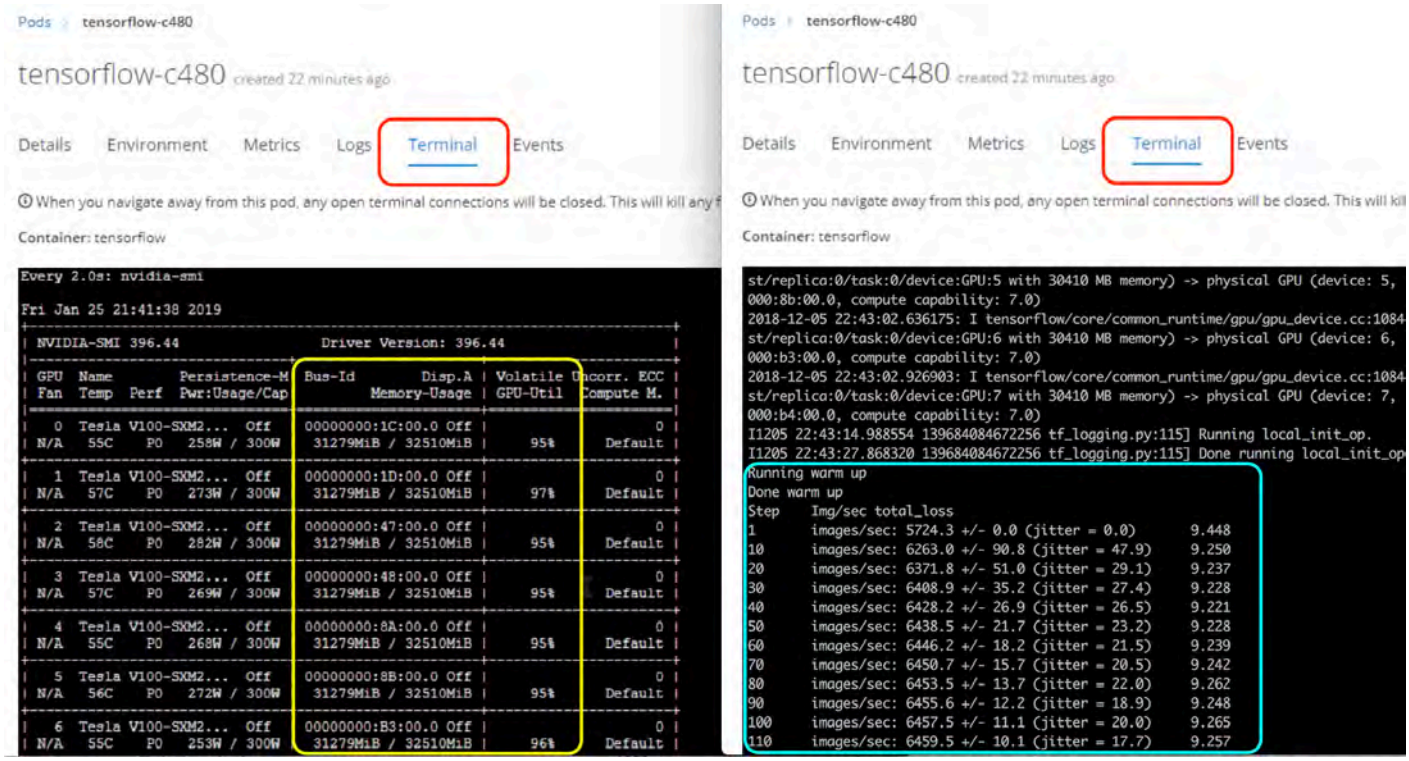

3. Now the new POD “tensorflow-c480” has been created under projects “kube-system”.




4. After successful creation of container pod “tensorflow-c480”. Now you can see the newly created container is running on App-node3 which is C480 ML M5 server. Then, the container has the command to automatically start TensorFlow benchmark testing for DL model “resnet50”.

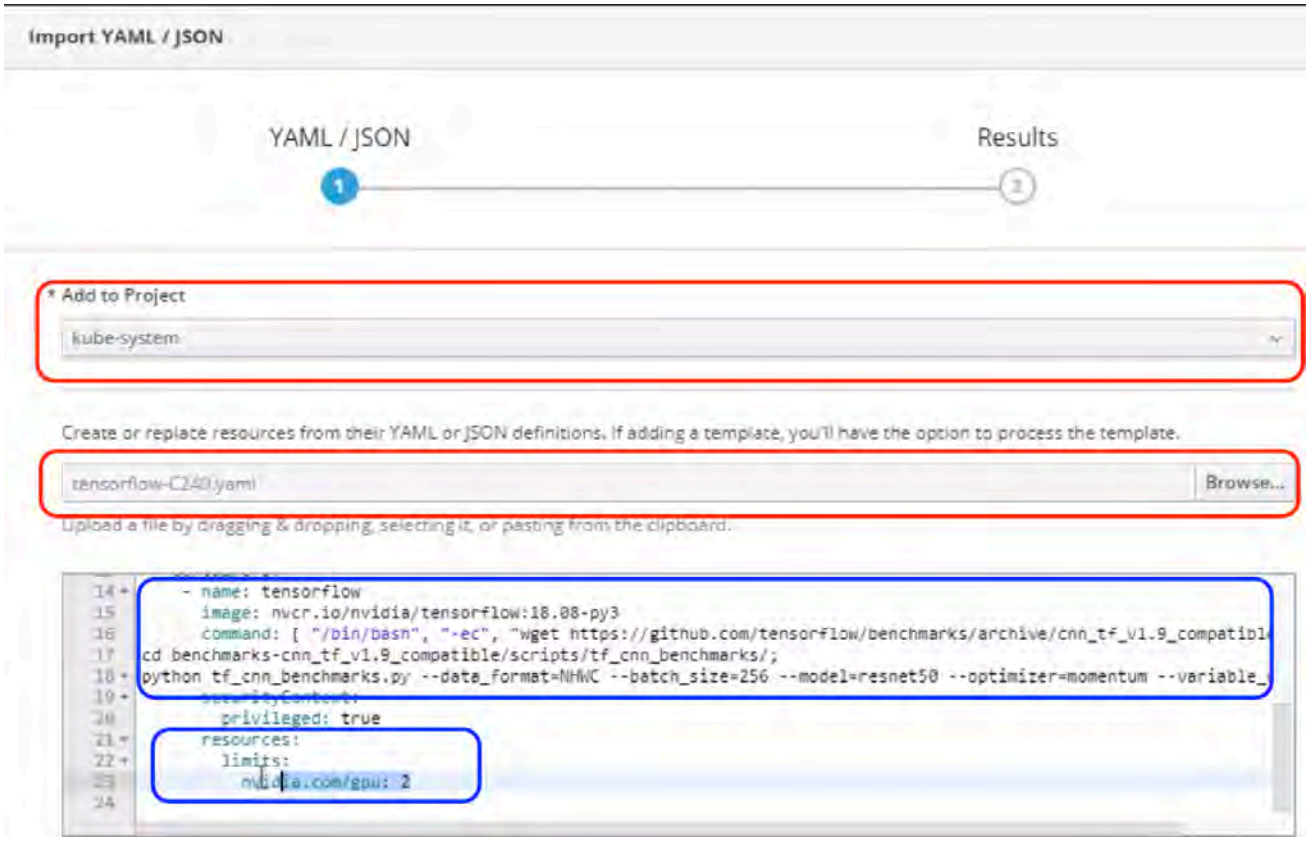



5. Click on Terminal tab for “tensorflow-c480”, which will show the “images/sec” for DL model resnet50. Launch the “TensorFlow-C480” from another browser and click on Terminal tab to monitor the GPU utilization of all 8 x NVIDIA V100 GPUs during the testing.

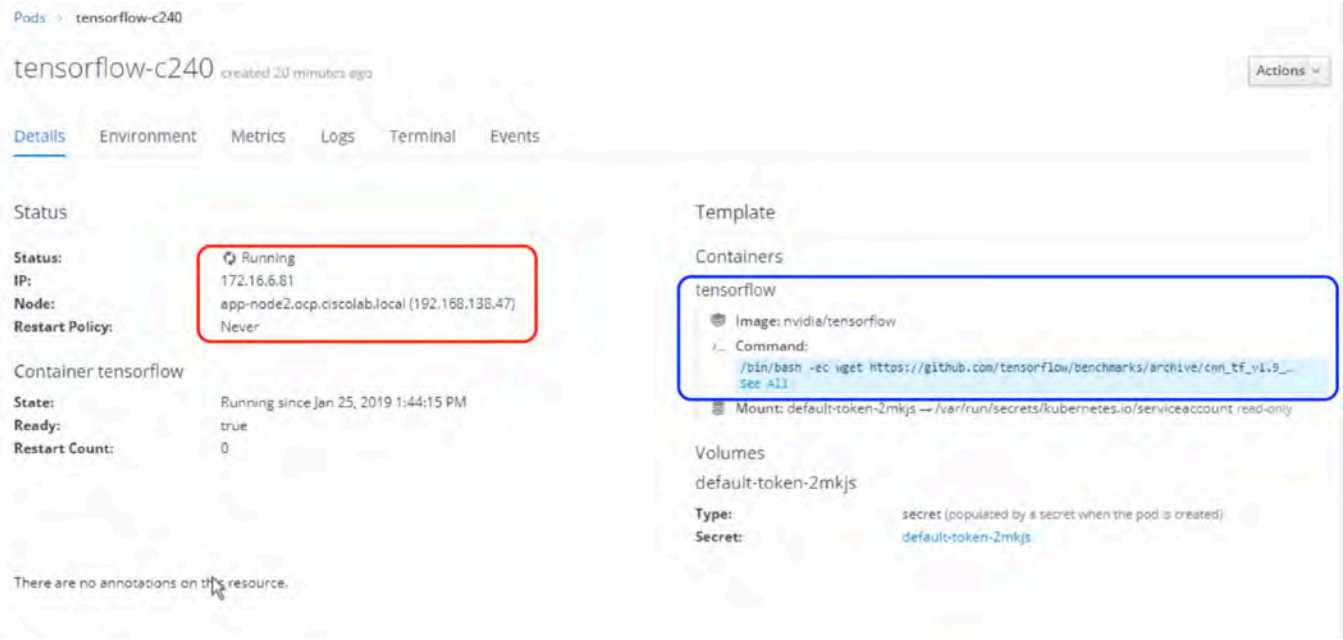


6. Similarly import the Yaml or JSON resource file you created for running Tensorflow benchmark on another App node (UCS C240 M5) testing 2 x NVIDIA V100 GPUs.

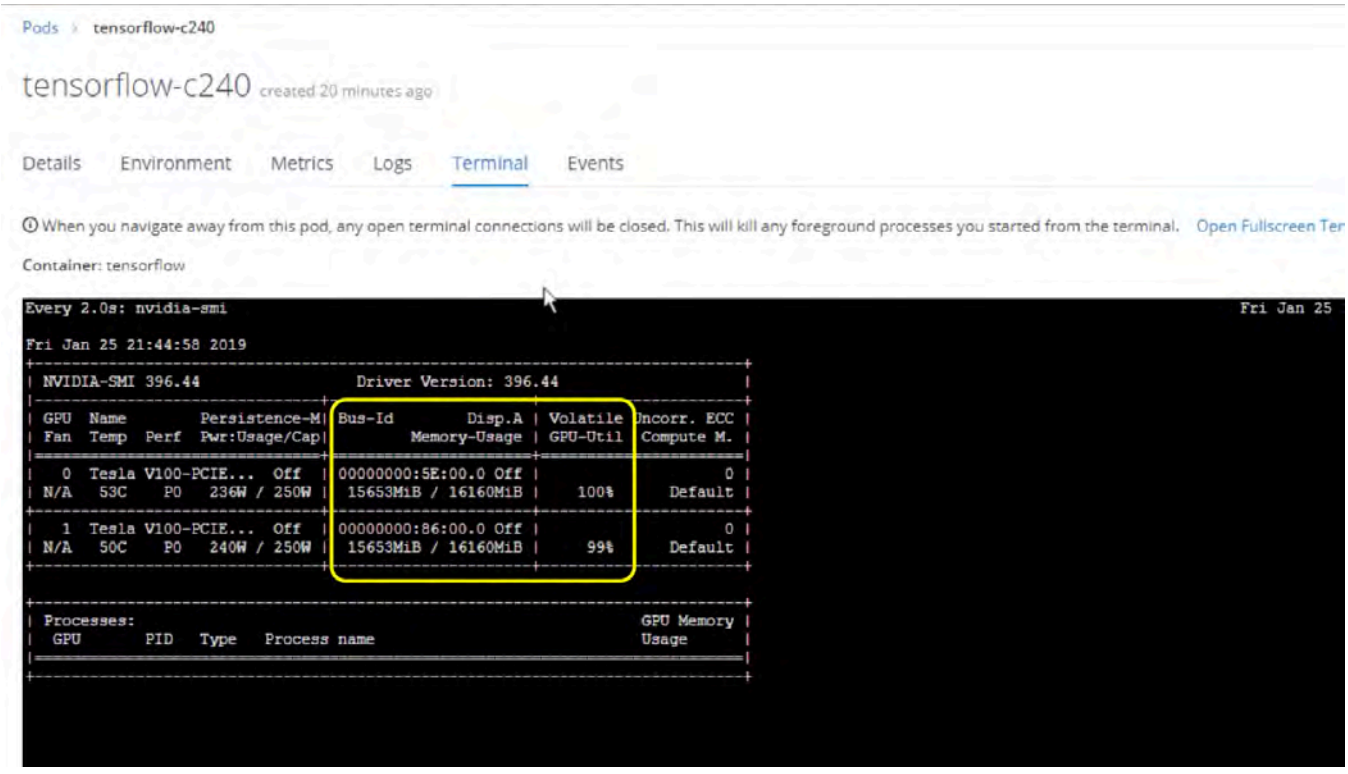
**Note:** "Refer Appendix A for creating .yaml file for running Tensorflow on C240 M5 server.



7. After successful creation of container pod “tensorflow-c240” for the App node (C240 M5), the container will automatically start the TensorFlow benchmark testing for DL model “resnet50”.




8. Click on Terminal tab to monitor the GPU utilization of NVIDIA V100 GPUs during the testing of DL model “Resnet50”.



## Conclusion

Experts and data scientists today believe that new industry leaders will be enterprises that invest in AI and turn their data into intelligence. Cisco has partnered with NVIDIA to design an affordable and simple yet powerful AI infrastructure on Cisco UCS rack servers that delivers high performance specifically for AI and machine-learning workloads.

Cisco now offers a solution using Cisco UCS infrastructure with the Cisco UCS C480 M5 ML Rack Server AI platform and Red Hat OpenShift Container Platform with container-native storage. This solution delivers a production-ready foundation that simplifies the deployment process, uses the latest best practices, and provides a stable, highly available environment for running containerized production applications and GPU-enabled containers for AI and machine-learning workloads.

## Appendix A: YAML configuration for running Tensorflow benchmark

Yaml configuration for "tensorflow-c480":

```
apiVersion: v1
kind: Pod
metadata:
  name: tensorflow-c480
  namespace: kube-system
labels:
   app: tensorflow
spec:
   hostNetwork: false
   restartPolicy: Never
   imagePullSecrets:
      - name: ngc
   containers:
    - name: tensorflow
      image: nvcr.io/nvidia/tensorflow:18.08-py3
      command: [ "/bin/bash", "-ec", "wget
https://github.com/tensorflow/benchmarks/archive/cnn_tf_v1.9_compatible.zip; unzip
cnn_tf_v1.9_compatible.zip;
cd benchmarks-cnn_tf_v1.9_compatible/scripts/tf_cnn_benchmarks/; python tf_cnn_benchmarks.py --
data_format=NHWC --batch_size=256 --model=resnet50 --optimizer=momentum --
variable_update=replicated --nodistortions --gradient_repacking=8 --num_gpus=8 --num_epochs=50 -
-weight_decay=1e-4 --all_reduce_spec=nccl --local_parameter_device=gpu --use_fp16" ]
      securityContext:
        privileged: true
      resources:
        limits:
          nvidia.com/gpu: 8
```

Yaml configuration for “tensorflow-c240”:

```
apiVersion: v1
kind: Pod
metadata:
  name: tensorflow-c240
  namespace: kube-system
labels:
   app: tensorflow
spec:
   hostNetwork: false
   restartPolicy: Never
   imagePullSecrets:
      - name: ngc
   containers:
    - name: tensorflow
      image: nvcr.io/nvidia/tensorflow:18.08-py3
      command: [ "/bin/bash", "-ec", "wget
https://github.com/tensorflow/benchmarks/archive/cnn_tf_v1.9_compatible.zip; unzip
cnn_tf_v1.9_compatible.zip;
cd benchmarks-cnn_tf_v1.9_compatible/scripts/tf_cnn_benchmarks/;
python tf_cnn_benchmarks.py --data_format=NHWC --batch_size=256 --model=resnet50 --
optimizer=momentum --variable_update=replicated --nodistortions --gradient_repacking=2 --
num_gpus=2 --num_epochs=50 --weight_decay=1e-4 --all_reduce_spec=nccl --
local_parameter_device=gpu --use_fp16" ]
      securityContext:
        privileged: true
      resources:
        limits:
          nvidia.com/gpu: 2
```

## About the Authors

**Tushar Patel**

Principal Engineer, UCS Performance and AI/ML Solutions, Cisco Systems

**Vijay Durairaj**

Technical Marketing Engineer, UCS Performance and Solutions, Cisco Systems

## Acknowledgements

**Vikas Ratna:** Cisco Systems

**Vishwanath Jakka:** Cisco Systems

**Sicong Ji:** Nvidia Corporation

## For more information

For additional information, see the following:

- Red Hat OpenShift Container Platform: https://www.redhat.com/en/technologies/cloud-computing/openshift
- Red Hat OpenShift Container Platform 3.11 architecture: https://access.redhat.com/documentation/en-us/openshift_container_platform/3.11/html/architecture/
- Cisco UCS C480 ML M5 Rack Server: https://www.cisco.com/c/en/us/products/collateral/serversunified-computing/ucs-c-series-rack-servers/datasheet-c78-741211.html
- NVIDIA GPU Cloud: https://www.nvidia.com/en-us/gpu-cloud/
- Deep-learning benchmarks and data sets:
  - TensorFlow CNN benchmark: https://github.com/TensorFlow/benchmarks/tree/master/scripts/tf_cnn_benchmarks
  - ImageNet data sets: http://www.image-net.org/

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 www.cisco.com/go/offices.



Printed in USA C11-742121-00 5/19