



## **NetApp Solutions**

NetApp Solutions

NetApp

August 18, 2021

This PDF was generated from <https://docs.netapp.com/us-en/netapp-solutions/index.html> on August 18, 2021. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# Table of Contents

NetApp Solutions .....	1
Artificial Intelligence .....	2
AI Converged Infrastructures .....	2
Data Pipelines, Data Lakes and Management .....	2
Use Cases .....	86
Modern Data Analytics .....	258
Hybrid Cloud / Virtualization .....	259
Get Started With NetApp & VMware .....	259
VMware Virtualization for ONTAP .....	262
VMware Private Cloud .....	332
Red Hat Private Cloud .....	332
Workload Performance .....	332
Demos and Tutorials .....	332
Virtual Desktops .....	333
Virtual Desktop Services (VDS) .....	333
VMware Horizon .....	372
Citrix Virtual Apps and Desktops .....	372
Virtual Desktop Applications .....	403
Containers .....	404
Archived Solutions .....	404
NVA-1160: Red Hat OpenShift with NetApp .....	439
Google Anthos .....	575
Enterprise Applications and Databases .....	662
Oracle Database .....	662
Microsoft SQL Server .....	671
Data Protection and Security .....	685
Data Protection .....	685
Security .....	717
Infrastructure .....	718
NVA-1148: NetApp HCI with Red Hat Virtualization .....	718
TR-4857: NetApp HCI with Cisco ACI .....	801
Solution Automation .....	839
NetApp Solution Automation .....	839
Setup the Ansible control node (For CLI based deployments) .....	839
NetApp solution automation .....	841
NetApp Solutions Change Log .....	844
About this Repository .....	846
Navigation of the Repository .....	846
PDF Generation .....	847
Change Log .....	847
Feedback .....	847
Legal notices .....	848
Copyright .....	848

Trademarks .....	848
Patents .....	848
Privacy policy .....	848
Open source .....	848

# **NetApp Solutions**

# Artificial Intelligence

## AI Converged Infrastructures

### ONTAP AI with NVIDIA

### EF-Series AI with NVIDIA

## Data Pipelines, Data Lakes and Management

### NetApp AI Control Plane

#### NetApp AI Control Plane

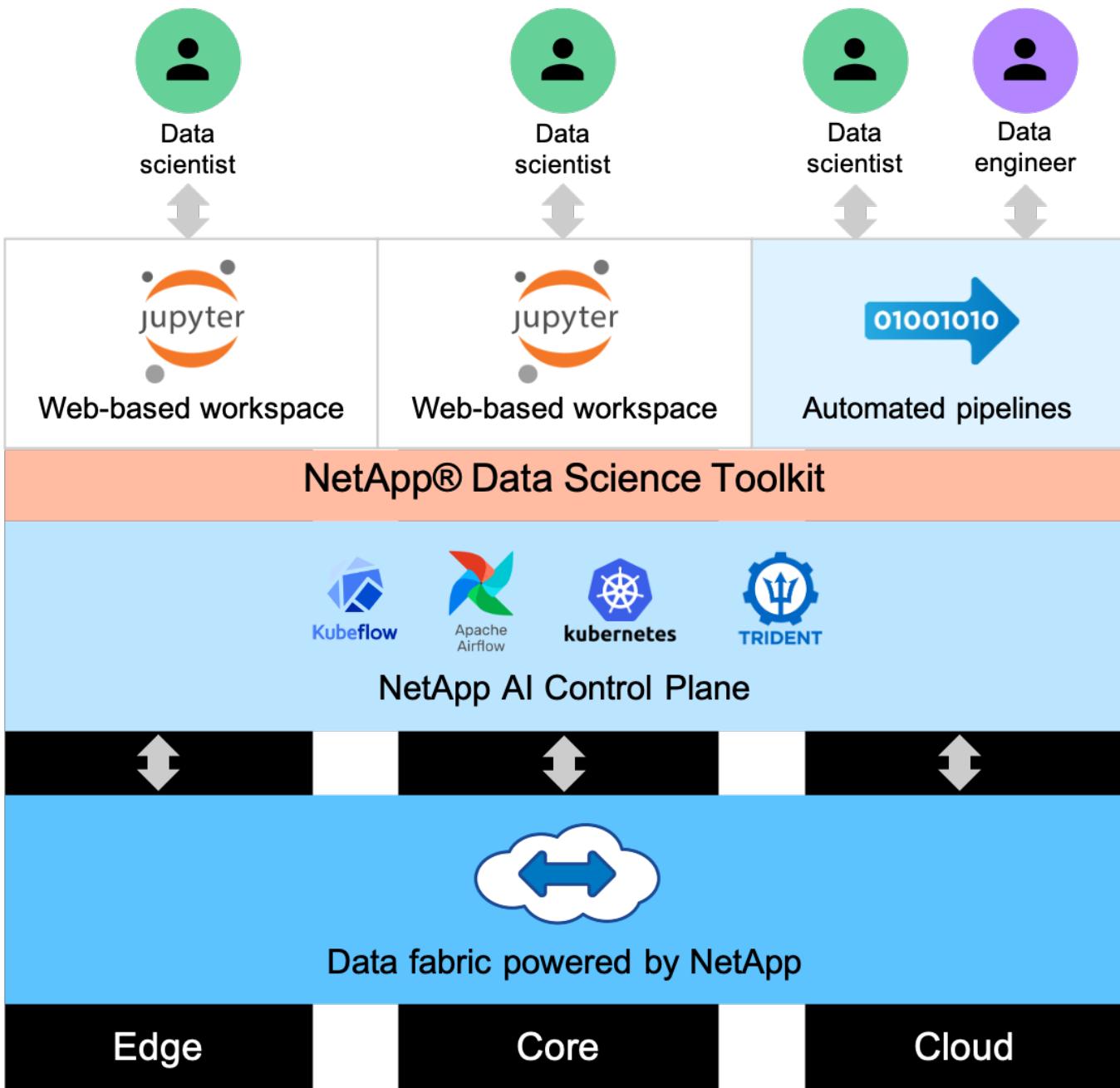
Mike Oglesby, NetApp

Companies and organizations of all sizes and across many industries are turning to artificial intelligence (AI), machine learning (ML), and deep learning (DL) to solve real-world problems, deliver innovative products and services, and to get an edge in an increasingly competitive marketplace. As organizations increase their use of AI, ML, and DL, they face many challenges, including workload scalability and data availability. This document demonstrates how you can address these challenges by using the NetApp AI Control Plane, a solution that pairs NetApp data management capabilities with popular open-source tools and frameworks.

This report shows you how to rapidly clone a data namespace. It also shows you how to seamlessly replicate data across sites and regions to create a cohesive and unified AI/ML/DL data pipeline. Additionally, it walks you through the defining and implementing of AI, ML, and DL training workflows that incorporate the near-instant creation of data and model baselines for traceability and versioning. With this solution, you can trace every model training run back to the exact dataset that was used to train and/or validate the model. Lastly, this document shows you how to swiftly provision Jupyter Notebook workspaces with access to massive datasets.

Note: For HPC style distributed training at scale involving a large number of GPU servers that require shared access to the same dataset, or if you require/prefer a parallel file system, check out [TR-4890](#). This technical report describes how to include [NetApp's fully supported parallel file system solution BeeGFS](#) as part of the NetApp AI Control Plane. This solution is designed to scale from a handful of NVIDIA DGX A100 systems, up to a full blown 140 node SuperPOD.

The NetApp AI Control Plane is targeted towards data scientists and data engineers, and, thus, minimal NetApp or NetApp ONTAP® expertise is required. With this solution, data management functions can be executed using simple and familiar tools and interfaces. If you already have NetApp storage in your environment, you can test drive the NetApp AI Control plane today. If you want to test drive the solution but you do not have already have NetApp storage, visit [cloud.netapp.com](#), and you can be up and running with a cloud-based NetApp storage solution in minutes. The following figure provides a visualization of the solution.



[Next: Concepts and Components](#)

## Concepts and Components

### Artificial Intelligence

AI is a computer science discipline in which computers are trained to mimic the cognitive functions of the human mind. AI developers train computers to learn and to solve problems in a manner that is similar to, or even superior to, humans. Deep learning and machine learning are subfields of AI. Organizations are increasingly adopting AI, ML, and DL to support their critical business needs. Some examples are as follows:

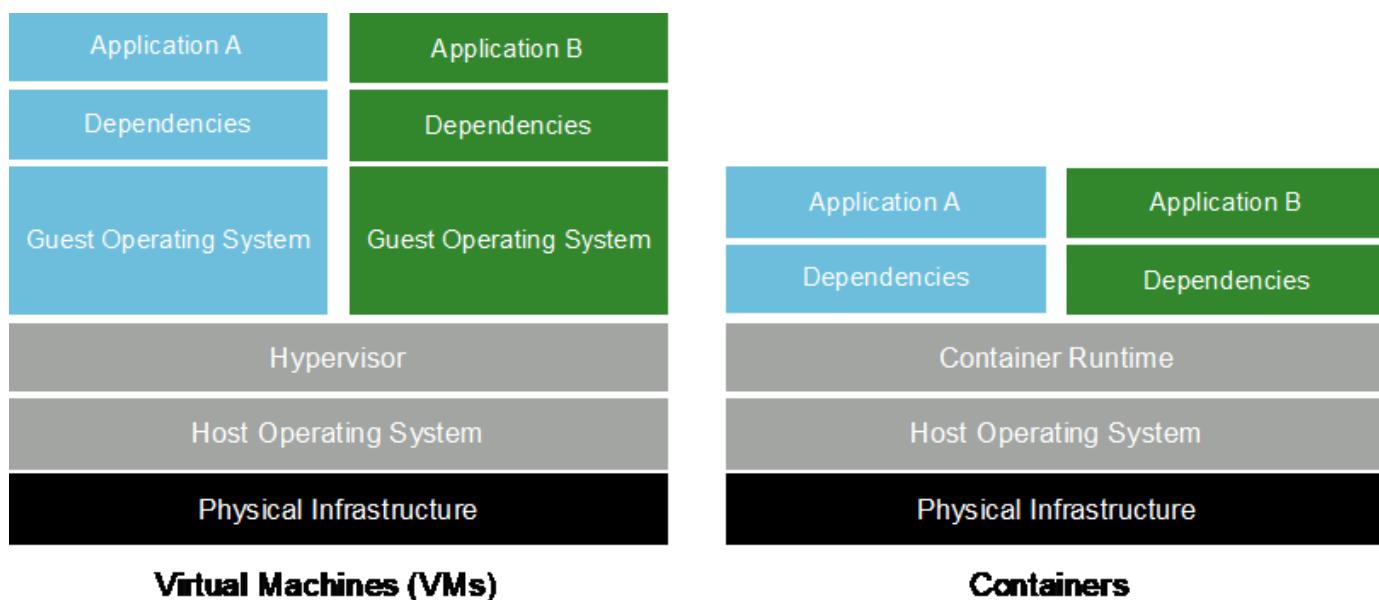
- Analyzing large amounts of data to unearth previously unknown business insights
- Interacting directly with customers by using natural language processing
- Automating various business processes and functions

Modern AI training and inference workloads require massively parallel computing capabilities. Therefore, GPUs are increasingly being used to execute AI operations because the parallel processing capabilities of GPUs are vastly superior to those of general-purpose CPUs.

## Containers

Containers are isolated user-space instances that run on top of a shared host operating system kernel. The adoption of containers is increasing rapidly. Containers offer many of the same application sandboxing benefits that virtual machines (VMs) offer. However, because the hypervisor and guest operating system layers that VMs rely on have been eliminated, containers are far more lightweight. The following figure depicts a visualization of virtual machines versus containers.

Containers also allow the efficient packaging of application dependencies, run times, and so on, directly with an application. The most commonly used container packaging format is the Docker container. An application that has been containerized in the Docker container format can be executed on any machine that can run Docker containers. This is true even if the application's dependencies are not present on the machine because all dependencies are packaged in the container itself. For more information, visit the [Docker website](#).



## Kubernetes

Kubernetes is an open source, distributed, container orchestration platform that was originally designed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF). Kubernetes enables the automation of deployment, management, and scaling functions for containerized applications. In recent years, Kubernetes has emerged as the dominant container orchestration platform. Although other container packaging formats and run times are supported, Kubernetes is most often used as an orchestration system for Docker containers. For more information, visit the [Kubernetes website](#).

## NetApp Trident

Trident is an open source storage orchestrator developed and maintained by NetApp that greatly simplifies the creation, management, and consumption of persistent storage for Kubernetes workloads. Trident, itself a Kubernetes-native application, runs directly within a Kubernetes cluster. With Trident, Kubernetes users (developers, data scientists, Kubernetes administrators, and so on) can create, manage, and interact with persistent storage volumes in the standard Kubernetes format that they are already familiar with. At the same time, they can take advantage of NetApp advanced data management capabilities and a data fabric that is powered by NetApp technology. Trident abstracts away the complexities of persistent storage and makes it

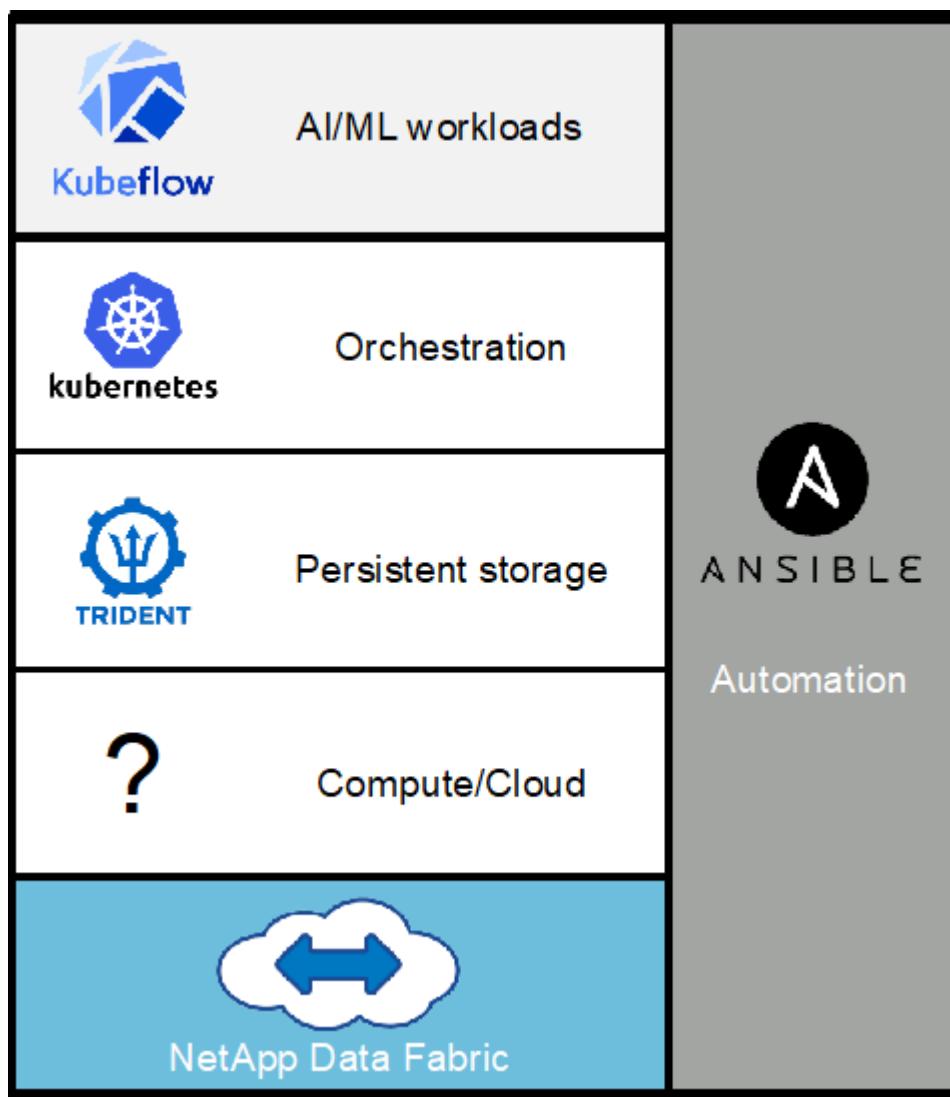
simple to consume. For more information, visit the [Trident website](#).

## NVIDIA DeepOps

DeepOps is an open source project from NVIDIA that, by using Ansible, automates the deployment of GPU server clusters according to best practices. DeepOps is modular and can be used for various deployment tasks. For this document and the validation exercise that it describes, DeepOps is used to deploy a Kubernetes cluster that consists of GPU server worker nodes. For more information, visit the [DeepOps website](#).

## Kubeflow

Kubeflow is an open source AI and ML toolkit for Kubernetes that was originally developed by Google. The Kubeflow project makes deployments of AI and ML workflows on Kubernetes simple, portable, and scalable. Kubeflow abstracts away the intricacies of Kubernetes, allowing data scientists to focus on what they know best—data science. See the following figure for a visualization. Kubeflow has been gaining significant traction as enterprise IT departments have increasingly standardized on Kubernetes. For more information, visit the [Kubeflow website](#).



## Kubeflow Pipelines

Kubeflow Pipelines are a key component of Kubeflow. Kubeflow Pipelines are a platform and standard for defining and deploying portable and scalable AI and ML workflows. For more information, see the [official](#)

[Kubeflow documentation](#).

## Jupyter Notebook Server

A Jupyter Notebook Server is an open source web application that allows data scientists to create wiki-like documents called Jupyter Notebooks that contain live code as well as descriptive text. Jupyter Notebooks are widely used in the AI and ML community as a means of documenting, storing, and sharing AI and ML projects. Kubeflow simplifies the provisioning and deployment of Jupyter Notebook Servers on Kubernetes. For more information on Jupyter Notebooks, visit the [Jupyter website](#). For more information about Jupyter Notebooks within the context of Kubeflow, see the [official Kubeflow documentation](#).

## Apache Airflow

Apache Airflow is an open-source workflow management platform that enables programmatic authoring, scheduling, and monitoring for complex enterprise workflows. It is often used to automate ETL and data pipeline workflows, but it is not limited to these types of workflows. The Airflow project was started by Airbnb but has since become very popular in the industry and now falls under the auspices of The Apache Software Foundation. Airflow is written in Python, Airflow workflows are created via Python scripts, and Airflow is designed under the principle of "configuration as code." Many enterprise Airflow users now run Airflow on top of Kubernetes.

## Directed Acyclic Graphs (DAGs)

In Airflow, workflows are called Directed Acyclic Graphs (DAGs). DAGs are made up of tasks that are executed in sequence, in parallel, or a combination of the two, depending on the DAG definition. The Airflow scheduler executes individual tasks on an array of workers, adhering to the task-level dependencies that are specified in the DAG definition. DAGs are defined and created via Python scripts.

## NetApp ONTAP 9

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

## Simplify Data Management

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

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

## Accelerate and Protect Data

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

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

## Future-Proof Infrastructure

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

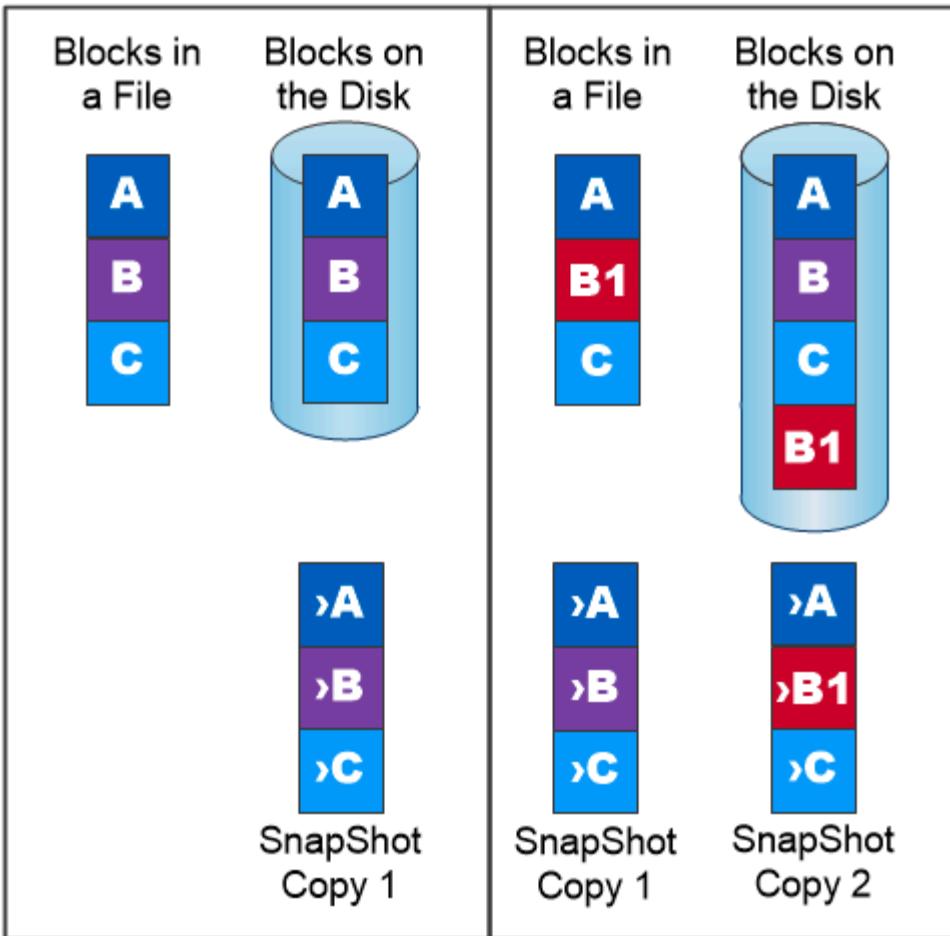
- **Seamless scaling and nondisruptive operations.** ONTAP supports the nondisruptive addition of capacity to existing controllers and to scale-out clusters. You can upgrade to the latest technologies, such as NVMe and 32Gb FC, without costly data migrations or outages.
- **Cloud connection.** ONTAP is one of the most cloud-connected storage management software, with options for software-defined storage (ONTAP Select) and cloud-native instances (NetApp Cloud Volumes Service) in all public clouds.
- **Integration with emerging applications.** By using the same infrastructure that supports existing enterprise apps, ONTAP offers enterprise-grade data services for next-generation platforms and applications such as OpenStack, Hadoop, and MongoDB.

## NetApp Snapshot Copies

A NetApp Snapshot copy is a read-only, point-in-time image of a volume. The image consumes minimal storage space and incurs negligible performance overhead because it only records changes to files created since the last Snapshot copy was made, as depicted in the following figure.

Snapshot copies owe their efficiency to the core ONTAP storage virtualization technology, the Write Anywhere File Layout (WAFL). Like a database, WAFL uses metadata to point to actual data blocks on disk. But, unlike a database, WAFL does not overwrite existing blocks. It writes updated data to a new block and changes the metadata. It's because ONTAP references metadata when it creates a Snapshot copy, rather than copying data blocks, that Snapshot copies are so efficient. Doing so eliminates the seek time that other systems incur in locating the blocks to copy, as well as the cost of making the copy itself.

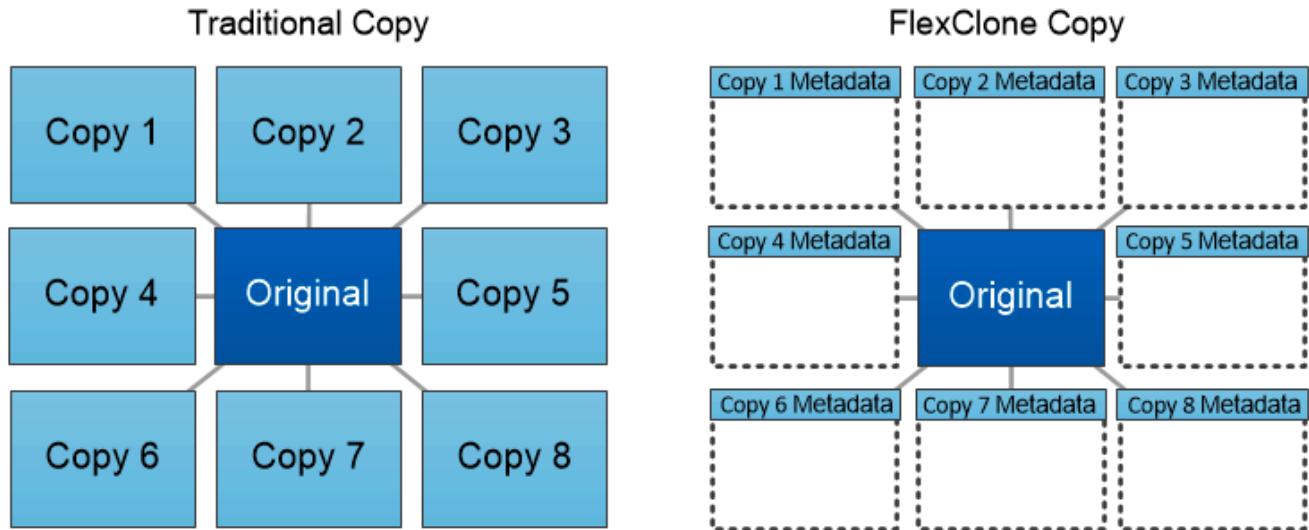
You can use a Snapshot copy to recover individual files or LUNs or to restore the entire contents of a volume. ONTAP compares pointer information in the Snapshot copy with data on disk to reconstruct the missing or damaged object, without downtime or a significant performance cost.



*A Snapshot copy records only changes to the active file system since the last Snapshot copy.*

#### NetApp FlexClone Technology

NetApp FlexClone technology references Snapshot metadata to create writable, point-in-time copies of a volume. Copies share data blocks with their parents, consuming no storage except what is required for metadata until changes are written to the copy, as depicted in the following figure. Where traditional copies can take minutes or even hours to create, FlexClone software lets you copy even the largest datasets almost instantaneously. That makes it ideal for situations in which you need multiple copies of identical datasets (a development workspace, for example) or temporary copies of a dataset (testing an application against a production dataset).

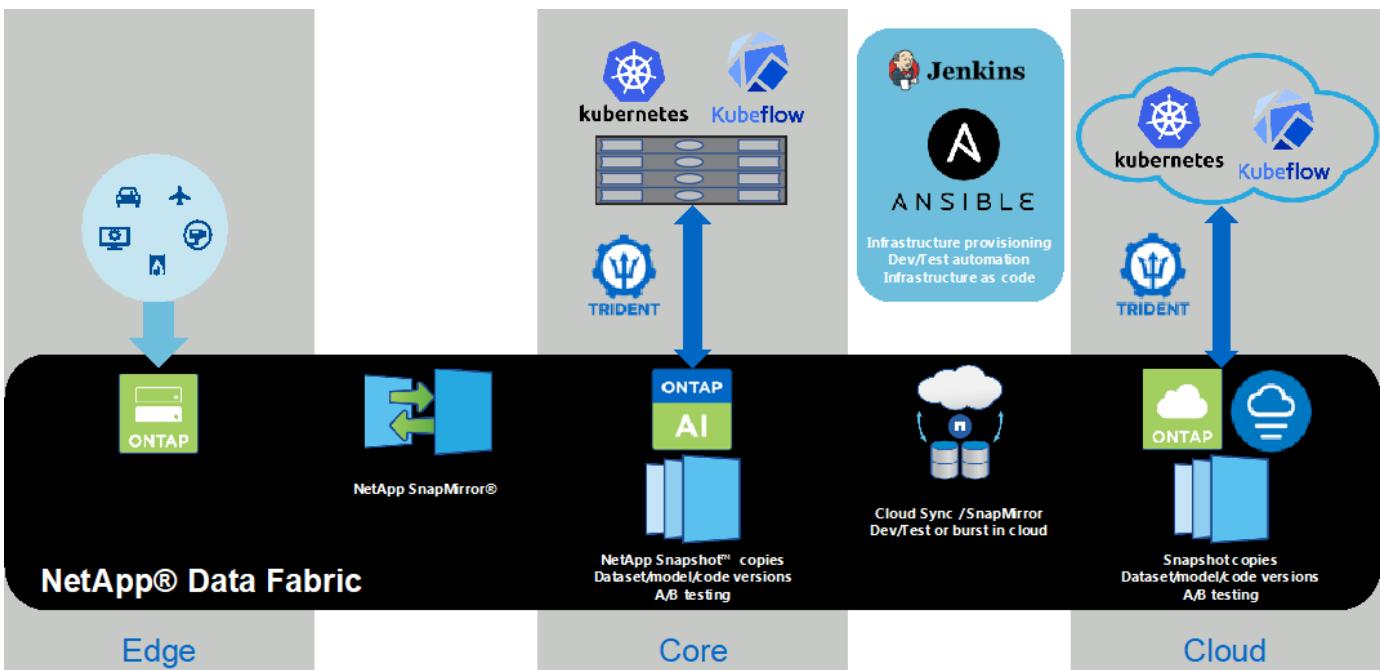


*FlexClone copies share data blocks with their parents, consuming no storage except what is required for metadata.*

#### NetApp SnapMirror Data Replication Technology

NetApp SnapMirror software is a cost-effective, easy-to-use unified replication solution across the data fabric. It replicates data at high speeds over LAN or WAN. It gives you high data availability and fast data replication for applications of all types, including business critical applications in both virtual and traditional environments. When you replicate data to one or more NetApp storage systems and continually update the secondary data, your data is kept current and is available whenever you need it. No external replication servers are required. See the following figure for an example of an architecture that leverages SnapMirror technology.

SnapMirror software leverages NetApp ONTAP storage efficiencies by sending only changed blocks over the network. SnapMirror software also uses built-in network compression to accelerate data transfers and reduce network bandwidth utilization by up to 70%. With SnapMirror technology, you can leverage one thin replication data stream to create a single repository that maintains both the active mirror and prior point-in-time copies, reducing network traffic by up to 50%.



### NetApp Cloud Sync

Cloud Sync is a NetApp service for rapid and secure data synchronization. Whether you need to transfer files between on-premises NFS or SMB file shares, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, AWS S3, AWS EFS, Azure Blob, Google Cloud Storage, or IBM Cloud Object Storage, Cloud Sync moves the files where you need them quickly and securely.

After your data is transferred, it is fully available for use on both source and target. Cloud Sync can sync data on-demand when an update is triggered or continuously sync data based on a predefined schedule. Regardless, Cloud Sync only moves the deltas, so time and money spent on data replication is minimized.

Cloud Sync is a software as a service (SaaS) tool that is extremely simple to set up and use. Data transfers that are triggered by Cloud Sync are carried out by data brokers. Cloud Sync data brokers can be deployed in AWS, Azure, Google Cloud Platform, or on-premises.

### NetApp XCP

NetApp XCP is client-based software for any-to-NetApp and NetApp-to-NetApp data migrations and file system insights. XCP is designed to scale and achieve maximum performance by utilizing all available system resources to handle high-volume datasets and high-performance migrations. XCP helps you to gain complete visibility into the file system with the option to generate reports.

NetApp XCP is available in a single package that supports NFS and SMB protocols. XCP includes a Linux binary for NFS data sets and a windows executable for SMB data sets.

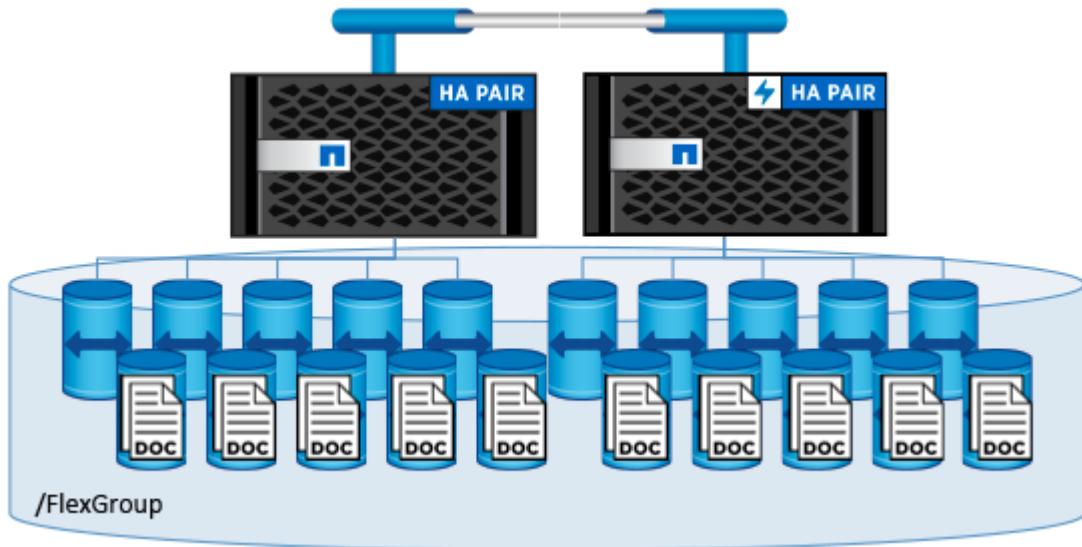
NetApp XCP File Analytics is host-based software that detects file shares, runs scans on the file system, and provides a dashboard for file analytics. XCP File Analytics is compatible with both NetApp and non-NetApp systems and runs on Linux or Windows hosts to provide analytics for NFS and SMB-exported file systems.

### NetApp ONTAP FlexGroup Volumes

A training dataset can be a collection of potentially billions of files. Files can include text, audio, video, and other forms of unstructured data that must be stored and processed to be read in parallel. The storage system must store large numbers of small files and must read those files in parallel for sequential and random I/O.

A FlexGroup volume is a single namespace that comprises multiple constituent member volumes, as shown in the following figure. From a storage administrator viewpoint, a FlexGroup volume is managed and acts like a NetApp FlexVol volume. Files in a FlexGroup volume are allocated to individual member volumes and are not striped across volumes or nodes. They enable the following capabilities:

- FlexGroup volumes provide multiple petabytes of capacity and predictable low latency for high-metadata workloads.
- They support up to 400 billion files in the same namespace.
- They support parallelized operations in NAS workloads across CPUs, nodes, aggregates, and constituent FlexVol volumes.



[Next: Hardware and Software Requirements](#)

### Hardware and Software Requirements

The NetApp AI Control Plane solution is not dependent on this specific hardware. The solution is compatible with any NetApp physical storage appliance, software-defined instance, or cloud service, that is supported by Trident. Examples include a NetApp AFF storage system, Azure NetApp Files, NetApp Cloud Volumes Service, a NetApp ONTAP Select software-defined storage instance, or a NetApp Cloud Volumes ONTAP instance. Additionally, the solution can be implemented on any Kubernetes cluster as long as the Kubernetes version used is supported by Kubeflow and NetApp Trident. For a list of Kubernetes versions that are supported by Kubeflow, see the [official Kubeflow documentation](#). For a list of Kubernetes versions that are supported by Trident, see the [Trident documentation](#). See the following tables for details on the environment that was used to validate the solution.

Infrastructure Component	Quantity	Details	Operating System
Deployment jump host	1	VM	Ubuntu 20.04.2 LTS

Infrastructure Component	Quantity	Details	Operating System
Kubernetes master nodes	1	VM	Ubuntu 20.04.2 LTS
Kubernetes worker nodes	2	VM	Ubuntu 20.04.2 LTS
Kubernetes GPU worker nodes	2	NVIDIA DGX-1 (bare-metal)	NVIDIA DGX OS 4.0.5 (based on Ubuntu 18.04.2 LTS)
Storage	1 HA Pair	NetApp AFF A220	NetApp ONTAP 9.7 P6

Software Component	Version
Apache Airflow	2.0.1
Apache Airflow Helm Chart	8.0.8
Docker	19.03.12
Kubeflow	1.2
Kubernetes	1.18.9
NetApp Trident	21.01.2
NVIDIA DeepOps	Trident deployment functionality from master branch as of commit <a href="#">61898cdfda</a> ; All other functionality from version 21.03

## Support

NetApp does not offer enterprise support for Apache Airflow, Docker, Kubeflow, Kubernetes, or NVIDIA DeepOps. If you are interested in a fully supported solution with capabilities similar to the NetApp AI Control Plane solution, [contact NetApp](#) about fully supported AI/ML solutions that NetApp offers jointly with partners.

[Next: Kubernetes Deployment.](#)

## Kubernetes Deployment

This section describes the tasks that you must complete to deploy a Kubernetes cluster in which to implement the NetApp AI Control Plane solution. If you already have a Kubernetes cluster, then you can skip this section as long as you are running a version of Kubernetes that is supported by Kubeflow and NetApp Trident. For a list of Kubernetes versions that are supported by Kubeflow, see the [official Kubeflow documentation](#). For a list of Kubernetes versions that are supported by Trident, see the [Trident documentation](#).

For on-premises Kubernetes deployments that incorporate bare-metal nodes featuring NVIDIA GPU(s), NetApp recommends using NVIDIA's DeepOps Kubernetes deployment tool. This section outlines the deployment of a Kubernetes cluster using DeepOps.

## Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already

performed the following tasks:

1. You have already configured any bare-metal Kubernetes nodes (for example, an NVIDIA DGX system that is part of an ONTAP AI pod) according to standard configuration instructions.
2. You have installed a supported operating system on all Kubernetes master and worker nodes and on a deployment jump host. For a list of operating systems that are supported by DeepOps, see the [DeepOps GitHub site](#).

## Use NVIDIA DeepOps to Install and Configure Kubernetes

To deploy and configure your Kubernetes cluster with NVIDIA DeepOps, perform the following tasks from a deployment jump host:

1. Download NVIDIA DeepOps by following the instructions on the [Getting Started page](#) on the NVIDIA DeepOps GitHub site.
2. Deploy Kubernetes in your cluster by following the instructions on the [Kubernetes Deployment Guide page](#) on the NVIDIA DeepOps GitHub site.

Next: [NetApp Trident Deployment and Configuration Overview](#)

## NetApp Trident Deployment and Configuration

This section describes the tasks that you must complete to install and configure NetApp Trident in your Kubernetes cluster.

### Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You already have a working Kubernetes cluster, and you are running a version of Kubernetes that is supported by Trident. For a list of supported versions, see the [Trident documentation](#).
2. You already have a working NetApp storage appliance, software-defined instance, or cloud storage service, that is supported by Trident.

### Install Trident

To install and configure NetApp Trident in your Kubernetes cluster, perform the following tasks from the deployment jump host:

1. Deploy Trident using one of the following methods:
  - If you used NVIDIA DeepOps to deploy your Kubernetes cluster, you can also use NVIDIA DeepOps to deploy Trident in your Kubernetes cluster. To deploy Trident with DeepOps, follow the [Trident deployment instructions](#) on the NVIDIA DeepOps GitHub site.
  - If you did not use NVIDIA DeepOps to deploy your Kubernetes cluster or if you simply prefer to deploy Trident manually, you can deploy Trident by following the [deployment instructions](#) in the Trident documentation. Be sure to create at least one Trident Backend and at least one Kubernetes StorageClass. For more information about Backends and StorageClasses, see the [Trident documentation](#).

 If you are deploying the NetApp AI Control Plane solution on an ONTAP AI pod, see [Example Trident Backends for ONTAP AI Deployments](#) for some examples of different Trident Backends that you might want to create and [Example Kubernetes Storageclasses for ONTAP AI Deployments](#) for some examples of different Kubernetes StorageClasses that you might want to create.

Next: [Example Trident Backends for ONTAP AI Deployments](#)

### NetApp Trident Deployment and Configuration

This section describes the tasks that you must complete to install and configure NetApp Trident in your Kubernetes cluster.

#### Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You already have a working Kubernetes cluster, and you are running a version of Kubernetes that is supported by Trident. For a list of supported versions, see the [Trident documentation](#).
2. You already have a working NetApp storage appliance, software-defined instance, or cloud storage service, that is supported by Trident.

#### Install Trident

To install and configure NetApp Trident in your Kubernetes cluster, perform the following tasks from the deployment jump host:

1. Deploy Trident using one of the following methods:
  - If you used NVIDIA DeepOps to deploy your Kubernetes cluster, you can also use NVIDIA DeepOps to deploy Trident in your Kubernetes cluster. To deploy Trident with DeepOps, follow the [Trident deployment instructions](#) on the NVIDIA DeepOps GitHub site.
  - If you did not use NVIDIA DeepOps to deploy your Kubernetes cluster or if you simply prefer to deploy Trident manually, you can deploy Trident by following the [deployment instructions](#) in the Trident documentation. Be sure to create at least one Trident Backend and at least one Kubernetes StorageClass. For more information about Backends and StorageClasses, see the [Trident documentation](#).

 If you are deploying the NetApp AI Control Plane solution on an ONTAP AI pod, see [Example Trident Backends for ONTAP AI Deployments](#) for some examples of different Trident Backends that you might want to create and [Example Kubernetes Storageclasses for ONTAP AI Deployments](#) for some examples of different Kubernetes StorageClasses that you might want to create.

Next: [Example Trident Backends for ONTAP AI Deployments](#)

### Example Trident Backends for ONTAP AI Deployments

Before you can use Trident to dynamically provision storage resources within your Kubernetes cluster, you must create one or more Trident Backends. The examples that follow represent different types of Backends that you might want to create if you are

deploying the NetApp AI Control Plane solution on an ONTAP AI pod. For more information about Backends, see the [Trident documentation](#).

1. NetApp recommends creating a FlexGroup-enabled Trident Backend for each data LIF (logical network interface that provides data access) that you want to use on your NetApp AFF system. This will allow you to balance volume mounts across LIFs

The example commands that follow show the creation of two FlexGroup-enabled Trident Backends for two different data LIFs that are associated with the same ONTAP storage virtual machine (SVM). These Backends use the `ontap-nas-flexgroup` storage driver. ONTAP supports two main data volume types: FlexVol and FlexGroup. FlexVol volumes are size-limited (as of this writing, the maximum size depends on the specific deployment). FlexGroup volumes, on the other hand, can scale linearly to up to 20PB and 400 billion files, providing a single namespace that greatly simplifies data management. Therefore, FlexGroup volumes are optimal for AI and ML workloads that rely on large amounts of data.

If you are working with a small amount of data and want to use FlexVol volumes instead of FlexGroup volumes, you can create Trident Backends that use the `ontap-nas` storage driver instead of the `ontap-nas-flexgroup` storage driver.

```
$ cat << EOF > ./trident-backend-ontap-ai-flexgroups-iface1.json
{
    "version": 1,
    "storageDriverName": "ontap-nas-flexgroup",
    "backendName": "ontap-ai-flexgroups-iface1",
    "managementLIF": "10.61.218.100",
    "dataLIF": "192.168.11.11",
    "svm": "ontapai_nfs",
    "username": "admin",
    "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexgroups-
iface1.json -n trident
+-----+
+-----+-----+
|           NAME          |   STORAGE DRIVER   |
UUID          | STATE   | VOLUMES  |
+-----+-----+
+-----+-----+-----+
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online |      0 |
+-----+
+-----+-----+-----+
$ cat << EOF > ./trident-backend-ontap-ai-flexgroups-iface2.json
{
    "version": 1,
    "storageDriverName": "ontap-nas-flexgroup",
    "backendName": "ontap-ai-flexgroups-iface2",
```

```

    "managementLIF": "10.61.218.100",
    "dataLIF": "192.168.12.12",
    "svm": "ontapai_nfs",
    "username": "admin",
    "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexgroups-
iface2.json -n trident
+-----+-----+
+-----+-----+-----+
|           NAME          |   STORAGE DRIVER   |
UUID          | STATE   | VOLUMES |
+-----+-----+
+-----+-----+-----+
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |      0 |
+-----+-----+
+-----+-----+-----+
$ tridentctl get backend -n trident
+-----+-----+
+-----+-----+-----+
|           NAME          |   STORAGE DRIVER   |
UUID          | STATE   | VOLUMES |
+-----+-----+
+-----+-----+-----+
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online |      0 |
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |      0 |
+-----+-----+
+-----+-----+-----+

```

2. NetApp also recommends creating one or more FlexVol- enabled Trident Backends. If you use FlexGroup volumes for training dataset storage, you might want to use FlexVol volumes for storing results, output, debug information, and so on. If you want to use FlexVol volumes, you must create one or more FlexVol-enabled Trident Backends. The example commands that follow show the creation of a single FlexVol-enabled Trident Backend that uses a single data LIF.

```

$ cat << EOF > ./trident-backend-ontap-ai-flexvols.json
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "ontap-ai-flexvols",
    "managementLIF": "10.61.218.100",
    "dataLIF": "192.168.11.11",
    "svm": "ontapai_nfs",
    "username": "admin",
    "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexvols.json -n
trident
+-----+
+-----+-----+
|           NAME          |   STORAGE DRIVER   |           UUID
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+
| ontap-ai-flexvols      | ontap-nas          | 52bdb3b1-13a5-4513-
a9c1-52a69657fabe | online | 0 |
+-----+-----+
+-----+-----+
$ tridentctl get backend -n trident
+-----+
+-----+-----+
|           NAME          |   STORAGE DRIVER   |           UUID
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+
| ontap-ai-flexvols      | ontap-nas          | 52bdb3b1-13a5-4513-
a9c1-52a69657fabe | online | 0 |
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online | 0 |
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online | 0 |
+-----+-----+
+-----+-----+

```

[Next: Example Kubernetes Storageclasses for ONTAP AI Deployments](#)

#### Example Kubernetes StorageClasses for ONTAP AI Deployments

Before you can use Trident to dynamically provision storage resources within your

Kubernetes cluster, you must create one or more Kubernetes StorageClasses. The examples that follow represent different types of StorageClasses that you might want to create if you are deploying the NetApp AI Control Plane solution on an ONTAP AI pod. For more information about StorageClasses, see the [Trident documentation](#).

1. NetApp recommends creating a separate StorageClass for each FlexGroup-enabled Trident Backend that you created in the section [Example Trident Backends for ONTAP AI Deployments](#), step 1. These granular StorageClasses enable you to add NFS mounts that correspond to specific LIFs (the LIFs that you specified when you created the Trident Backends) as a particular Backend that is specified in the StorageClass spec file. The example commands that follow show the creation of two StorageClasses that correspond to the two example Backends that were created in the section [Example Trident Backends for ONTAP AI Deployments](#), step 1. For more information about StorageClasses, see the [Trident documentation](#).

So that a persistent volume isn't deleted when the corresponding PersistentVolumeClaim (PVC) is deleted, the following example uses a `reclaimPolicy` value of `Retain`. For more information about the `reclaimPolicy` field, see the official [Kubernetes documentation](#).

```

$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain-iface1.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain-iface1
  provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
  storagePools: "ontap-ai-flexgroups-iface1:.*"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain-
iface1.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain-iface1 created
$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain-iface2.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain-iface2
  provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
  storagePools: "ontap-ai-flexgroups-iface2:.*"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain-
iface2.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain-iface2 created
$ kubectl get storageclass
NAME                      PROVISIONER          AGE
ontap-ai-flexgroups-retain-iface1   netapp.io/trident   0m
ontap-ai-flexgroups-retain-iface2   netapp.io/trident   0m

```

2. NetApp also recommends creating a StorageClass that corresponds to the FlexVol-enabled Trident Backend that you created in the section [Example Trident Backends for ONTAP AI Deployments](#), step 2. The example commands that follow show the creation of a single StorageClass for FlexVol volumes.

In the following example, a particular Backend is not specified in the StorageClass definition file because only one FlexVol-enabled Trident backend was created. When you use Kubernetes to administer volumes that use this StorageClass, Trident attempts to use any available backend that uses the `ontap-nas` driver.

```

$ cat << EOF > ./storage-class-ontap-ai-flexvols-retain.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexvols-retain
  provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas"
  reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexvols-retain.yaml
storageclass.storage.k8s.io/ontap-ai-flexvols-retain created
$ kubectl get storageclass
NAME                      PROVISIONER          AGE
ontap-ai-flexgroups-retain-iface1   netapp.io/trident   1m
ontap-ai-flexgroups-retain-iface2   netapp.io/trident   1m
ontap-ai-flexvols-retain           netapp.io/trident   0m

```

3. NetApp also recommends creating a generic StorageClass for FlexGroup volumes. The following example commands show the creation of a single generic StorageClass for FlexGroup volumes.

Note that a particular backend is not specified in the StorageClass definition file. Therefore, when you use Kubernetes to administer volumes that use this StorageClass, Trident attempts to use any available backend that uses the `ontap-nas-flexgroup` driver.

```

$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain
  provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
  reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain created
$ kubectl get storageclass
NAME                      PROVISIONER          AGE
ontap-ai-flexgroups-retain           netapp.io/trident   0m
ontap-ai-flexgroups-retain-iface1   netapp.io/trident   2m
ontap-ai-flexgroups-retain-iface2   netapp.io/trident   2m
ontap-ai-flexvols-retain           netapp.io/trident   1m

```

[Next: Kubeflow Deployment Overview](#)

## Kubeflow Deployment

This section describes the tasks that you must complete to deploy Kubeflow in your Kubernetes cluster.

### Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You already have a working Kubernetes cluster, and you are running a version of Kubernetes that is supported by Kubeflow. For a list of supported versions, see the [official Kubeflow documentation](#).
2. You have already installed and configured NetApp Trident in your Kubernetes cluster as outlined in [Trident Deployment and Configuration](#).

### Set Default Kubernetes StorageClass

Before you deploy Kubeflow, you must designate a default StorageClass within your Kubernetes cluster. The Kubeflow deployment process attempts to provision new persistent volumes using the default StorageClass. If no StorageClass is designated as the default StorageClass, then the deployment fails. To designate a default StorageClass within your cluster, perform the following task from the deployment jump host. If you have already designated a default StorageClass within your cluster, then you can skip this step.

1. Designate one of your existing StorageClasses as the default StorageClass. The example commands that follow show the designation of a StorageClass named `ontap-ai- flexvols-retain` as the default StorageClass.



The `ontap-nas-flexgroup` Trident Backend type has a minimum PVC size that is fairly large. By default, Kubeflow attempts to provision PVCs that are only a few GBs in size. Therefore, you should not designate a StorageClass that utilizes the `ontap-nas-flexgroup` Backend type as the default StorageClass for the purposes of Kubeflow deployment.

```

$ kubectl get sc
NAME                               PROVISIONER          AGE
ontap-ai-flexgroups-retain        csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface1 csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface2 csi.trident.netapp.io   25h
ontap-ai-flexvols-retain         csi.trident.netapp.io   3s
$ kubectl patch storageclass ontap-ai-flexvols-retain -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/ontap-ai-flexvols-retain patched
$ kubectl get sc
NAME                               PROVISIONER          AGE
ontap-ai-flexgroups-retain        csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface1 csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface2 csi.trident.netapp.io   25h
ontap-ai-flexvols-retain (default) csi.trident.netapp.io   54s

```

## Use NVIDIA DeepOps to Deploy Kubeflow

NetApp recommends using the Kubeflow deployment tool that is provided by NVIDIA DeepOps. To deploy Kubeflow in your Kubernetes cluster using the DeepOps deployment tool, perform the following tasks from the deployment jump host.



Alternatively, you can deploy Kubeflow manually by following the [installation instructions](#) in the official Kubeflow documentation

1. Deploy Kubeflow in your cluster by following the [Kubeflow deployment instructions](#) on the NVIDIA DeepOps GitHub site.
2. Note down the Kubeflow Dashboard URL that the DeepOps Kubeflow deployment tool outputs.

```

$ ./scripts/k8s/deploy_kubeflow.sh -x
...
INFO[0007] Applied the configuration Successfully!
filename="cmd/apply.go:72"
Kubeflow app installed to: /home/ai/kubeflow
It may take several minutes for all services to start. Run 'kubectl get
pods -n kubeflow' to verify
To remove (excluding CRDs, istio, auth, and cert-manager), run:
./scripts/k8s_deploy_kubeflow.sh -d
To perform a full uninstall : ./scripts/k8s_deploy_kubeflow.sh -D
Kubeflow Dashboard (HTTP NodePort): http://10.61.188.111:31380

```

3. Confirm that all pods deployed within the Kubeflow namespace show a **STATUS** of **Running** and confirm that no components deployed within the namespace are in an error state. It may take several minutes for all pods to start.

```
$ kubectl get all -n kubeflow
NAME                                         READY
STATUS    RESTARTS   AGE
pod/admission-webhook-bootstrap-stateful-set-0      1/1
Running   0          95s
pod/admission-webhook-deployment-6b89c84c98-vrtbh     1/1
Running   0          91s
pod/application-controller-stateful-set-0      1/1
Running   0          98s
pod/argo-ui-5dcf5d8b4f-m2wn4      1/1
Running   0          97s
pod/centraldashboard-cf4874ddc-7hcr8     1/1
Running   0          97s
pod/jupyter-web-app-deployment-685b455447-gjhh7     1/1
Running   0          96s
pod/katib-controller-88c97d85c-kgqq66     1/1
Running   1          95s
pod/katib-db-8598468fd8-5jw2c      1/1
Running   0          95s
pod/katib-manager-574c8c67f9-wtrf5      1/1
Running   1          95s
pod/katib-manager-rest-778857c989-fjbzn     1/1
Running   0          95s
pod/katib-suggestion-bayesianoptimization-65df4d7455-qthmw 1/1
Running   0          94s
pod/katib-suggestion-grid-56bf69f597-98vwn     1/1
Running   0          94s
pod/katib-suggestion-hyperband-7777b76cb9-9v6dq     1/1
Running   0          93s
pod/katib-suggestion-nasrl-77f6f9458c-2qzxq      1/1
Running   0          93s
pod/katib-suggestion-random-77b88b5c79-164j9      1/1
Running   0          93s
pod/katib-ui-7587c5b967-nd629      1/1
Running   0          95s
pod/metacontroller-0      1/1
Running   0          96s
pod/metadata-db-5dd459cc-swzkm     1/1
Running   0          94s
pod/metadata-deployment-6cf77db994-69fk7      1/1
Running   3          93s
pod/metadata-deployment-6cf77db994-mpbjt      1/1
Running   3          93s
pod/metadata-deployment-6cf77db994-xg7tz      1/1
Running   3          94s
pod/metadata-ui-78f5b59b56-qb6kr      1/1
```

Running	0	94s	
pod/minio-758b769d67-11vdr			1/1
Running	0	91s	
pod/ml-pipeline-5875b9db95-g8t2k			1/1
Running	0	91s	
pod/ml-pipeline-persistenceagent-9b69ddd46-bt9r9			1/1
Running	0	90s	
pod/ml-pipeline-scheduledworkflow-7b8d756c76-7x56s			1/1
Running	0	90s	
pod/ml-pipeline-ui-79ffd9c76-fcwpd			1/1
Running	0	90s	
pod/ml-pipeline-viewer-controller-deployment-5fdc87f58-b2t9r			1/1
Running	0	90s	
pod/mysql-657f87857d-15k9z			1/1
Running	0	91s	
pod/notebook-controller-deployment-56b4f59bbf-8bvnr			1/1
Running	0	92s	
pod/profiles-deployment-6bc745947-mrdkh			2/2
Running	0	90s	
pod/pytorch-operator-77c97f4879-hmlrv			1/1
Running	0	92s	
pod/seldon-operator-controller-manager-0			1/1
Running	1	91s	
pod/spartakus-volunteer-5fdfddb779-17qkm			1/1
Running	0	92s	
pod/tensorboard-6544748d94-nh8b2			1/1
Running	0	92s	
pod/tf-job-dashboard-56f79c59dd-6w59t			1/1
Running	0	92s	
pod/tf-job-operator-79cbfd6dbc-rb58c			1/1
Running	0	91s	
pod/workflow-controller-db644d554-cwrnb			1/1
Running	0	97s	
NAME			TYPE
CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/admission-webhook-service			ClusterIP
10.233.51.169	<none>	443/TCP	97s
service/application-controller-service			ClusterIP
10.233.4.54	<none>	443/TCP	98s
service/argo-ui			NodePort
10.233.47.191	<none>	80:31799/TCP	97s
service/centraldashboard			ClusterIP
10.233.8.36	<none>	80/TCP	97s
service/jupyter-web-app-service			ClusterIP
10.233.1.42	<none>	80/TCP	97s
service/katib-controller			ClusterIP

10.233.25.226	<none>	443/TCP	96s
service/katib-db			ClusterIP
10.233.33.151	<none>	3306/TCP	97s
service/katib-manager			ClusterIP
10.233.46.239	<none>	6789/TCP	96s
service/katib-manager-rest			ClusterIP
10.233.55.32	<none>	80/TCP	96s
service/katib-suggestion-bayesianoptimization			ClusterIP
10.233.49.191	<none>	6789/TCP	95s
service/katib-suggestion-grid			ClusterIP
10.233.9.105	<none>	6789/TCP	95s
service/katib-suggestion-hyperband			ClusterIP
10.233.22.2	<none>	6789/TCP	95s
service/katib-suggestion-nasrl			ClusterIP
10.233.63.73	<none>	6789/TCP	95s
service/katib-suggestion-random			ClusterIP
10.233.57.210	<none>	6789/TCP	95s
service/katib-ui			ClusterIP
10.233.6.116	<none>	80/TCP	96s
service/metadata-db			ClusterIP
10.233.31.2	<none>	3306/TCP	96s
service/metadata-service			ClusterIP
10.233.27.104	<none>	8080/TCP	96s
service/metadata-ui			ClusterIP
10.233.57.177	<none>	80/TCP	96s
service/minio-service			ClusterIP
10.233.44.90	<none>	9000/TCP	94s
service/ml-pipeline			ClusterIP
10.233.41.201	<none>	8888/TCP, 8887/TCP	94s
service/ml-pipeline-tensorboard-ui			ClusterIP
10.233.36.207	<none>	80/TCP	93s
service/ml-pipeline-ui			ClusterIP
10.233.61.150	<none>	80/TCP	93s
service/mysql			ClusterIP
10.233.55.117	<none>	3306/TCP	94s
service/notebook-controller-service			ClusterIP
10.233.10.166	<none>	443/TCP	95s
service/profiles-kfam			ClusterIP
10.233.33.79	<none>	8081/TCP	92s
service/pytorch-operator			ClusterIP
10.233.37.112	<none>	8443/TCP	95s
service/seldon-operator-controller-manager-service			ClusterIP
10.233.30.178	<none>	443/TCP	92s
service/tensorboard			ClusterIP
10.233.58.151	<none>	9000/TCP	94s
service/tf-job-dashboard			ClusterIP

10.233.4.17	<none>	80/TCP	94s	
service/tf-job-operator				ClusterIP
10.233.60.32	<none>	8443/TCP	94s	
service/webhook-server-service				ClusterIP
10.233.32.167	<none>	443/TCP	87s	
NAME				READY UP-
TO-DATE	AVAILABLE	AGE		
deployment.apps/admission-webhook-deployment			1/1	1
1	97s			
deployment.apps/argo-ui			1/1	1
1	97s			
deployment.apps/centraldashboard			1/1	1
1	97s			
deployment.apps/jupyter-web-app-deployment			1/1	1
1	97s			
deployment.apps/katib-controller			1/1	1
1	96s			
deployment.apps/katib-db			1/1	1
1	97s			
deployment.apps/katib-manager			1/1	1
1	96s			
deployment.apps/katib-manager-rest			1/1	1
1	96s			
deployment.apps/katib-suggestion-bayesianoptimization			1/1	1
1	95s			
deployment.apps/katib-suggestion-grid			1/1	1
1	95s			
deployment.apps/katib-suggestion-hyperband			1/1	1
1	95s			
deployment.apps/katib-suggestion-nasrl			1/1	1
1	95s			
deployment.apps/katib-suggestion-random			1/1	1
1	95s			
deployment.apps/katib-ui			1/1	1
1	96s			
deployment.apps/metadata-db			1/1	1
1	96s			
deployment.apps/metadata-deployment			3/3	3
3	96s			
deployment.apps/metadata-ui			1/1	1
1	96s			
deployment.apps/minio			1/1	1
1	94s			
deployment.apps/ml-pipeline			1/1	1
1	94s			
deployment.apps/ml-pipeline-persistenceagent			1/1	1

1	93s			
deployment.apps/ml-pipeline-scheduledworkflow		1/1	1	
1	93s			
deployment.apps/ml-pipeline-ui		1/1	1	
1	93s			
deployment.apps/ml-pipeline-viewer-controller-deployment		1/1	1	
1	93s			
deployment.apps/mysql		1/1	1	
1	94s			
deployment.apps/notebook-controller-deployment		1/1	1	
1	95s			
deployment.apps/profiles-deployment		1/1	1	
1	92s			
deployment.apps/pytorch-operator		1/1	1	
1	95s			
deployment.apps/spartakus-volunteer		1/1	1	
1	94s			
deployment.apps/tensorboard		1/1	1	
1	94s			
deployment.apps/tf-job-dashboard		1/1	1	
1	94s			
deployment.apps/tf-job-operator		1/1	1	
1	94s			
deployment.apps/workflow-controller		1/1	1	
1	97s			
NAME				
DESIRED	CURRENT	READY	AGE	
replicaset.apps/admission-webhook-deployment-6b89c84c98				1
1	1	97s		
replicaset.apps/argo-ui-5dcf5d8b4f				1
1	1	97s		
replicaset.apps/centraldashboard-cf4874ddc				1
1	1	97s		
replicaset.apps/jupyter-web-app-deployment-685b455447				1
1	1	97s		
replicaset.apps/katib-controller-88c97d85c				1
1	1	96s		
replicaset.apps/katib-db-8598468fd8				1
1	1	97s		
replicaset.apps/katib-manager-574c8c67f9				1
1	1	96s		
replicaset.apps/katib-manager-rest-778857c989				1
1	1	96s		
replicaset.apps/katib-suggestion-bayesianoptimization-65df4d7455				1
1	1	95s		
replicaset.apps/katib-suggestion-grid-56bf69f597				1

			READY	AGE
1	1	95s		
replicaset.apps/katib-suggestion-hyperband-7777b76cb9				1
1	1	95s		
replicaset.apps/katib-suggestion-nasrl-77f6f9458c				1
1	1	95s		
replicaset.apps/katib-suggestion-random-77b88b5c79				1
1	1	95s		
replicaset.apps/katib-ui-7587c5b967				1
1	1	96s		
replicaset.apps/metadata-db-5dd459cc				1
1	1	96s		
replicaset.apps/metadata-deployment-6cf77db994				3
3	3	96s		
replicaset.apps/metadata-ui-78f5b59b56				1
1	1	96s		
replicaset.apps/minio-758b769d67				1
1	1	93s		
replicaset.apps/ml-pipeline-5875b9db95				1
1	1	93s		
replicaset.apps/ml-pipeline-persistenceagent-9b69ddd46				1
1	1	92s		
replicaset.apps/ml-pipeline-scheduledworkflow-7b8d756c76				1
1	1	91s		
replicaset.apps/ml-pipeline-ui-79ffd9c76				1
1	1	91s		
replicaset.apps/ml-pipeline-viewer-controller-deployment-5fdc87f58				1
1	1	91s		
replicaset.apps/mysql-657f87857d				1
1	1	92s		
replicaset.apps/notebook-controller-deployment-56b4f59bbf				1
1	1	94s		
replicaset.apps/profiles-deployment-6bc745947				1
1	1	91s		
replicaset.apps/pytorch-operator-77c97f4879				1
1	1	94s		
replicaset.apps/spartakus-volunteer-5fdfddb779				1
1	1	94s		
replicaset.apps/tensorboard-6544748d94				1
1	1	93s		
replicaset.apps/tf-job-dashboard-56f79c59dd				1
1	1	93s		
replicaset.apps/tf-job-operator-79cbfd6dbc				1
1	1	93s		
replicaset.apps/workflow-controller-db644d554				1
1	1	97s		

```
statefulset.apps/admission-webhook-bootstrap-stateful-set    1/1    97s
statefulset.apps/application-controller-stateful-set        1/1    98s
statefulset.apps/metacontroller                          1/1    98s
statefulset.apps/seldon-operator-controller-manager      1/1    92s
$ kubectl get pvc -n kubeflow
NAME              STATUS    VOLUME
CAPACITY  ACCESS MODES  STORAGECLASS          AGE
katib-mysql     RWO      Bound    pvc-b07f293e-d028-11e9-9b9d-00505681a82d
10Gi           RWO      Bound    ontap-ai-flexvols-retain   27m
metadata-mysql  RWO      Bound    pvc-b0f3f032-d028-11e9-9b9d-00505681a82d
10Gi           RWO      Bound    ontap-ai-flexvols-retain   27m
minio-pv-claim  RWO      Bound    pvc-b22727ee-d028-11e9-9b9d-00505681a82d
20Gi           RWO      Bound    ontap-ai-flexvols-retain   27m
mysql-pv-claim  RWO      Bound    pvc-b2429afcd028-11e9-9b9d-00505681a82d
20Gi           RWO      Bound    ontap-ai-flexvols-retain   27m
```

4. In your web browser, access the Kubeflow central dashboard by navigating to the URL that you noted down in step 2.

The default username is [admin@kubeflow.org](mailto:admin@kubeflow.org), and the default password is [12341234](#). To create additional users, follow the instructions in the [official Kubeflow documentation](#).

The screenshot shows the Kubeflow Central Dashboard interface. On the left, a dark sidebar lists navigation options: Home, Pipelines, Notebook Servers, Katib, Artifact Store, GitHub, and Documentation. The main content area has tabs for 'Dashboard' (selected) and 'Activity'. The 'Dashboard' section contains three panels: 'Quick shortcuts' (with links to upload a pipeline, view pipeline runs, create a notebook server, view Katib studies, and view metadata artifacts), 'Recent Notebooks' (prompting to choose a namespace), and 'Recent Pipelines' (listing five sample pipelines: Basic - Exit Handler, Basic - Conditional execution, Basic - Parallel execution, Basic - Sequential execution, and ML - TFX - Taxi Tip Prediction...). The 'Documentation' panel on the right lists several guides with icons: 'Getting Started with Kubeflow', 'MiniKF', 'MicrOK8s for Kubeflow', 'Minikube for Kubeflow', 'Kubeflow on GCP', 'Kubeflow on AWS', and 'Requirements for Kubeflow'. At the bottom left of the main area, there's a link to 'Privacy + Usage Reporting build version v0.6.2'.

[Next: Example Kubeflow Operations and Tasks](#)

## Example Kubeflow Operations and Tasks

This section includes examples of various operations and tasks that you may want to perform using Kubeflow.

[Next: Provision a Jupyter Notebook Workspace for Data Scientist or Developer Use](#)

## Example Kubeflow Operations and Tasks

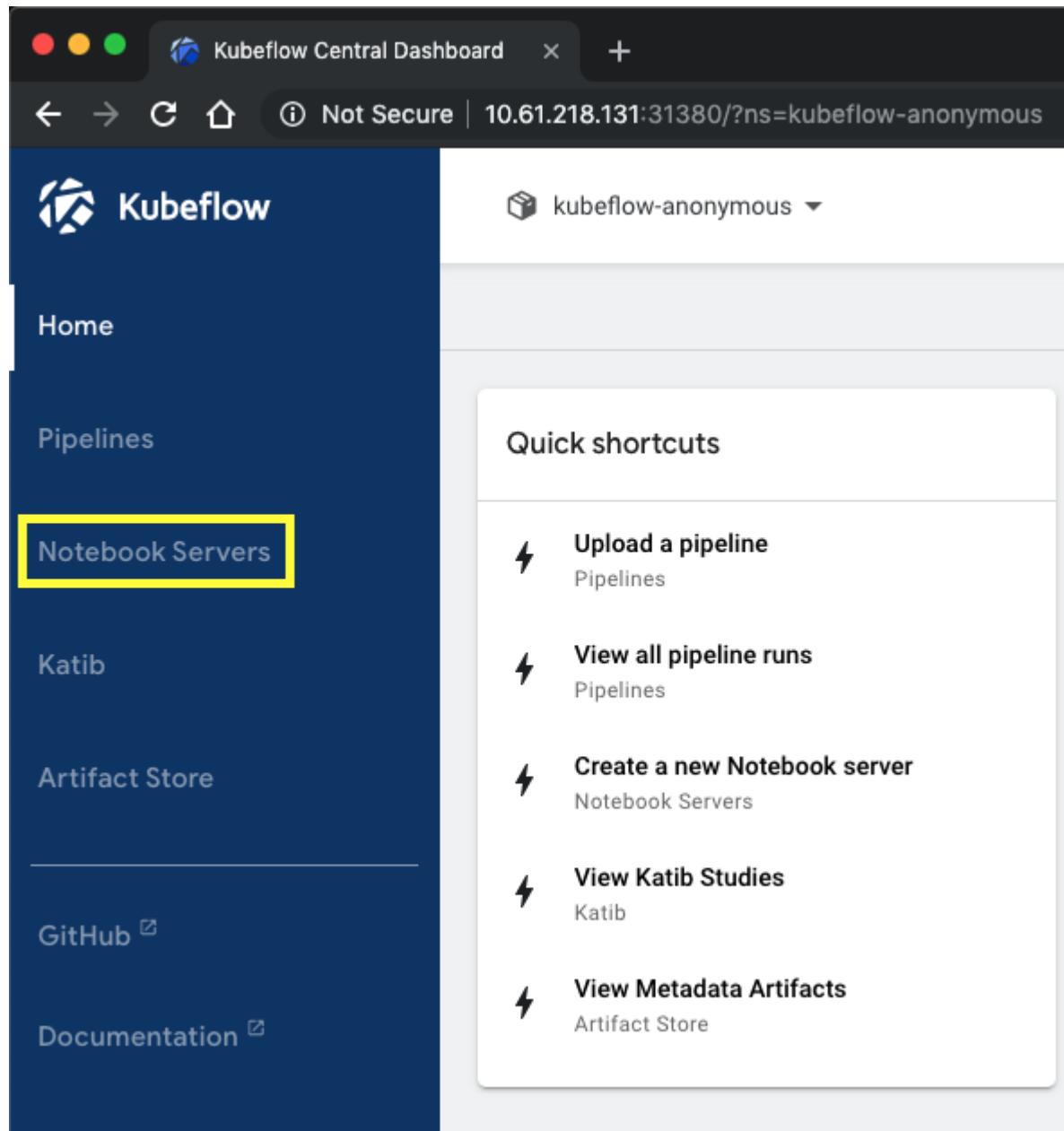
This section includes examples of various operations and tasks that you may want to perform using Kubeflow.

[Next: Provision a Jupyter Notebook Workspace for Data Scientist or Developer Use](#)

## Provision a Jupyter Notebook Workspace for Data Scientist or Developer Use

Kubeflow is capable of rapidly provisioning new Jupyter Notebook servers to act as data scientist workspaces. To provision a new Jupyter Notebook server with Kubeflow, perform the following tasks. For more information about Jupyter Notebooks within the Kubeflow context, see the [official Kubeflow documentation](#).

1. From the Kubeflow central dashboard, click Notebook Servers in the main menu to navigate to the Jupyter Notebook server administration page.



2. Click New Server to provision a new Jupyter Notebook server.

The screenshot shows a web browser window titled "Kubeflow Central Dashboard". The URL is "Not Secure | 10.61.218.131:31380/\_jupyter/?ns=kubeflow-anonymous". The page header includes the Kubeflow logo and the namespace "kubeflow-anonymous". The main content area is titled "Notebook Servers" and features a table with columns: Status, Name, Age, Image, CPU, Memory, and Volumes. A yellow-bordered button labeled "+ NEW SERVER" is located in the top right corner of the table header.

3. Give your new server a name, choose the Docker image that you want your server to be based on, and specify the amount of CPU and RAM to be reserved by your server. If the Namespace field is blank, use the Select Namespace menu in the page header to choose a namespace. The Namespace field is then auto-populated with the chosen namespace.

In the following example, the `kubeflow-anonymous` namespace is chosen. In addition, the default values for Docker image, CPU, and RAM are accepted.

The screenshot shows the Kubeflow Central Dashboard interface for creating a new Notebook Server. The browser title is "Kubeflow Central Dashboard". The URL is "Not Secure | 10.61.218.131:31380/\_/jupyter/?ns=kubeflow-anonym...". The dashboard has a header with the Kubeflow logo and a user profile icon.

**Name**

Specify the name of the Notebook Server and the Namespace it will belong to.

Name: mike

Namespace: kubeflow-anonymous

**Image**

A starter Jupyter Docker Image with a baseline deployment and typical ML packages.

Custom Image

Image: gcr.io/kubeflow-images-public/tensorflow-1.13.1-notebook-cpu:v0.5.0

**CPU / RAM**

Specify the total amount of CPU and RAM reserved by your Notebook Server. For CPU-intensive workloads, you can choose more than 1 CPU (e.g. 1.5).

CPU: 0.5

Memory: 1.0Gi

**Workspace Volume**

Configure the Volume to be mounted as your personal Workspace.

Don't use Persistent Storage for User's home

Type: New

Name: workspace-mike

Size: 10Gi

Mode: ReadWriteOnce

Mount Point: /home/jovyan

4. Specify the workspace volume details. If you choose to create a new volume, then that volume or PVC is provisioned using the default StorageClass. Because a StorageClass utilizing Trident was designated as the default StorageClass in the section [Kubeflow Deployment](#), the volume or PVC is provisioned with Trident. This volume is automatically mounted as the default workspace within the Jupyter Notebook Server container. Any notebooks that a user creates on the server that are not saved to a separate data volume are automatically saved to this workspace volume. Therefore, the notebooks are persistent across reboots.

5. Add data volumes. The following example specifies an existing PVC named 'pb-fg-all' and accepts the default mount point.

## Data Volumes

Configure the Volumes to be mounted as your Datasets.

[+ ADD VOLUME](#)

Type

Existing

Name

pb-fg-all

Size

10Gi

Mode

ReadWriteOnce

Mount Point

/home/jovyan/data-vol-1



6. **Optional:** Request that the desired number of GPUs be allocated to your notebook server. In the following example, one GPU is requested.

## Configurations

Extra layers of configurations that will be applied to the new Notebook. (e.g. Insert credentials as Secrets, set Environment Variables.)

Configurations



## Extra Resources

Specify extra resources that might be needed in the Notebook Server.

Enable Shared Memory

Extra Resources \*

{"nvidia.com/gpu": 1}

Extra Resources available in the cluster (ex. NVIDIA GPUs)

[LAUNCH](#)

[CANCEL](#)

7. Click Launch to provision your new notebook server.

8. Wait for your notebook server to be fully provisioned. This can take several minutes if you have never provisioned a server using the Docker image that you specified because the image needs to be downloaded. When your server has been fully provisioned, you see a green check mark in the Status column on the Jupyter Notebook server administration page.

The screenshot shows the Kubeflow Central Dashboard interface. At the top, there's a header bar with the title "Kubeflow Central Dashboard". Below it is a browser-style address bar showing the URL "Not Secure | 10.61.218.131:31380/\_jupyter/?ns=kubeflow-anonym...". The main content area is titled "Notebook Servers" and contains a table with columns: Status, Name, Age, Image, CPU, Memory, Volumes, and Actions. One row is highlighted with a yellow box around its "Status" column icon. The "Actions" column includes a "CONNECT" button and a trash bin icon.

9. Click Connect to connect to your new server web interface.
10. Confirm that the dataset volume that was specified in step 6 is mounted on the server. Note that this volume is mounted within the default workspace by default. From the perspective of the user, this is just another folder within the workspace. The user, who is likely a data scientist and not an infrastructure expert, does not need to possess any storage expertise in order to use this volume.

The screenshot shows the Jupyter notebook interface. The title bar says "Home" and the address bar shows the URL "Not Secure | 10.61.218.131:31380/notebook/kubeflow-anonymous...". The main area has tabs for "Files", "Running", and "Clusters". Below the tabs, there's a message "Select items to perform actions on them." and a toolbar with "Upload", "New", and other icons. A file list table shows a single item: a folder named "data-vol-1" which is highlighted with a yellow box. The table also includes columns for "Name", "Last Modified", and "File size".

The screenshot shows the Jupyter Notebook interface with a file browser window titled 'data-vol-1/'. The browser lists several files and folders under the root directory. A context menu is open over the 'banking.csv' file, showing options for creating a new notebook in Python 2 or 3, a text file, a folder, or opening a terminal. The 'Terminal' option is highlighted.

Name	Last Modified	File size
..	seconds ago	
blas_folder	2 months ago	
collected_trace	2 months ago	
container	3 months ago	
dataset	5 hours ago	
fio_test	3 months ago	
parabricks	7 months ago	
banking.csv	a month ago	4.88 MB

11. Open a Terminal and, assuming that a new volume was requested in step 5, execute `df -h` to confirm that a new Trident-provisioned persistent volume is mounted as the default workspace.

The default workspace directory is the base directory that you are presented with when you first access the server's web interface. Therefore, any artifacts that you create by using the web interface are stored on this Trident-provisioned persistent volume.

The screenshot shows the Jupyter Notebook interface with a file browser window titled 'data-vol-1/'. The browser lists several files and folders under the root directory. A context menu is open over the 'banking.csv' file, showing options for creating a new notebook in Python 2 or 3, a text file, a folder, or opening a terminal. The 'Terminal' option is highlighted.

Name	Last Modified	File size
..	seconds ago	
blas_folder	2 months ago	
collected_trace	2 months ago	
container	3 months ago	
dataset	5 hours ago	
fio_test	3 months ago	
parabricks	7 months ago	
banking.csv	a month ago	4.88 MB

```
$ df -h
Filesystem              Size  Used Avail
overlay                  439G  34G  382G
9% /
tmpfs                   64M   0    64M
0% /dev
tmpfs                   252G   0   252G
0% /sys/fs/cgroup
/dev/sda2                439G  34G  382G
9% /etc/hosts
192.168.11.11:/trident_pvc_3dcfe7e5_d5a9_11e9_9b9d_00505681a82d  10G  320K  10G
1% /home/jovyan
tmpfs                   252G   0   252G
0% /dev/shm
192.168.11.11:/pb_fg_all          10T  10T  47G
100% /home/jovyan/data-vol-1
tmpfs                   252G  12K  252G
1% /run/secrets/kubernetes.io/serviceaccount
tmpfs                   252G  12K  252G
1% /proc/driver/nvidia
tmpfs                   51G  4.9M  51G
1% /run/nvidia-persistenced/socket
udev                    252G   0   252G
0% /dev/nvidia5
tmpfs                   252G   0   252G
0% /proc/acpi
tmpfs                   252G   0   252G
0% /proc/scsi
tmpfs                   252G   0   252G
0% /sys/firmware
$
```

12. Using the terminal, run `nvidia-smi` to confirm that the correct number of GPUs were allocated to the notebook server. In the following example, one GPU has been allocated to the notebook server as requested in step 7.

```
$ nvidia-smi
Fri Sep 13 13:52:15 2019
+-----+
| NVIDIA-SMI 410.104      Driver Version: 410.104      CUDA Version: N/A |
+-----+
| GPU  Name     Persistence-M | Bus-Id     Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|-----+
| 0  Tesla V100-SXM2...  On  | 00000000:86:00.0 Off |          0 |
| N/A  38C    P0    46W / 300W |      0MiB / 32480MiB |     0%      Default |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU  PID  Type  Process name        Usage        |
|-----+
| No running processes found            |
+-----+
$
```

Next: Example Notebooks and Pipelines

## Example Notebooks and Pipelines

The [NetApp Data Science Toolkit for Kubernetes](#) can be used in conjunction with Kubeflow. Using the NetApp Data Science Toolkit with Kubeflow provides the following benefits:

- Data scientists can perform advanced NetApp data management operations directly from within a Jupyter Notebook.
- Advanced NetApp data management operations can be incorporated into automated workflows using the Kubeflow Pipelines framework.

Refer to the [Kubeflow Examples](#) section within the NetApp Data Science Toolkit GitHub repository for details on using the toolkit with Kubeflow.

Next: [Apache Airflow Deployment](#)

## Apache Airflow Deployment

NetApp recommends running Apache Airflow on top of Kubernetes. This section describes the tasks that you must complete to deploy Airflow in your Kubernetes cluster.



It is possible to deploy Airflow on platforms other than Kubernetes. Deploying Airflow on platforms other than Kubernetes is outside of the scope of this solution.

### Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You already have a working Kubernetes cluster.
2. You have already installed and configured NetApp Trident in your Kubernetes cluster as outlined in the section “[NetApp Trident Deployment and Configuration](#).”

### Install Helm

Airflow is deployed using Helm, a popular package manager for Kubernetes. Before you deploy Airflow, you must install Helm on the deployment jump host. To install Helm on the deployment jump host, follow the [installation instructions](#) in the official Helm documentation.

### Set Default Kubernetes StorageClass

Before you deploy Airflow, you must designate a default StorageClass within your Kubernetes cluster. The Airflow deployment process attempts to provision new persistent volumes using the default StorageClass. If no StorageClass is designated as the default StorageClass, then the deployment fails. To designate a default StorageClass within your cluster, follow the instructions outlined in the section [Kubeflow Deployment](#). If you have already designated a default StorageClass within your cluster, then you can skip this step.

### Use Helm to Deploy Airflow

To deploy Airflow in your Kubernetes cluster using Helm, perform the following tasks from the deployment jump host:

1. Deploy Airflow using Helm by following the [deployment instructions](#) for the official Airflow chart on the Artifact Hub. The example commands that follow show the deployment of Airflow using Helm. Modify, add, and/or remove values in the `custom-values.yaml` file as needed depending on your environment and desired configuration.

```
$ cat << EOF > custom-values.yaml
#####
# Airflow - Common Configs
#####
airflow:
    ## the airflow executor type to use
    ##
    executor: "CeleryExecutor"
    ## environment variables for the web/scheduler/worker Pods (for
    airflow configs)
    ##
    #
#####
# Airflow - WebUI Configs
#####
web:
    ## configs for the Service of the web Pods
    ##
    service:
        type: NodePort
#####
# Airflow - Logs Configs
#####
logs:
    persistence:
        enabled: true
#####
# Airflow - DAGs Configs
#####
dags:
    ## configs for the DAG git repository & sync container
    ##
    gitSync:
        enabled: true
        ## url of the git repository
        ##
        repo: "git@github.com:mboglesby/airflow-dev.git"
        ## the branch/tag/sha1 which we clone
        ##
        branch: master
        revision: HEAD
```

```

## the name of a pre-created secret containing files for ~/.ssh/
##
## NOTE:
## - this is ONLY RELEVANT for SSH git repos
## - the secret commonly includes files: id_rsa, id_rsa.pub,
known_hosts
## - known_hosts is NOT NEEDED if `git.sshKeyscan` is true
##
sshSecret: "airflow-ssh-git-secret"
## the name of the private key file in your `git.secret`
##
## NOTE:
## - this is ONLY RELEVANT for PRIVATE SSH git repos
##
sshSecretKey: id_rsa
## the git sync interval in seconds
##
syncWait: 60
EOF
$ helm install airflow airflow-stable/airflow -n airflow --version 8.0.8
--values ./custom-values.yaml
...
Congratulations. You have just deployed Apache Airflow!
1. Get the Airflow Service URL by running these commands:
   export NODE_PORT=$(kubectl get --namespace airflow -o
   jsonpath=".spec.ports[0].nodePort" services airflow-web)
   export NODE_IP=$(kubectl get nodes --namespace airflow -o
   jsonpath=".items[0].status.addresses[0].address")
   echo http://$NODE_IP:$NODE_PORT/
2. Open Airflow in your web browser

```

2. Confirm that all Airflow pods are up and running. It may take a few minutes for all pods to start.

NAME	READY	STATUS	RESTARTS	AGE
airflow-flower-b5656d44f-h8qjk	1/1	Running	0	2h
airflow-postgresql-0	1/1	Running	0	2h
airflow-redis-master-0	1/1	Running	0	2h
airflow-scheduler-9d95fcdf9-clf4b	2/2	Running	2	2h
airflow-web-59c94db9c5-z7rg4	1/1	Running	0	2h
airflow-worker-0	2/2	Running	2	2h

3. Obtain the Airflow web service URL by following the instructions that were printed to the console when you deployed Airflow using Helm in step 1.

```
$ export NODE_PORT=$(kubectl get --namespace airflow -o jsonpath=".spec.ports[0].nodePort" services airflow-web)
$ export NODE_IP=$(kubectl get nodes --namespace airflow -o jsonpath=".items[0].status.addresses[0].address")
$ echo http://$NODE_IP:$NODE_PORT/
```

#### 4. Confirm that you can access the Airflow web service.

	<b>DAG</b>	<b>Schedule</b>	<b>Owner</b>	<b>Recent Tasks</b>	<b>Last Run</b>	<b>DAG Runs</b>	<b>Links</b>
<input checked="" type="checkbox"/>	Off ai_training_run	None	NetApp	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off create_data_scientist_workspace	None	NetApp	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_bash_operator	0 0 * * *	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_branch_dop_operator_v3	*/1 * * * *	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_branch_operator	@daily	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_complex	None	airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_external_task_marker_child	None	airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_external_task_marker_parent	None	airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_http_operator	1 day, 0:00:00	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_kubernetes_executor_config	None	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_nested_branch_dag	@daily	airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_passing_params_via_test_command	*/1 * * * *	airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_pig_operator	None	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_python_operator	None	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_short_circuit_operator	1 day, 0:00:00	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●
<input checked="" type="checkbox"/>	Off example_skip_dag	1 day, 0:00:00	Airflow	○○○○○○○○○○○○		○○○○	○●●●●●●●●●●●●

Next: [Example Apache Airflow Workflows](#)

### Example Apache Airflow Workflows

The [NetApp Data Science Toolkit for Kubernetes](#) can be used in conjunction with Airflow. Using the NetApp Data Science Toolkit with Airflow enables you to incorporate NetApp data management operations into automated workflows that are orchestrated by Airflow.

Refer to the [Airflow Examples](#) section within the NetApp Data Science Toolkit GitHub repository for details on using the toolkit with Airflow.

Next: Example Trident Operations

## Example Trident Operations

This section includes examples of various operations that you may want to perform with Trident.

### Import an Existing Volume

If there are existing volumes on your NetApp storage system/platform that you want to mount on containers within your Kubernetes cluster, but that are not tied to PVCs in the cluster, then you must import these volumes. You can use the Trident volume import functionality to import these volumes.

The example commands that follow show the importing of the same volume, named `pb_fg_all`, twice, once for each Trident Backend that was created in the example in the section [Example Trident Backends for ONTAP AI Deployments](#), step 1. Importing the same volume twice in this manner enables you to mount the volume (an existing FlexGroup volume) multiple times across different LIFs, as described in the section [Example Trident Backends for ONTAP AI Deployments](#), step 1. For more information about PVCs, see the [official Kubernetes documentation](#). For more information about the volume import functionality, see the [Trident documentation](#).

An `accessModes` value of `ReadOnlyMany` is specified in the example PVC spec files. For more information about the `accessMode` field, see the [official Kubernetes documentation](#).



The Backend names that are specified in the following example import commands correspond to the Backends that were created in the example in the section [Example Trident Backends for ONTAP AI Deployments](#), step 1. The StorageClass names that are specified in the following example PVC definition files correspond to the StorageClasses that were created in the example in the section [Example Kubernetes StorageClasses for ONTAP AI Deployments](#), step 1.

```
$ cat << EOF > ./pvc-import-pb_fg_all-iface1.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface1
  namespace: default
spec:
  accessModes:
    - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface1
EOF
$ tridentctl import volume ontap-ai-flexgroups-iface1 pb_fg_all -f ./pvc-import-pb_fg_all-iface1.yaml -n trident
+-----+-----+
+-----+-----+
+-----+-----+
|           NAME           |   SIZE   |      STORAGE CLASS
| PROTOCOL |           BACKEND UUID           | STATE   |
MANAGED |
```

```

+-----+
+-----+
+-----+-----+-----+
| default-pb-fg-all-iface1-7d9f1 | 10 TiB | ontap-ai-flexgroups-retain-
iface1 | file      | b74cbddb-e0b8-40b7-b263-b6da6dec0bdd | online | true
|
+-----+-----+
+-----+
+-----+-----+-----+
$ cat << EOF > ./pvc-import-pb_fg_all-iface2.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface2
  namespace: default
spec:
  accessModes:
    - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface2
EOF
$ tridentctl import volume ontap-ai-flexgroups-iface2 pb_fg_all -f ./pvc-
import-pb_fg_all-iface2.yaml -n trident
+-----+
+-----+-----+
|           NAME          |   SIZE   |        STORAGE CLASS
| PROTOCOL |           BACKEND UUID            | STATE |
MANAGED |
+-----+
+-----+-----+
+-----+-----+-----+
| default-pb-fg-all-iface2-85aee | 10 TiB | ontap-ai-flexgroups-retain-
iface2 | file      | 61814d48-c770-436b-9cb4-cf7ee661274d | online | true
|
+-----+
+-----+-----+
+-----+-----+-----+
$ tridentctl get volume -n trident
+-----+
+-----+-----+
+-----+-----+-----+
|           NAME          |   SIZE   |        STORAGE CLASS
| PROTOCOL |           BACKEND UUID            | STATE | MANAGED |
+-----+
+-----+-----+
+-----+-----+-----+

```

```

| default-pb-fg-all-iface1-7d9f1 | 10 TiB | ontap-ai-flexgroups-retain-
iface1 | file      | b74cbddb-e0b8-40b7-b263-b6da6dec0bdd | online | true
|
| default-pb-fg-all-iface2-85aee | 10 TiB | ontap-ai-flexgroups-retain-
iface2 | file      | 61814d48-c770-436b-9cb4-cf7ee661274d | online | true
|
+-----+-----+
+-----+-----+
+-----+-----+-----+
$ kubectl get pvc
NAME           STATUS    VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS
pb-fg-all-iface1   Bound    default-pb-fg-all-iface1-7d9f1
10995116277760   ROX      ontap-ai-flexgroups-retain-iface1   25h
pb-fg-all-iface2   Bound    default-pb-fg-all-iface2-85aee
10995116277760   ROX      ontap-ai-flexgroups-retain-iface2   25h

```

## Provision a New Volume

You can use Trident to provision a new volume on your NetApp storage system or platform. The following example commands show the provisioning of a new FlexVol volume. In this example, the volume is provisioned using the StorageClass that was created in the example in the section [Example Kubernetes StorageClasses for ONTAP AI Deployments](#), step 2.

An `accessModes` value of `ReadWriteMany` is specified in the following example PVC definition file. For more information about the `accessMode` field, see the [official Kubernetes documentation](#).

```

$ cat << EOF > ./pvc-tensorflow-results.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: tensorflow-results
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-ai-flexvols-retain
EOF
$ kubectl create -f ./pvc-tensorflow-results.yaml
persistentvolumeclaim/tensorflow-results created
$ kubectl get pvc
NAME                      STATUS        VOLUME
CAPACITY      ACCESS MODES   STORAGECLASS          AGE
pb-fg-all-iface1           Bound       default-pb-fg-all-iface1-7d9f1
10995116277760   ROX          ontap-ai-flexgroups-retain-iface1 26h
pb-fg-all-iface2           Bound       default-pb-fg-all-iface2-85aee
10995116277760   ROX          ontap-ai-flexgroups-retain-iface2 26h
tensorflow-results         Bound       default-tensorflow-results-
2fd60      1073741824     RWX          ontap-ai-flexvols-retain
25h

```

[Next: Example High-Performance Jobs for ONTAP AI Deployments Overview](#)

### Example High-performance Jobs for ONTAP AI Deployments

This section includes examples of various high-performance jobs that can be executed when Kubernetes is deployed on an ONTAP AI pod.

[Next: Execute a Single-Node AI Workload](#)

### Example High-performance Jobs for ONTAP AI Deployments

This section includes examples of various high-performance jobs that can be executed when Kubernetes is deployed on an ONTAP AI pod.

[Next: Execute a Single-Node AI Workload](#)

### Execute a Single-Node AI Workload

To execute a single-node AI and ML job in your Kubernetes cluster, perform the following tasks from the deployment jump host. With Trident, you can quickly and easily make a data volume, potentially containing petabytes of data, accessible to a Kubernetes

workload. To make such a data volume accessible from within a Kubernetes pod, simply specify a PVC in the pod definition. This step is a Kubernetes-native operation; no NetApp expertise is required.



This section assumes that you have already containerized (in the Docker container format) the specific AI and ML workload that you are attempting to execute in your Kubernetes cluster.

1. The following example commands show the creation of a Kubernetes job for a TensorFlow benchmark workload that uses the ImageNet dataset. For more information about the ImageNet dataset, see the [ImageNet website](#).

This example job requests eight GPUs and therefore can run on a single GPU worker node that features eight or more GPUs. This example job could be submitted in a cluster for which a worker node featuring eight or more GPUs is not present or is currently occupied with another workload. If so, then the job remains in a pending state until such a worker node becomes available.

Additionally, in order to maximize storage bandwidth, the volume that contains the needed training data is mounted twice within the pod that this job creates. Another volume is also mounted in the pod. This second volume will be used to store results and metrics. These volumes are referenced in the job definition by using the names of the PVCs. For more information about Kubernetes jobs, see the [official Kubernetes documentation](#).

An `emptyDir` volume with a `medium` value of `Memory` is mounted to `/dev/shm` in the pod that this example job creates. The default size of the `/dev/shm` virtual volume that is automatically created by the Docker container runtime can sometimes be insufficient for TensorFlow's needs. Mounting an `emptyDir` volume as in the following example provides a sufficiently large `/dev/shm` virtual volume. For more information about `emptyDir` volumes, see the [official Kubernetes documentation](#).

The single container that is specified in this example job definition is given a `securityContext > privileged` value of `true`. This value means that the container effectively has root access on the host. This annotation is used in this case because the specific workload that is being executed requires root access. Specifically, a clear cache operation that the workload performs requires root access. Whether or not this `privileged: true` annotation is necessary depends on the requirements of the specific workload that you are executing.

```
$ cat << EOF > ./netapp-tensorflow-single-imagenet.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-single-imagenet
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
        - name: dshm
          emptyDir:
            medium: Memory
        - name: testdata-iface1
EOF
```

```

    persistentVolumeClaim:
        claimName: pb-fg-all-iface1
    - name: testdata-iface2
        persistentVolumeClaim:
            claimName: pb-fg-all-iface2
    - name: results
        persistentVolumeClaim:
            claimName: tensorflow-results
    containers:
    - name: netapp-tensorflow-py2
        image: netapp/tensorflow-py2:19.03.0
        command: ["python", "/netapp/scripts/run.py", "--dataset_dir=/mnt/mount_0/dataset/imagenet", "--dgx_version=dgx1", "--num_devices=8"]
        resources:
            limits:
                nvidia.com/gpu: 8
    volumeMounts:
    - mountPath: /dev/shm
        name: dshm
    - mountPath: /mnt/mount_0
        name: testdata-iface1
    - mountPath: /mnt/mount_1
        name: testdata-iface2
    - mountPath: /tmp
        name: results
    securityContext:
        privileged: true
    restartPolicy: Never
EOF
$ kubectl create -f ./netapp-tensorflow-single-imagenet.yaml
job.batch/netapp-tensorflow-single-imagenet created
$ kubectl get jobs
NAME                               COMPLETIONS   DURATION   AGE
netapp-tensorflow-single-imagenet   0/1          24s        24s

```

2. Confirm that the job that you created in step 1 is running correctly. The following example command confirms that a single pod was created for the job, as specified in the job definition, and that this pod is currently running on one of the GPU worker nodes.

```
$ kubectl get pods -o wide
```

NAME	READY	STATUS
RESTARTS	AGE	
IP	NODE	NOMINATED NODE
netapp-tensorflow-single-imagenet-m7x92	1/1	Running 0
3m 10.233.68.61	10.61.218.154	<none>

3. Confirm that the job that you created in step 1 completes successfully. The following example commands confirm that the job completed successfully.

```

$ kubectl get jobs
NAME                                COMPLETIONS   DURATION
AGE
netapp-tensorflow-single-imagenet      1/1          5m42s
10m

$ kubectl get pods
NAME                                READY   STATUS
RESTARTS   AGE
netapp-tensorflow-single-imagenet-m7x92   0/1    Completed
0          11m

$ kubectl logs netapp-tensorflow-single-imagenet-m7x92
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO-
PERMISSIONS in file gds_dstore.c at line 702
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO-
PERMISSIONS in file gds_dstore.c at line 711
Total images/sec = 6530.59125
===== Clean Cache !!! =====
mpirun -allow-run-as-root -np 1 -H localhost:1 bash -c 'sync; echo 1 >
/proc/sys/vm/drop_caches'
=====
mpirun -allow-run-as-root -np 8 -H localhost:8 -bind-to none -map-by
slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH python
/netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_be
nchmarks.py --model=resnet50 --batch_size=256 --device=gpu
--force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48
--variable_update=horovod --batch_group_size=20 --num_batches=500
--nodistortions --num_gpus=1 --data_format=NCHW --use_fp16=True
--use_tf_layers=False --data_name=imagenet --use_datasets=True
--data_dir=/mnt/mount_0/dataset/imagenet
--datasets_parallel_interleave_cycle_length=10
--datasets_sloppy_parallel_interleave=False --num_mounts=2
--mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000
--datasets_use_prefetch=True --datasets_num_private_threads=4
--horovod_device=gpu >
/tmp/20190814_105450_tensorflow_horovod_rdma_resnet50_gpu_8_256_b500_im
genet_nodistort_fp16_r10_m2_nockpt.txt 2>&1

```

4. **Optional:** Clean up job artifacts. The following example commands show the deletion of the job object that was created in step 1.

When you delete the job object, Kubernetes automatically deletes any associated pods.

```

$ kubectl get jobs
NAME                                COMPLETIONS   DURATION
AGE
netapp-tensorflow-single-imagenet      1/1          5m42s
10m

$ kubectl get pods
NAME                                READY   STATUS
RESTARTS   AGE
netapp-tensorflow-single-imagenet-m7x92  0/1    Completed
0          11m

$ kubectl delete job netapp-tensorflow-single-imagenet
job.batch "netapp-tensorflow-single-imagenet" deleted

$ kubectl get jobs
No resources found.

$ kubectl get pods
No resources found.

```

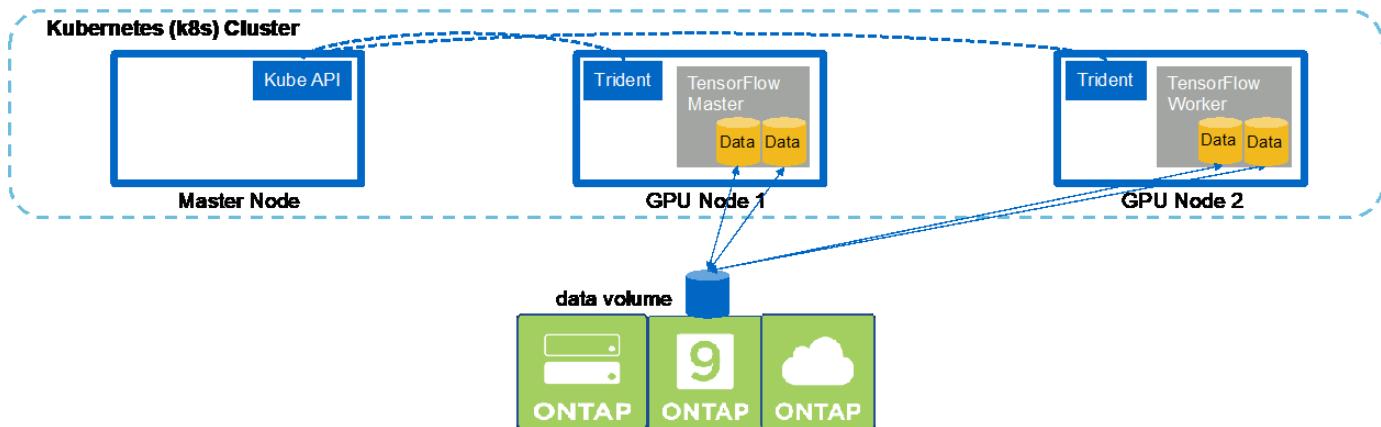
## Next: Execute a Synchronous Distributed AI Workload

### Execute a Synchronous Distributed AI Workload

To execute a synchronous multinode AI and ML job in your Kubernetes cluster, perform the following tasks on the deployment jump host. This process enables you to take advantage of data that is stored on a NetApp volume and to use more GPUs than a single worker node can provide. See the following figure for a depiction of a synchronous distributed AI job.



Synchronous distributed jobs can help increase performance and training accuracy compared with asynchronous distributed jobs. A discussion of the pros and cons of synchronous jobs versus asynchronous jobs is outside the scope of this document.



1. The following example commands show the creation of one worker that participates in the synchronous distributed execution of the same TensorFlow benchmark job that was executed on a single node in the example in the section [Execute a Single-Node AI Workload](#). In this specific example, only a single worker

is deployed because the job is executed across two worker nodes.

This example worker deployment requests eight GPUs and thus can run on a single GPU worker node that features eight or more GPUs. If your GPU worker nodes feature more than eight GPUs, to maximize performance, you might want to increase this number to be equal to the number of GPUs that your worker nodes feature. For more information about Kubernetes deployments, see the [official Kubernetes documentation](#).

A Kubernetes deployment is created in this example because this specific containerized worker would never complete on its own. Therefore, it doesn't make sense to deploy it by using the Kubernetes job construct. If your worker is designed or written to complete on its own, then it might make sense to use the job construct to deploy your worker.

The pod that is specified in this example deployment specification is given a `hostNetwork` value of `true`. This value means that the pod uses the host worker node's networking stack instead of the virtual networking stack that Kubernetes usually creates for each pod. This annotation is used in this case because the specific workload relies on Open MPI, NCCL, and Horovod to execute the workload in a synchronous distributed manner. Therefore, it requires access to the host networking stack. A discussion about Open MPI, NCCL, and Horovod is outside the scope of this document. Whether or not this `hostNetwork: true` annotation is necessary depends on the requirements of the specific workload that you are executing. For more information about the `hostNetwork` field, see the [official Kubernetes documentation](#).

```
$ cat << EOF > ./netapp-tensorflow-multi-imagenet-worker.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: netapp-tensorflow-multi-imagenet-worker
spec:
  replicas: 1
  selector:
    matchLabels:
      app: netapp-tensorflow-multi-imagenet-worker
  template:
    metadata:
      labels:
        app: netapp-tensorflow-multi-imagenet-worker
    spec:
      hostNetwork: true
      volumes:
        - name: dshm
          emptyDir:
            medium: Memory
        - name: testdata-iface1
          persistentVolumeClaim:
            claimName: pb-fg-all-iface1
        - name: testdata-iface2
          persistentVolumeClaim:
            claimName: pb-fg-all-iface2
EOF
```

```

- name: results
  persistentVolumeClaim:
    claimName: tensorflow-results
  containers:
- name: netapp-tensorflow-py2
  image: netapp/tensorflow-py2:19.03.0
  command: ["bash", "/netapp/scripts/start-slave-multi.sh",
"22122"]
  resources:
    limits:
      nvidia.com/gpu: 8
  volumeMounts:
- mountPath: /dev/shm
  name: dshm
- mountPath: /mnt/mount_0
  name: testdata-iface1
- mountPath: /mnt/mount_1
  name: testdata-iface2
- mountPath: /tmp
  name: results
  securityContext:
    privileged: true
EOF
$ kubectl create -f ./netapp-tensorflow-multi-imagenet-worker.yaml
deployment.apps/netapp-tensorflow-multi-imagenet-worker created
$ kubectl get deployments
NAME                                DESIRED   CURRENT   UP-TO-DATE
AVAILABLE   AGE
netapp-tensorflow-multi-imagenet-worker   1         1         1
1           4s

```

2. Confirm that the worker deployment that you created in step 1 launched successfully. The following example commands confirm that a single worker pod was created for the deployment, as indicated in the deployment definition, and that this pod is currently running on one of the GPU worker nodes.

```

$ kubectl get pods -o wide
NAME                                READY
STATUS     RESTARTS   AGE
IP          NODE          NOMINATED NODE
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running     0          60s   10.61.218.154   10.61.218.154   <none>
$ kubectl logs netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725
22122

```

3. Create a Kubernetes job for a master that kicks off, participates in, and tracks the execution of the

synchronous multinode job. The following example commands create one master that kicks off, participates in, and tracks the synchronous distributed execution of the same TensorFlow benchmark job that was executed on a single node in the example in the section [Execute a Single-Node AI Workload](#).

This example master job requests eight GPUs and thus can run on a single GPU worker node that features eight or more GPUs. If your GPU worker nodes feature more than eight GPUs, to maximize performance, you might want to increase this number to be equal to the number of GPUs that your worker nodes feature.

The master pod that is specified in this example job definition is given a `hostNetwork` value of `true`, just as the worker pod was given a `hostNetwork` value of `true` in step 1. See step 1 for details about why this value is necessary.

```
$ cat << EOF > ./netapp-tensorflow-multi-imagenet-master.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-multi-imagenet-master
spec:
  backoffLimit: 5
  template:
    spec:
      hostNetwork: true
      volumes:
        - name: dshm
          emptyDir:
            medium: Memory
        - name: testdata-iface1
          persistentVolumeClaim:
            claimName: pb-fg-all-iface1
        - name: testdata-iface2
          persistentVolumeClaim:
            claimName: pb-fg-all-iface2
        - name: results
          persistentVolumeClaim:
            claimName: tensorflow-results
      containers:
        - name: netapp-tensorflow-py2
          image: netapp/tensorflow-py2:19.03.0
          command: ["python", "/netapp/scripts/run.py", "--dataset_dir=/mnt/mount_0/dataset/imagenet", "--port=22122", "--num_devices=16", "--dgx_version=dgx1", "--nodes=10.61.218.152,10.61.218.154"]
          resources:
            limits:
              nvidia.com/gpu: 8
      volumeMounts:
        - mountPath: /dev/shm
```

```

        name: dshm
      - mountPath: /mnt/mount_0
        name: testdata-iface1
      - mountPath: /mnt/mount_1
        name: testdata-iface2
      - mountPath: /tmp
        name: results
    securityContext:
      privileged: true
  restartPolicy: Never
EOF
$ kubectl create -f ./netapp-tensorflow-multi-imagenet-master.yaml
job.batch/netapp-tensorflow-multi-imagenet-master created
$ kubectl get jobs
NAME                               COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master   0/1          25s        25s

```

4. Confirm that the master job that you created in step 3 is running correctly. The following example command confirms that a single master pod was created for the job, as indicated in the job definition, and that this pod is currently running on one of the GPU worker nodes. You should also see that the worker pod that you originally saw in step 1 is still running and that the master and worker pods are running on different nodes.

```

$ kubectl get pods -o wide
NAME                                     READY
STATUS      RESTARTS     AGE
IP           NODE         NOMINATED NODE
netapp-tensorflow-multi-imagenet-master-ppwwj   1/1
Running      0            45s   10.61.218.152  10.61.218.152  <none>
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running      0            26m   10.61.218.154  10.61.218.154  <none>

```

5. Confirm that the master job that you created in step 3 completes successfully. The following example commands confirm that the job completed successfully.

```

$ kubectl get jobs
NAME                               COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master   1/1          5m50s     9m18s
$ kubectl get pods
NAME                                     READY
STATUS      RESTARTS     AGE
netapp-tensorflow-multi-imagenet-master-ppwwj   0/1
Completed    0            9m38s
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running      0            35m
$ kubectl logs netapp-tensorflow-multi-imagenet-master-ppwwj

```

```

[10.61.218.152:00008] WARNING: local probe returned unhandled
shell:unknown assuming bash
rm: cannot remove '/lib': Is a directory
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 702
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 711
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 702
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 711
Total images/sec = 12881.33875
===== Clean Cache !!! =====
mpirun -allow-run-as-root -np 2 -H 10.61.218.152:1,10.61.218.154:1 -mca
pml ob1 -mca btl ^openib -mca btl_tcp_if_include enp1s0f0 -mca
plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" bash -c 'sync; echo 1 >
/proc/sys/vm/drop_caches'
=====
mpirun -allow-run-as-root -np 16 -H 10.61.218.152:8,10.61.218.154:8
-bind-to none -map-by slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH
-mca pml ob1 -mca btl ^openib -mca btl_tcp_if_include enp1s0f0 -x
NCCL_IB_HCA=mlx5 -x NCCL_NET_GDR_READ=1 -x NCCL_IB_SL=3 -x
NCCL_IB_GID_INDEX=3 -x
NCCL_SOCKET_IFNAME=enp5s0.3091,enp12s0.3092,enp132s0.3093,enp139s0.3094
-x NCCL_IB_CUDA_SUPPORT=1 -mca orte_base_help_aggregate 0 -mca
plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" python
/netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_be
nchmarks.py --model=resnet50 --batch_size=256 --device=gpu
--force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48
--variable_update=horovod --batch_group_size=20 --num_batches=500
--nodistortions --num_gpus=1 --data_format=NCHW --use_fp16=True
--use_tf_layers=False --data_name=imagenet --use_datasets=True
--data_dir=/mnt/mount_0/dataset/imagenet
--datasets_parallel_interleave_cycle_length=10
--datasets_sloppy_parallel_interleave=False --num_mounts=2
--mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000 --
datasets_use_prefetch=True --datasets_num_private_threads=4
--horovod_device=gpu >
/tmp/20190814_161609_tensorflow_horovod_rdma_resnet50_gpu_16_256_b500_im
agenet_nodistort_fp16_r10_m2_nockpt.txt 2>&1

```

6. Delete the worker deployment when you no longer need it. The following example commands show the deletion of the worker deployment object that was created in step 1.

When you delete the worker deployment object, Kubernetes automatically deletes any associated worker pods.

```

$ kubectl get deployments
NAME                                DESIRED   CURRENT   UP-TO-DATE
AVAILABLE   AGE
netapp-tensorflow-multi-imagenet-worker   1         1         1
1           43m

$ kubectl get pods
NAME                                         READY
STATUS      RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj   0/1
Completed   0          17m
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running     0          43m

$ kubectl delete deployment netapp-tensorflow-multi-imagenet-worker
deployment.extensions "netapp-tensorflow-multi-imagenet-worker" deleted
$ kubectl get deployments
No resources found.

$ kubectl get pods
NAME                                READY   STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj   0/1     Completed   0
18m

```

7. **Optional:** Clean up the master job artifacts. The following example commands show the deletion of the master job object that was created in step 3.

When you delete the master job object, Kubernetes automatically deletes any associated master pods.

```

$ kubectl get jobs
NAME                                COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master   1/1          5m50s    19m
$ kubectl get pods
NAME                                READY   STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj   0/1     Completed   0
19m

$ kubectl delete job netapp-tensorflow-multi-imagenet-master
job.batch "netapp-tensorflow-multi-imagenet-master" deleted
$ kubectl get jobs
No resources found.

$ kubectl get pods
No resources found.

```

[Next: Performance Testing](#)

## Performance Testing

We performed a simple performance comparison as part of the creation of this solution. We executed several standard NetApp AI benchmarking jobs by using Kubernetes, and we compared the benchmark results with executions that were performed by using a simple Docker run command. We did not see any noticeable differences in performance. Therefore, we concluded that the use of Kubernetes to orchestrate containerized AI training jobs does not adversely affect performance. See the following table for the results of our performance comparison.

Benchmark	Dataset	Docker Run (images/sec)	Kubernetes (images/sec)
Single-node TensorFlow	Synthetic data	6,667.2475	6,661.93125
Single-node TensorFlow	ImageNet	6,570.2025	6,530.59125
Synchronous distributed two-node TensorFlow	Synthetic data	13,213.70625	13,218.288125
Synchronous distributed two-node TensorFlow	ImageNet	12,941.69125	12,881.33875

[Next: Conclusion](#)

## Conclusion

Companies and organizations of all sizes and across all industries are turning to artificial intelligence (AI), machine learning (ML), and deep learning (DL) to solve real-world problems, deliver innovative products and services, and to get an edge in an increasingly competitive marketplace. As organizations increase their use of AI, ML, and DL, they face many challenges, including workload scalability and data availability. These challenges can be addressed through the use of the NetApp AI Control Plane solution.

This solution enables you to rapidly clone a data namespace. Additionally, it allows you to define and implement AI, ML, and DL training workflows that incorporate the near-instant creation of data and model baselines for traceability and versioning. With this solution, you can trace every single model training run back to the exact dataset(s) that the model was trained and/or validated with. Lastly, this solution enables you to swiftly provision Jupyter Notebook workspaces with access to massive datasets.

Because this solution is targeted towards data scientists and data engineers, minimal NetApp or NetApp ONTAP expertise is required. With this solution, data management functions can be executed using simple and familiar tools and interfaces. Furthermore, this solution utilizes fully open-source and free components. Therefore, if you already have NetApp storage in your environment, you can implement this solution today. If you want to test drive this solution but you do not have already have NetApp storage, visit [cloud.netapp.com](http://cloud.netapp.com), and you can be up and running with a cloud-based NetApp storage solution in no time.

## MLRun Pipeline

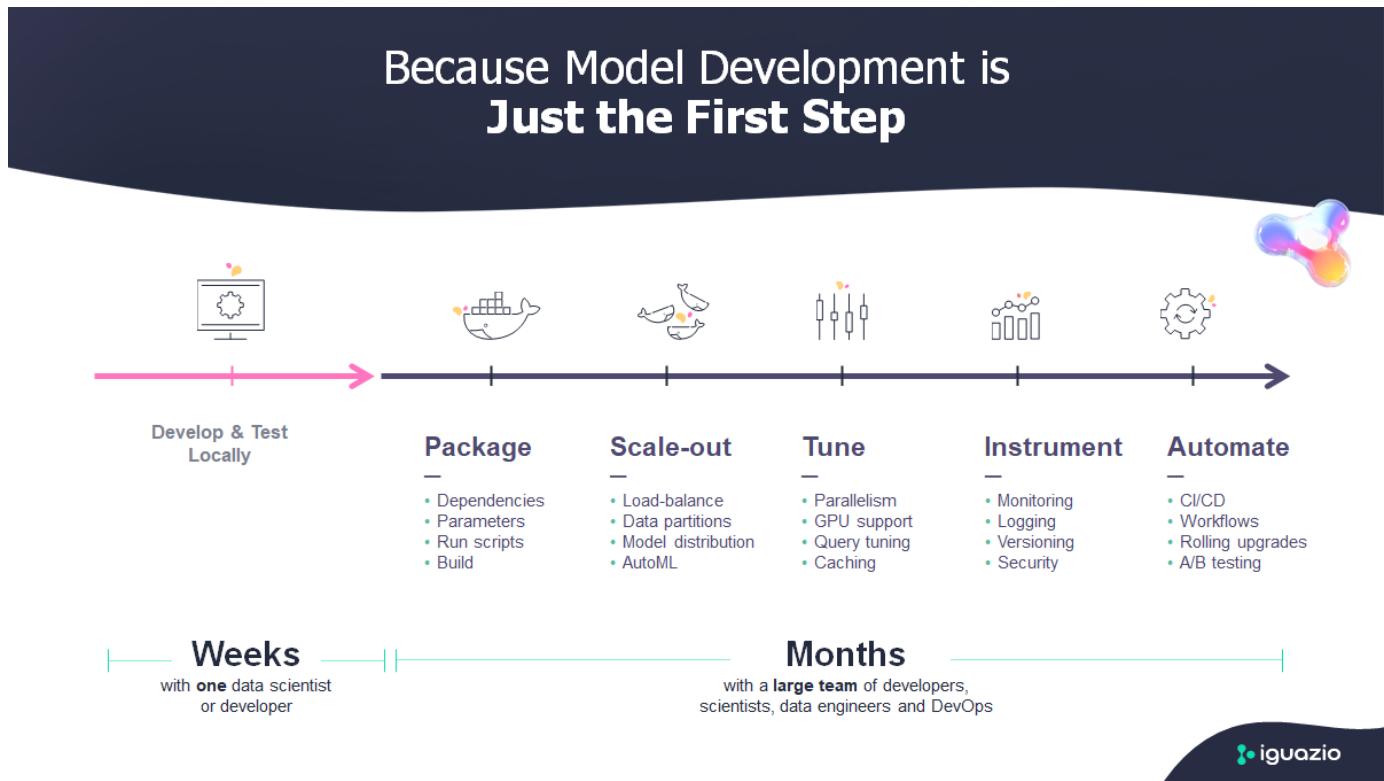
### TR-4834: NetApp and Iguazio for MLRun Pipeline

Rick Huang, David Arnette, NetApp  
Marcelo Litovsky, Iguazio

This document covers the details of the MLRun pipeline using NetApp ONTAP AI, NetApp AI Control Plane,

NetApp Cloud Volumes software, and the Iguazio Data Science Platform. We used Nuclio serverless function, Kubernetes Persistent Volumes, NetApp Cloud Volumes, NetApp Snapshot copies, Grafana dashboard, and other services on the Iguazio platform to build an end-to-end data pipeline for the simulation of network failure detection. We integrated Iguazio and NetApp technologies to enable fast model deployment, data replication, and production monitoring capabilities on premises as well as in the cloud.

The work of a data scientist should be focused on the training and tuning of machine learning (ML) and artificial intelligence (AI) models. However, according to research by Google, data scientists spend ~80% of their time figuring out how to make their models work with enterprise applications and run at scale, as shown in the following image depicting model development in the AI/ML workflow.



To manage end-to-end AI/ML projects, a wider understanding of enterprise components is needed. Although DevOps have taken over the definition, integration, and deployment these types of components, machine learning operations target a similar flow that includes AI/ML projects. To get an idea of what an end-to-end AI/ML pipeline touches in the enterprise, see the following list of required components:

- Storage
- Networking
- Databases
- File systems
- Containers
- Continuous integration and continuous deployment (CI/CD) pipeline
- Development integrated development environment (IDE)
- Security
- Data access policies
- Hardware

- Cloud
- Virtualization
- Data science toolsets and libraries

In this paper, we demonstrate how the partnership between NetApp and Iguazio drastically simplifies the development of an end-to-end AI/ML pipeline. This simplification accelerates the time to market for all of your AI/ML applications.

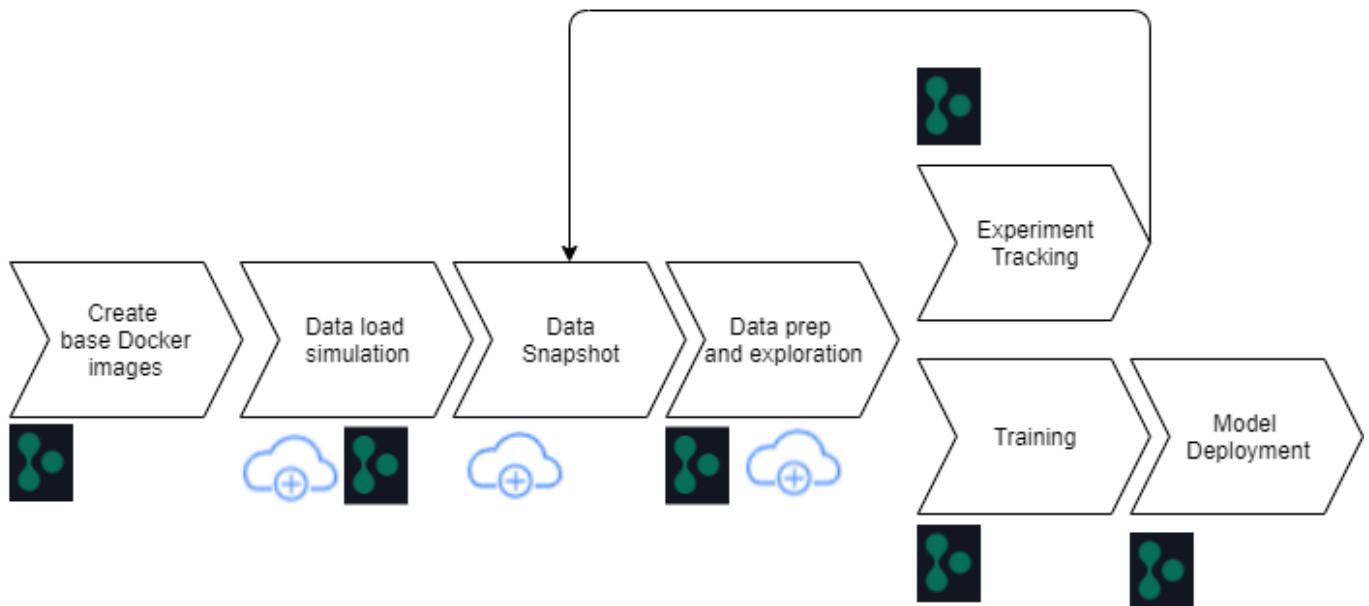
### Target Audience

The world of data science touches multiple disciplines in information technology and business.

- The data scientist needs the flexibility to use their tools and libraries of choice.
- The data engineer needs to know how the data flows and where it resides.
- A DevOps engineer needs the tools to integrate new AI/ML applications into their CI/CD pipelines.
- Business users want to have access to AI/ML applications. We describe how NetApp and Iguazio help each of these roles bring value to business with our platforms.

### Solution Overview

This solution follows the lifecycle of an AI/ML application. We start with the work of data scientists to define the different steps needed to prep data and train and deploy models. We follow with the work needed to create a full pipeline with the ability to track artifacts, experiment with execution, and deploy to Kubeflow. To complete the full cycle, we integrate the pipeline with NetApp Cloud Volumes to enable data versioning, as seen in the following image.



[Next: Technology Overview](#)

## Technology Overview

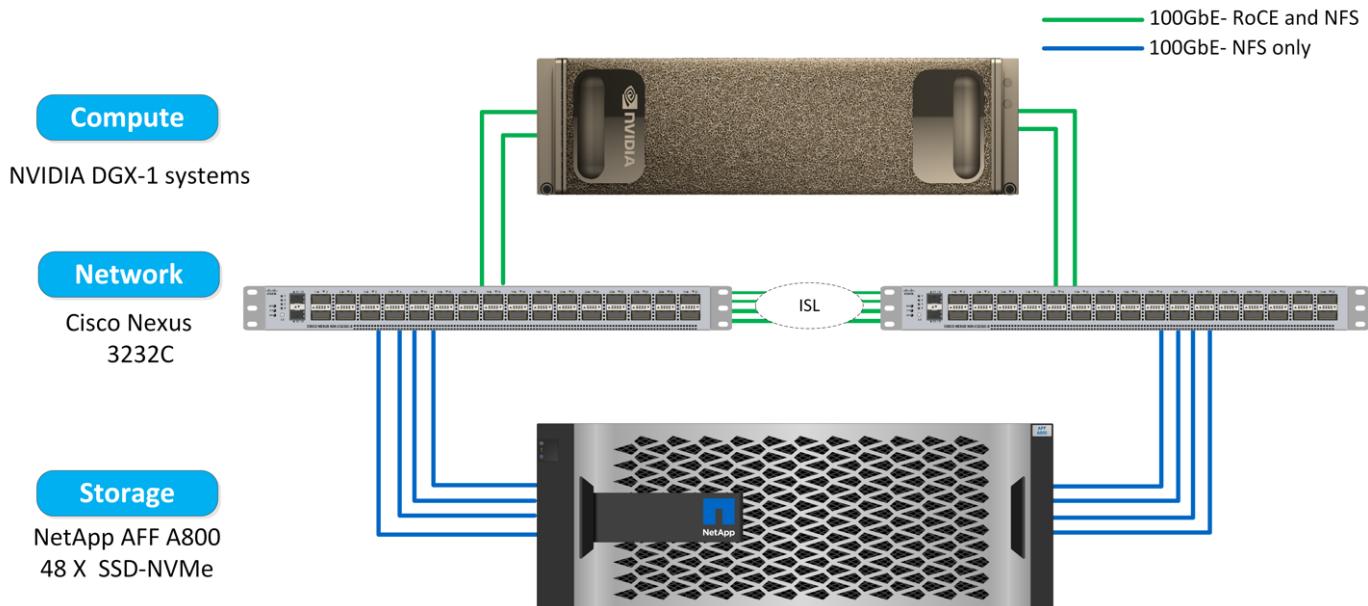
### NetApp Overview

NetApp is the data authority for the hybrid cloud. NetApp provides a full range of hybrid cloud data services that simplify management of applications and data across cloud and on-premises environments to accelerate digital transformation. Together with our partners, NetApp empowers global organizations to unleash the full potential of their data to expand customer touch points, foster greater innovation, and optimize their operations.

### NetApp ONTAP AI

NetApp ONTAP AI, powered by NVIDIA DGX systems and NetApp cloud-connected all-flash storage, streamlines the flow of data reliably and speeds up analytics, training, and inference with your data fabric that spans from edge to core to cloud. It gives IT organizations an architecture that provides the following benefits:

- Eliminates design complexities
  - Allows independent scaling of compute and storage
  - Enables customers to start small and scale seamlessly
  - Offers a range of storage options for various performance and cost points
- NetApp ONTAP AI offers converged infrastructure stacks incorporating NVIDIA DGX-1, a petaflop-scale AI system, and NVIDIA Mellanox high-performance Ethernet switches to unify AI workloads, simplify deployment, and accelerate ROI. We leveraged ONTAP AI with one DGX-1 and NetApp AFF A800 storage system for this technical report. The following image shows the topology of ONTAP AI with the DGX-1 system used in this validation.



### NetApp AI Control Plane

The NetApp AI Control Plane enables you to unleash AI and ML with a solution that offers extreme scalability, streamlined deployment, and nonstop data availability. The AI Control Plane solution integrates Kubernetes and Kubeflow with a data fabric enabled by NetApp. Kubernetes, the industry-standard container orchestration platform for cloud-native deployments, enables workload scalability and portability. Kubeflow is an open-source machine-learning platform that simplifies management and deployment, enabling developers to do more data science in less time. A data fabric enabled by NetApp offers uncompromising data availability and portability to make sure that your data is accessible across the pipeline, from edge to core to cloud. This technical report uses the NetApp AI Control Plane in an MLRun pipeline. The following image shows Kubernetes cluster

management page where you can have different endpoints for each cluster. We connected NFS Persistent Volumes to the Kubernetes cluster, and the following images show an Persistent Volume connected to the cluster, where [NetApp Trident](#) offers persistent storage support and data management capabilities.

The screenshot shows two entries for 'kubernetes' in the 'Kubernetes Clusters' section. Each entry includes a cluster icon, the name 'kubernetes', a 'Cluster Endpoint' (with links to https://3.20.111.39:6443 and https://172.31.14.31:6443), a 'Cluster Version' (v1.15.5), a 'Trident Version' (19.07.1), and a 'Working Environments' count (0 and 1 respectively).

## Persistent Volumes for Kubernetes

### Connected with Kubernetes Cluster

Cloud Volumes ONTAP is connected to 1 Kubernetes cluster. [View Cluster](#) i

You can connect another Kubernetes cluster to this Cloud Volumes ONTAP system. If the Kubernetes cluster is in a different network than Cloud Volumes ONTAP, specify a custom export policy to provide access to clients.

#### Kubernetes Cluster

##### Select Kubernetes Cluster

kubernetes

#### Custom Export Policy (Optional)

##### Custom Export Policy

172.31.0.0/16

Set as default storage class

NFS  iSCSI

[Connect](#)

[Cancel](#)



demo

Volumes

Instances

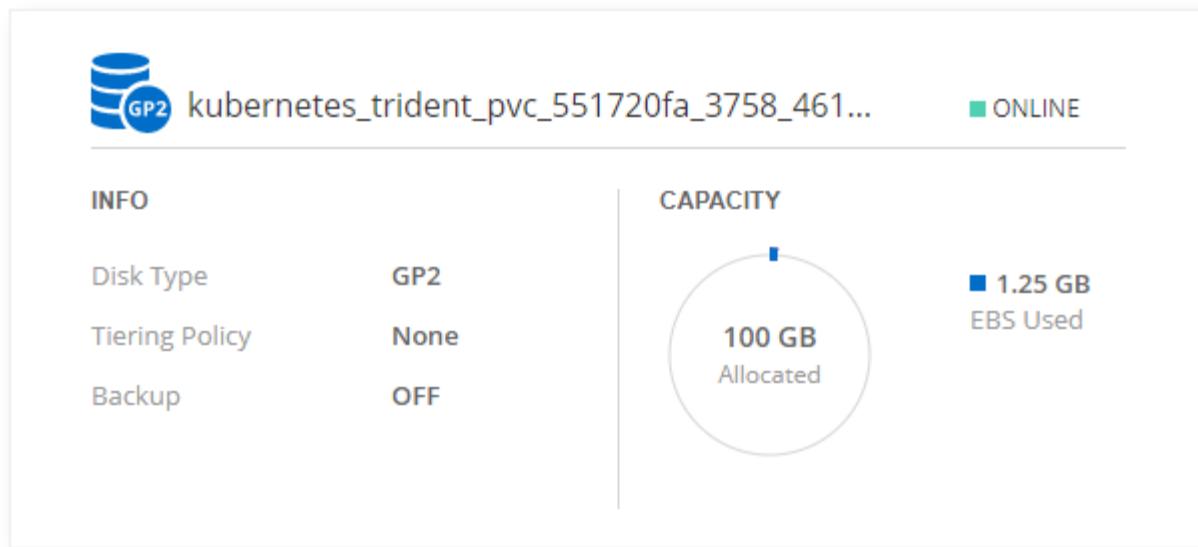
Cost

Replications

Sync to S3

## Volumes

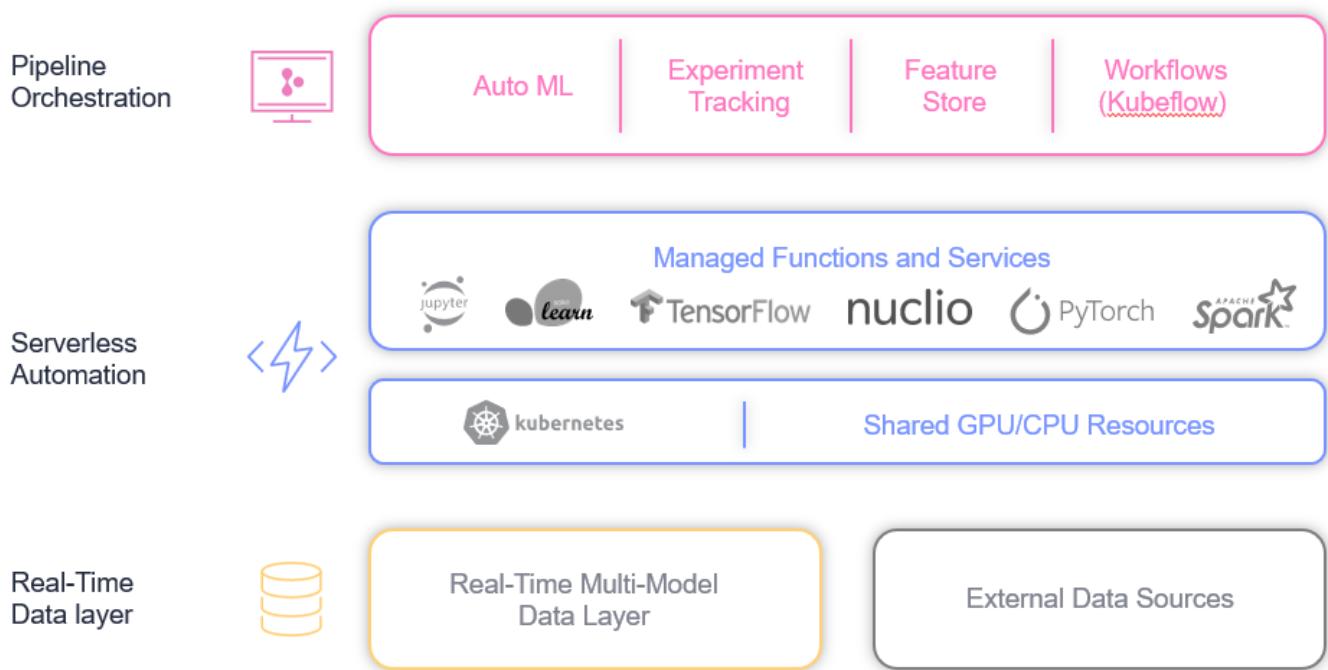
4 Volumes | 300 GB Allocated | 1.43 GB Total Used



## Iguazio Overview

The Iguazio Data Science Platform is a fully integrated and secure data-science platform as a service (PaaS) that simplifies development, accelerates performance, facilitates collaboration, and addresses operational challenges. This platform incorporates the following components, and the Iguazio Data Science Platform is presented in the following image:

- A data-science workbench that includes Jupyter Notebooks, integrated analytics engines, and Python packages
- Model management with experiments tracking and automated pipeline capabilities
- Managed data and ML services over a scalable Kubernetes cluster
- Nuclio, a real-time serverless functions framework
- An extremely fast and secure data layer that supports SQL, NoSQL, time-series databases, files (simple objects), and streaming
- Integration with third-party data sources such as NetApp, Amazon S3, HDFS, SQL databases, and streaming or messaging protocols
- Real-time dashboards based on Grafana



[Next: Software and Hardware Requirements](#)

## Software and Hardware Requirements

### Network Configuration

The following is the network configuration requirement for setting up in the cloud:

- The Iguazio cluster and NetApp Cloud Volumes must be in the same virtual private cloud.
- The cloud manager must have access to port 6443 on the Iguazio app nodes.
- We used Amazon Web Services in this technical report. However, users have the option of deploying the solution in any Cloud provider. For on-premises testing in ONTAP AI with NVIDIA DGX-1, we used the Iguazio hosted DNS service for convenience.

Clients must be able to access dynamically created DNS domains. Customers can use their own DNS if desired.

### Hardware Requirements

You can install Iguazio on-premises in your own cluster. We have verified the solution in NetApp ONTAP AI with an NVIDIA DGX-1 system. The following table lists the hardware used to test this solution.

Hardware	Quantity
DGX-1 systems	1
NetApp AFF A800 system	1 high-availability (HA) pair, includes 2 controllers and 48 NVMe SSDs (3.8TB or above)
Cisco Nexus 3232C network switches	2

The following table lists the software components required for on-premise testing:

Software	Version or Other Information
NetApp ONTAP data management software	9.7
Cisco NX-OS switch firmware	7.0(3)I6(1)
NVIDIA DGX OS	4.4 - Ubuntu 18.04 LTS
Docker container platform	19.03.5
Container version	20.01-tf1-py2
Machine learning framework	TensorFlow 1.15.0
Iguazio	Version 2.8+
ESX Server	6.5

This solution was fully tested with Iguazio version 2.5 and NetApp Cloud Volumes ONTAP for AWS. The Iguazio cluster and NetApp software are both running on AWS.

Software	Version or Type
Iguazio	Version 2.8+
App node	M5.4xlarge
Data node	I3.4xlarge

[Next: Network Device Failure Prediction Use Case Summary](#)

## Network Device Failure Prediction Use Case Summary

This use case is based on an Iguazio customer in the telecommunications space in Asia. With 100K enterprise customers and 125k network outage events per year, there was a critical need to predict and take proactive action to prevent network failures from affecting customers. This solution provided them with the following benefits:

- Predictive analytics for network failures
- Integration with a ticketing system
- Taking proactive action to prevent network failuresAs a result of this implementation of Iguazio, 60% of failures were proactively prevented.

[Next: Setup Overview](#)

## Setup Overview

### Iguazio Installation

Iguazio can be installed on-premises or on a cloud provider. Provisioning can be done as a service and managed by Iguazio or by the customer. In both cases, Iguazio provides a deployment application (Provazio) to deploy and manage clusters.

For on-premises installation, please refer to [NVA-1121](#) for compute, network, and storage setup. On-premises deployment of Iguazio is provided by Iguazio without additional cost to the customer. See [this page](#) for DNS and SMTP server configurations. The Provazio installation page is shown as follows.

Installation Scenario General Clusters Cloud

- Bare metal / virtual machines  
Installs the system on bare-metal or virtual-machine instances, pre-provisioned with prerequ...

- AWS  
Creates applicable compute/networking resources in AWS and installs the system on the in...

- Azure  
Creates applicable compute/networking resources in Azure and installs the system on the i...

- AWS (pre-provisioned)  
Installs the system on Amazon Web Services instances, manually provisioned beforehand

- Azure (pre-provisioned)  
Installs the system on Microsoft Azure instances, manually provisioned beforehand

- Advanced  
Show advanced options in the next steps

BACK

NEXT

## Next: Configuring Kubernetes Cluster

### Configuring Kubernetes Cluster

This section is divided into two parts for cloud and on-premises deployment respectively.

#### Cloud Deployment Kubernetes Configuration

Through NetApp Cloud Manager, you can define the connection to the Iguazio Kubernetes cluster. Trident requires access to multiple resources in the cluster to make the volume available.

1. To enable access, obtain the Kubernetes config file from one the Iguazio nodes. The file is located under `/home/Iguazio/.kube/config`. Download this file to your desktop.
2. Go to Discover Cluster to configure.

## 4 Kubernetes Clusters

Cluster Overview			
 kubernetes			
 https://3.20.111.39:6443	 v1.15.5	 19.07.1	 0
Cluster Endpoint	Cluster Version	Trident Version	Working Environments
 kubernetes			
 https://172.31.14.31:6443	 v1.15.5	 19.07.1	 1
Cluster Endpoint	Cluster Version	Trident Version	Working Environments

3. Upload the Kubernetes config file. See the following image.

## Upload Kubernetes Configuration File

Upload the Kubernetes configuration file (kubeconfig) so Cloud Manager can install Trident on the Kubernetes cluster.

Connecting Cloud Volumes ONTAP with a Kubernetes cluster enables users to request and manage persistent volumes using native Kubernetes interfaces and constructs. Users can take advantage of ONTAP's advanced data management features without having to know anything about it. Storage provisioning is enabled by using NetApp Trident.

Learn more about [Trident for Kubernetes](#).

[Upload File](#)

4. Deploy Trident and associate a volume with the cluster. See the following image on defining and assigning a Persistent Volume to the Iguazio cluster. This process creates a Persistent Volume (PV) in Iguazio's Kubernetes cluster. Before you can use it, you must define a Persistent Volume Claim (PVC).

## Persistent Volumes for Kubernetes

### Connected with Kubernetes Cluster

Cloud Volumes ONTAP is connected to 1 Kubernetes cluster. [View Cluster](#) ⓘ

You can connect another Kubernetes cluster to this Cloud Volumes ONTAP system. If the Kubernetes cluster is in a different network than Cloud Volumes ONTAP, specify a custom export policy to provide access to clients.

Kubernetes Cluster	Custom Export Policy (Optional) ⓘ
Select Kubernetes Cluster	Custom Export Policy
<input type="text" value="kubernetes"/>	<input type="text" value="172.31.0.0/16"/>
<input checked="" type="checkbox"/> Set as default storage class <input checked="" type="radio"/> NFS <input type="radio"/> iSCSI	
<div style="text-align: right;"><a href="#">Connect</a> <a href="#">Cancel</a></div>	

### On-Premises Deployment Kubernetes Configuration

For on-premises installation of NetApp Trident, see [TR-4798](#) for details. After configuring your Kubernetes cluster and installing NetApp Trident, you can connect Trident to the Iguazio cluster to enable NetApp data management capabilities, such as taking Snapshot copies of your data and model.

[Next: Define Persistent Volume Claim](#)

#### Define Persistent Volume Claim

1. Save the following YAML to a file to create a PVC of type Basic.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: netapp-file
```

## 2. Apply the YAML file to your Iguazio Kubernetes cluster.

```
Kubectl -n default-tenant apply -f <your yaml file>
```

### Attach NetApp Volume to the Jupyter Notebook

Iguazio offers several managed services to provide data scientists with a full end-to-end stack for development and deployment of AI/ML applications. You can read more about these components at the [Iguazio Overview of Application Services and Tools](#).

One of the managed services is Jupyter Notebook. Each developer gets its own deployment of a notebook container with the resources they need for development. To give them access to the NetApp Cloud Volume, you can assign the volume to their container and resource allocation, running user, and environment variable settings for Persistent Volume Claims is presented in the following image.

For an on-premises configuration, you can refer to [TR-4798](#) on the Trident setup to enable NetApp ONTAP data management capabilities, such as taking Snapshot copies of your data or model for versioning control. Add the following line in your Trident back- end config file to make Snapshot directories visible:

```
{  
  ...  
  "defaults": {  
    "snapshotDir": "true"  
  }  
}
```

You must create a Trident back- end config file in JSON format, and then run the following [Trident command](#) to reference it:

```
tridentctl create backend -f <backend-file>
```

The screenshot shows the Iguazio UI configuration interface for a Jupyter Notebook pod. It includes the following sections:

- Inactivity window:** A slider set to 10m.
- Resources:** Memory Request: 5GB, Limit: 10GB; CPU Request: 1500 milliCPU, Limit: 1500 milliCPU.
- Running User:** Set to admin.

The screenshot shows the Iguazio UI configuration interface for Persistent Volume Claims (PVCs). It includes the following sections:

- Environment Variables:** A button to "Create a new environment variable".
- Persistent Volume Claims (PVCs):** A table with one entry: Name: basic, Mount Path: /netapp. A "Add PVC" button is also present.

[Next: Deploying the Application](#)

## Deploying the Application

The following sections describe how to install and deploy the application.

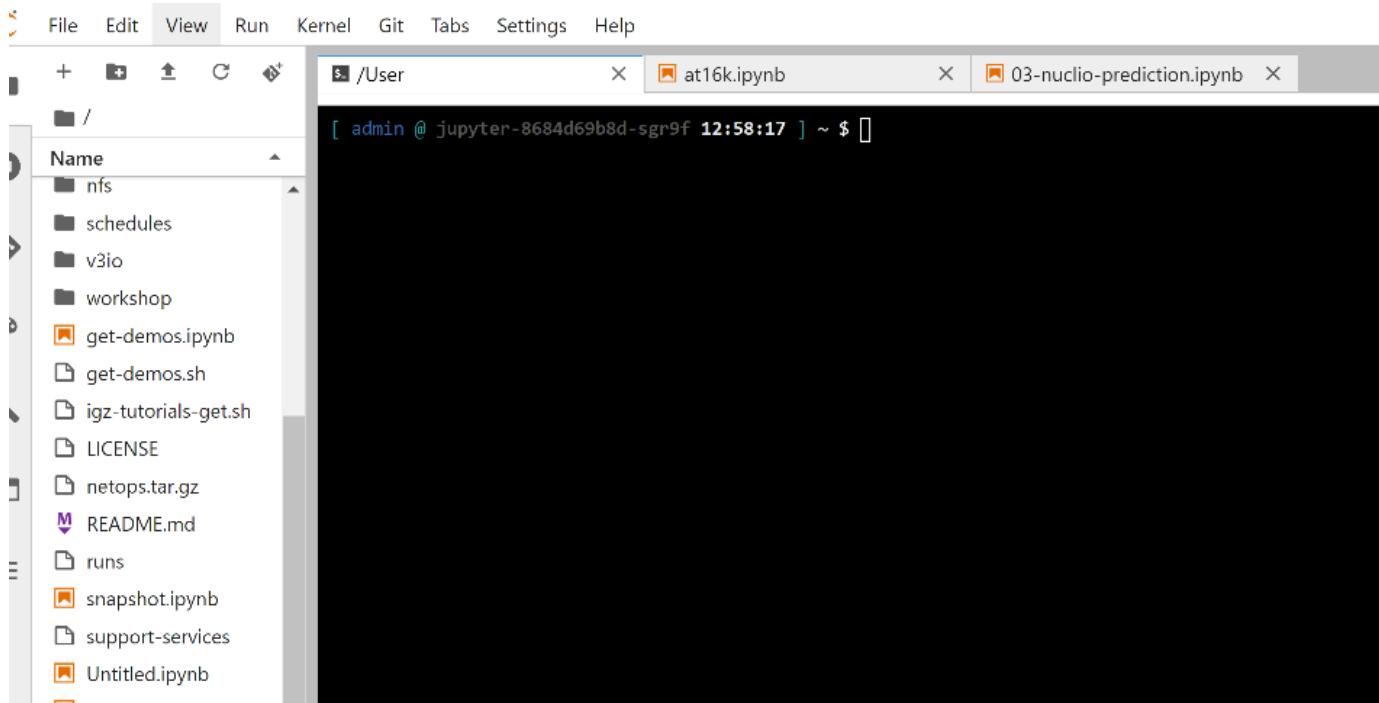
Next: [Get Code from GitHub](#).

### Get Code from GitHub

Now that the NetApp Cloud Volume or NetApp Trident volume is available to the Iguazio cluster and the developer environment, you can start reviewing the application.

Users have their own workspace (directory). On every notebook, the path to the user directory is `/User`. The Iguazio platform manages the directory. If you follow the instructions above, the NetApp Cloud volume is available in the `/netapp` directory.

Get the code from GitHub using a Jupyter terminal.



At the Jupyter terminal prompt, clone the project.

```
cd /User  
git clone .
```

You should now see the `netops-` `netapp` folder on the file tree in Jupyter workspace.

Next: [Configure Working Environment](#)

### Configure Working Environment

Copy the `Notebook set_env-Example.ipynb` as `set_env.ipynb`. Open and edit `set_env.ipynb`. This notebook sets variables for credentials, file locations, and

execution drivers.

If you follow the instructions above, the following steps are the only changes to make:

1. Obtain this value from the Iguazio services dashboard: `docker_registry`

Example: `docker-registry.default-tenant.app.clusterq.iguaziodev.com:80`

2. Change `admin` to your Iguazio username:

```
IGZ_CONTAINER_PATH = '/users/admin'
```

The following are the ONTAP system connection details. Include the volume name that was generated when Trident was installed. The following setting is for an on-premises ONTAP cluster:

```
ontapClusterMgmtHostname = '0.0.0.0'  
ontapClusterAdminUsername = 'USER'  
ontapClusterAdminPassword = 'PASSWORD'  
sourceVolumeName = 'SOURCE VOLUME'
```

The following setting is for Cloud Volumes ONTAP:

```
MANAGER=ontapClusterMgmtHostname  
svm='svm'  
email='email'  
password=ontapClusterAdminPassword  
weid="weid"  
volume=sourceVolumeName
```

## Create Base Docker Images

Everything you need to build an ML pipeline is included in the Iguazio platform. The developer can define the specifications of the Docker images required to run the pipeline and execute the image creation from Jupyter Notebook. Open the notebook `create- images.ipynb` and Run All Cells.

This notebook creates two images that we use in the pipeline.

- `iguazio/netapp`. Used to handle ML tasks.

## Create image for training pipeline

```
[4]: fn.build_config(image= docker_registry + '/iguazio/netapp', commands=['pip install \  
v3io_frames fsspec==0.3.3 PyYAML==5.1.2 pyarrow==0.15.1 pandas==0.25.3 matplotlib seaborn yellowb  
fn.deploy()
```

- `netapp/pipeline`. Contains utilities to handle NetApp Snapshot copies.

## Create image for Ontap utilities

```
[9]: fn.build_config(image= docker_registry + '/netapp/pipeline:latest', commands=['apt -y update','pip install vio_frames netapp_ontap']
fn.deploy()
```

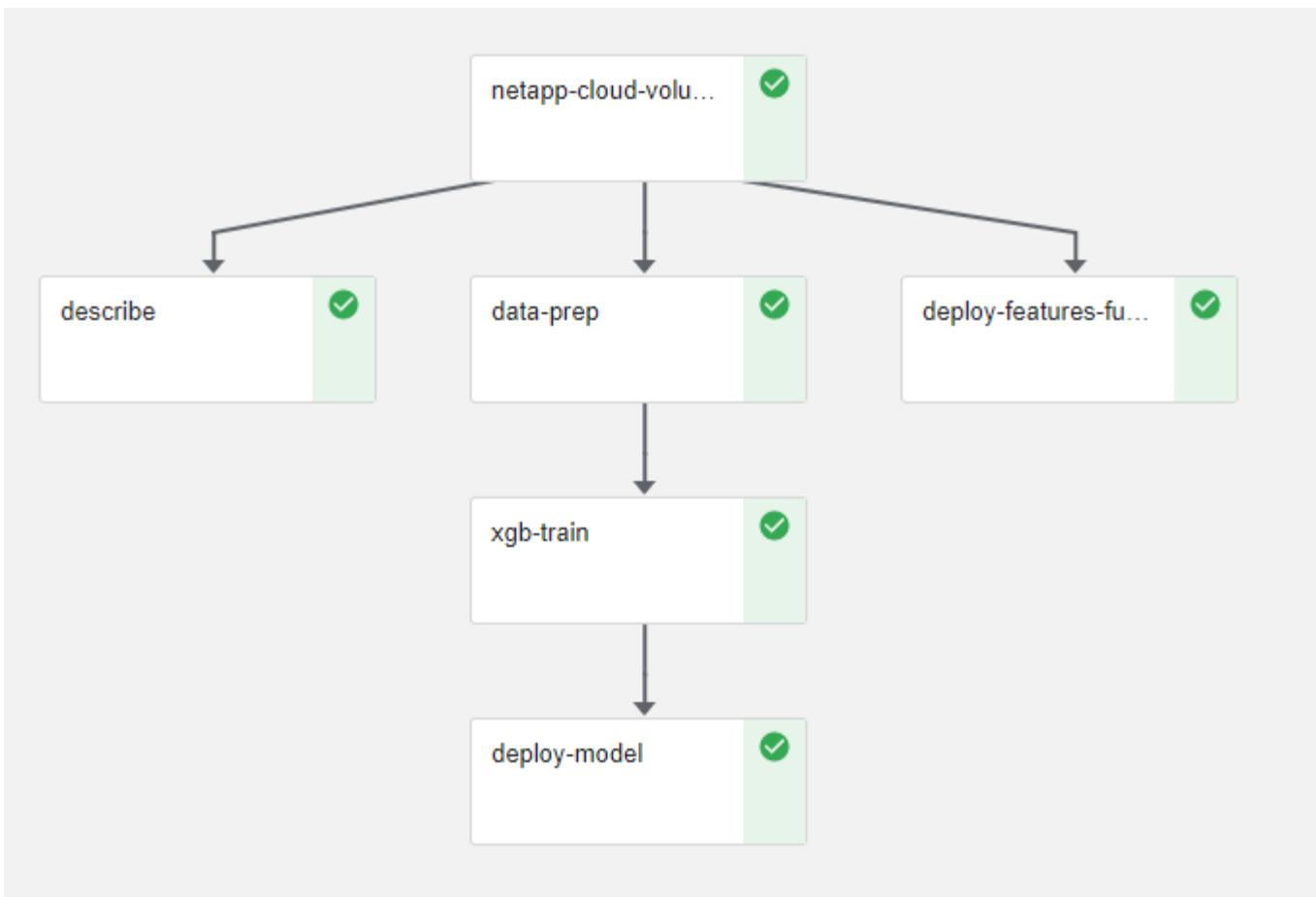
## Review Individual Jupyter Notebooks

The following table lists the libraries and frameworks we used to build this task. All these components have been fully integrated with Iguazio's role-based access and security controls.

Libraries/Framework	Description
MLRun	An managed by Iguazio to enable the assembly, execution, and monitoring of an ML/AI pipeline.
Nuclio	A serverless functions framework integrated with Iguazio. Also available as an open-source project managed by Iguazio.
Kubeflow	A Kubernetes-based framework to deploy the pipeline. This is also an open-source project to which Iguazio contributes. It is integrated with Iguazio for added security and integration with the rest of the infrastructure.
Docker	A Docker registry run as a service in the Iguazio platform. You can also change this to connect to your registry.
NetApp Cloud Volumes	Cloud Volumes running on AWS give us access to large amounts of data and the ability to take Snapshot copies to version the datasets used for training.
Trident	Trident is an open-source project managed by NetApp. It facilitates the integration with storage and compute resources in Kubernetes.

We used several notebooks to construct the ML pipeline. Each notebook can be tested individually before being brought together in the pipeline. We cover each notebook individually following the deployment flow of this demonstration application.

The desired result is a pipeline that trains a model based on a Snapshot copy of the data and deploys the model for inference. A block diagram of a completed MLRun pipeline is shown in the following image.



## Deploy Data Generation Function

This section describes how we used Nuclio serverless functions to generate network device data. The use case is adapted from an Iguazio client that deployed the pipeline and used Iguazio services to monitor and predict network device failures.

We simulated data coming from network devices. Executing the Jupyter notebook `data-generator.ipynb` creates a serverless function that runs every 10 minutes and generates a Parquet file with new data. To deploy the function, run all the cells in this notebook. See the [Nuclio website](#) to review any unfamiliar components in this notebook.

A cell with the following comment is ignored when generating the function. Every cell in the notebook is assumed to be part of the function. Import the Nuclio module to enable `%nuclio magic`.

```
# nuclio: ignore
import nuclio
```

In the spec for the function, we defined the environment in which the function executes, how it is triggered, and the resources it consumes.

```
spec = nuclio.ConfigSpec(config={"spec.triggers.inference.kind":"cron",
"spec.triggers.inference.attributes.interval" :"10m",
"spec.readinessTimeoutSeconds" : 60,
"spec.minReplicas" : 1},.....
```

The `init_context` function is invoked by the Nuclio framework upon initialization of the function.

```
def init_context(context):
    ...
```

Any code not in a function is invoked when the function initializes. When you invoke it, a handler function is executed. You can change the name of the handler and specify it in the function spec.

```
def handler(context, event):
    ...
```

You can test the function from the notebook prior to deployment.

```
%time
# nuclio: ignore
init_context(context)
event = nuclio.Event(body='')
output = handler(context, event)
output
```

The function can be deployed from the notebook or it can be deployed from a CI/CD pipeline (adapting this code).

```
addr = nuclio.deploy_file(name='generator', project='netops', spec=spec,
tag='v1.1')
```

## Pipeline Notebooks

These notebooks are not meant to be executed individually for this setup. This is just a review of each notebook. We invoked them as part of the pipeline. To execute them individually, review the MLRun documentation to execute them as Kubernetes jobs.

### **snap\_cv.ipynb**

This notebook handles the Cloud Volume Snapshot copies at the beginning of the pipeline. It passes the name of the volume to the pipeline context. This notebook invokes a shell script to handle the Snapshot copy. While running in the pipeline, the execution context contains variables to help locate all files needed for execution.

While writing this code, the developer does not have to worry about the file location in the container that executes it. As described later, this application is deployed with all its dependencies, and it is the definition of the pipeline parameters that provides the execution context.

```
command = os.path.join(context.get_param('APP_DIR'), "snap_cv.sh")
```

The created Snapshot copy location is placed in the MLRun context to be consumed by steps in the pipeline.

```
context.log_result('snapVolumeDetails', snap_path)
```

The next three notebooks are run in parallel.

### **data-prep.ipynb**

Raw metrics must be turned into features to enable model training. This notebook reads the raw metrics from the Snapshot directory and writes the features for model training to the NetApp volume.

When running in the context of the pipeline, the input `DATA_DIR` contains the Snapshot copy location.

```
metrics_table = os.path.join(str(mlruncontext.get_input('DATA_DIR',
os.getenv('DATA_DIR','/netpp'))),
                           mlruncontext.get_param('metrics_table',
os.getenv('metrics_table','netops_metrics_parquet')))
```

### **describe.ipynb**

To visualize the incoming metrics, we deploy a pipeline step that provides plots and graphs that are available through the Kubeflow and MLRun UIs. Each execution has its own version of this visualization tool.

```
ax.set_title("features correlation")
plt.savefig(os.path.join(base_path, "plots/corr.png"))
context.log_artifact(PlotArtifact("correlation", body=plt.gcf()),
local_path="plots/corr.html")
```

### **deploy-feature-function.ipynb**

We continuously monitor the metrics looking for anomalies. This notebook creates a serverless function that generates the features need to run prediction on incoming metrics. This notebook invokes the creation of the function. The function code is in the notebook `data-prep.ipynb`. Notice that we use the same notebook as a step in the pipeline for this purpose.

### **training.ipynb**

After we create the features, we trigger the model training. The output of this step is the model to be used for inferencing. We also collect statistics to keep track of each execution (experiment).

For example, the following command enters the accuracy score into the context for that experiment. This value is visible in Kubeflow and MLRun.

```
context.log_result('accuracy', score)
```

### deploy-inference-function.ipynb

The last step in the pipeline is to deploy the model as a serverless function for continuous inferencing. This notebook invokes the creation of the serverless function defined in [nuclio-inference-function.ipynb](#).

### Review and Build Pipeline

The combination of running all the notebooks in a pipeline enables the continuous run of experiments to reassess the accuracy of the model against new metrics. First, open the [pipeline.ipynb](#) notebook. We take you through details that show how NetApp and Iguazio simplify the deployment of this ML pipeline.

We use MLRun to provide context and handle resource allocation to each step of the pipeline. The MLRun API service runs in the Iguazio platform and is the point of interaction with Kubernetes resources. Each developer cannot directly request resources; the API handles the requests and enables access controls.

```
# MLRun API connection definition
mlconf.dbpath = 'http://mlrun-api:8080'
```

The pipeline can work with NetApp Cloud Volumes and on-premises volumes. We built this demonstration to use Cloud Volumes, but you can see in the code the option to run on-premises.

```

# Initialize the NetApp snap function once for all functions in a notebook
if [ NETAPP_CLOUD_VOLUME ]:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snap_cv.ipynb").apply(mount_v3io())
    snap_params = {
        "metrics_table" : metrics_table,
        "NETAPP_MOUNT_PATH" : NETAPP_MOUNT_PATH,
        'MANAGER' : MANAGER,
        'svm' : svm,
        'email': email,
        'password': password ,
        'weid': weid,
        'volume': volume,
        "APP_DIR" : APP_DIR
    }
else:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snapshot.ipynb").apply(mount_v3io())
...
snapfn.spec.image = docker_registry + '/netapp/pipeline:latest'
snapfn.spec.volume_mounts =
[snapfn.spec.volume_mounts[0],netapp_volume_mounts]
    snapfn.spec.volumes = [ snapfn.spec.volumes[0],netapp_volumes]

```

The first action needed to turn a Jupyter notebook into a Kubeflow step is to turn the code into a function. A function has all the specifications required to run that notebook. As you scroll down the notebook, you can see that we define a function for every step in the pipeline.

Part of the Notebook	Description
<code_to_function> (part of the MLRun module)	Name of the function: Project name. used to organize all project artifacts. This is visible in the MLRun UI. Kind. In this case, a Kubernetes job. This could be Dask, mpi, sparkk8s, and more. See the MLRun documentation for more details. File. The name of the notebook. This can also be a location in Git (HTTP).
image	The name of the Docker image we are using for this step. We created this earlier with the create-image.ipynb notebook.
volume_mounts & volumes	Details to mount the NetApp Cloud Volume at run time.

We also define parameters for the steps.

```

params={ "FEATURES_TABLE":FEATURES_TABLE,
          "SAVE_TO" : SAVE_TO,
          "metrics_table" : metrics_table,
          'FROM_TSDB': 0,
          'PREDICTIONS_TABLE': PREDICTIONS_TABLE,
          'TRAIN_ON_LAST': '1d',
          'TRAIN_SIZE':0.7,
          'NUMBER_OF_SHARDS' : 4,
          'MODEL_FILENAME' : 'netops.v3.model.pickle',
          'APP_DIR' : APP_DIR,
          'FUNCTION_NAME' : 'netops-inference',
          'PROJECT_NAME' : 'netops',
          'NETAPP_SIM' : NETAPP_SIM,
          'NETAPP_MOUNT_PATH': NETAPP_MOUNT_PATH,
          'NETAPP_PVC CLAIM' : NETAPP_PVC CLAIM,
          'IGZ_CONTAINER_PATH' : IGZ_CONTAINER_PATH,
          'IGZ_MOUNT_PATH' : IGZ_MOUNT_PATH
        }

```

After you have the function definition for all steps, you can construct the pipeline. We use the `kfp` module to make this definition. The difference between using MLRun and building on your own is the simplification and shortening of the coding.

The functions we defined are turned into step components using the `as_step` function of MLRun.

## Snapshot Step Definition

Initiate a Snapshot function, output, and mount v3io as source:

```

snap = snapfn.as_step(NewTask(handler='handler',params=snap_params),
name='NetApp_Cloud_Volume_Snapshot',outputs=['snapVolumeDetails','training
_parquet_file']).apply(mount_v3io())

```

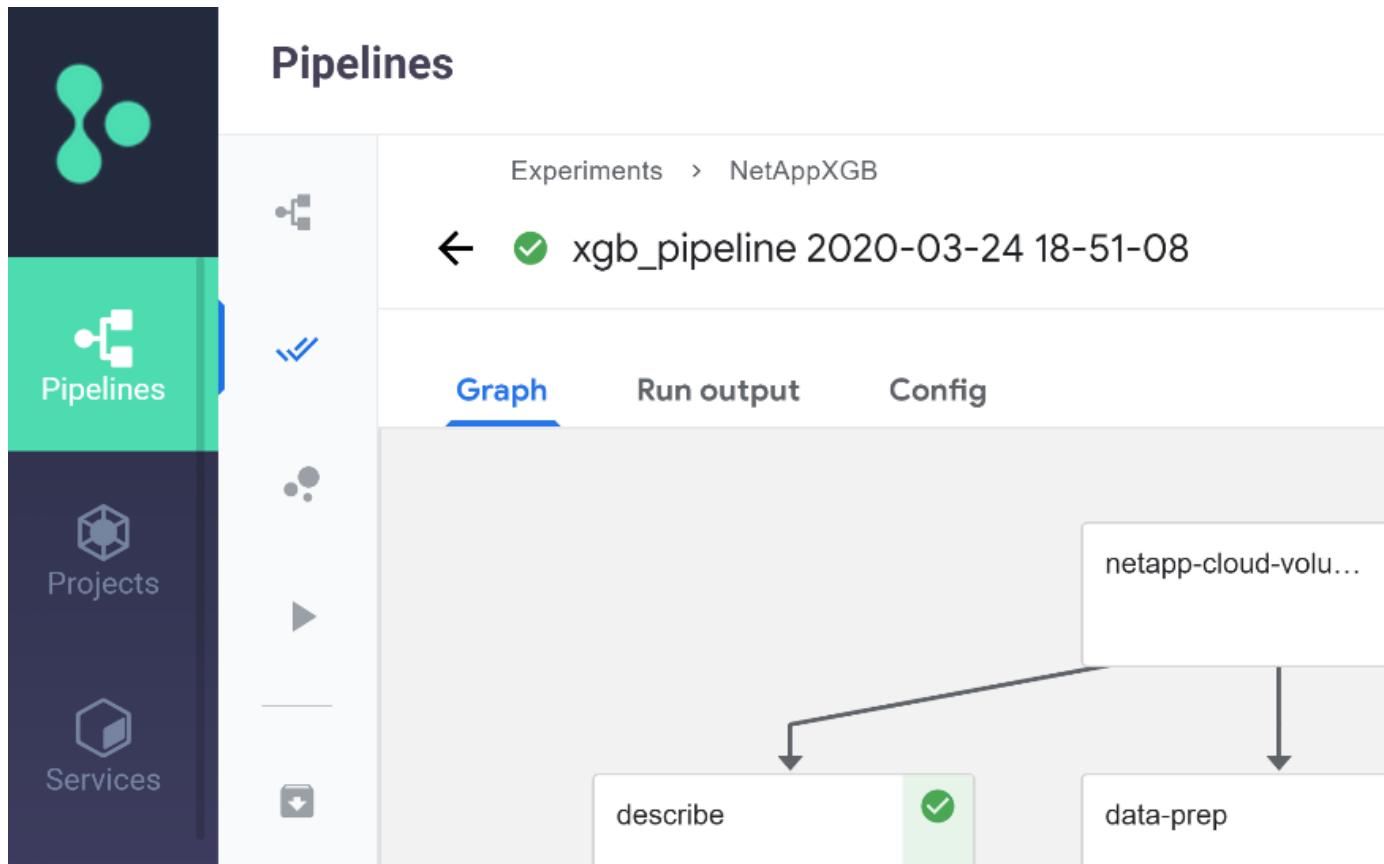
Parameters	Details
NewTask	NewTask is the definition of the function run.
(MLRun module)	Handler. Name of the Python function to invoke. We used the name <code>handler</code> in the notebook, but it is not required. params. The parameters we passed to the execution. Inside our code, we use <code>context.get_param('PARAMETER')</code> to get the values.

Parameters	Details
as_step	Name. Name of the Kubeflow pipeline step. outputs. These are the values that the step adds to the dictionary on completion. Take a look at the snap_cv.ipynb notebook. mount_v3io(). This configures the step to mount /User for the user executing the pipeline.

```
prep = data_prep.as_step(name='data-prep',
handler='handler', params=params,
                    inputs = {'DATA_DIR':
snap.outputs['snapVolumeDetails']} ,
out_path=artifacts_path).apply(mount_v3io()).after(snap)
```

Parameters	Details
inputs	You can pass to a step the outputs of a previous step. In this case, snap.outputs['snapVolumeDetails'] is the name of the Snapshot copy we created on the snap step.
out_path	A location to place artifacts generating using the MLRun module log_artifacts.

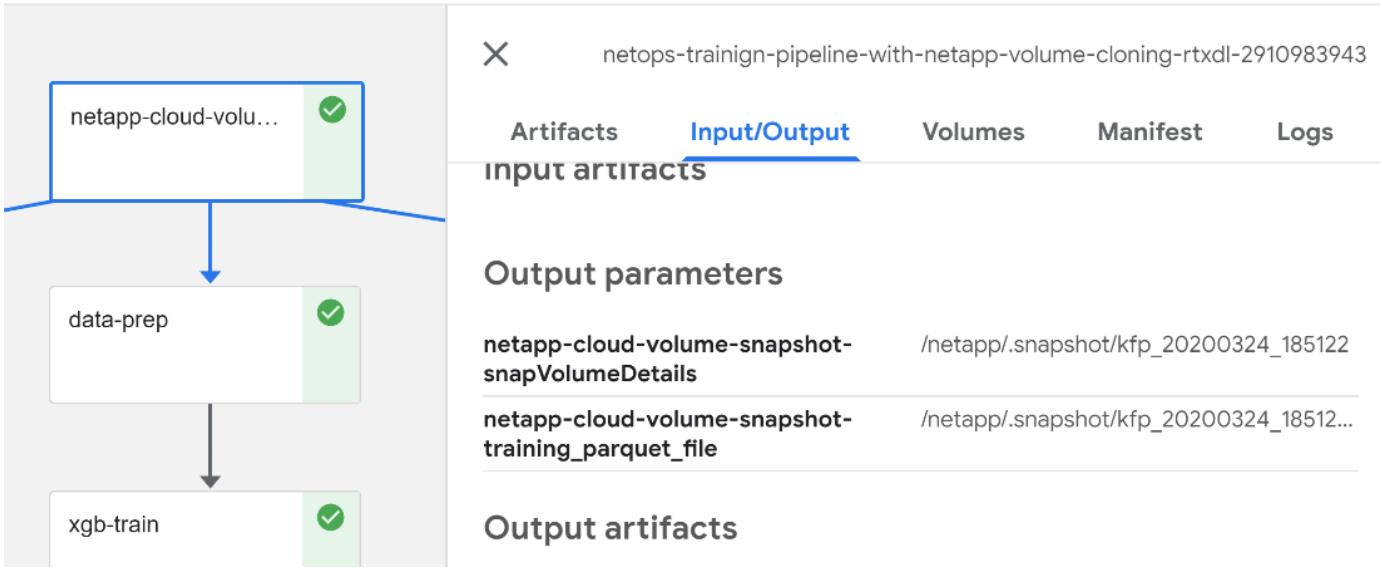
You can run [pipeline.ipynb](#) from top to bottom. You can then go to the Pipelines tab from the Iguazio dashboard to monitor progress as seen in the Iguazio dashboard Pipelines tab.



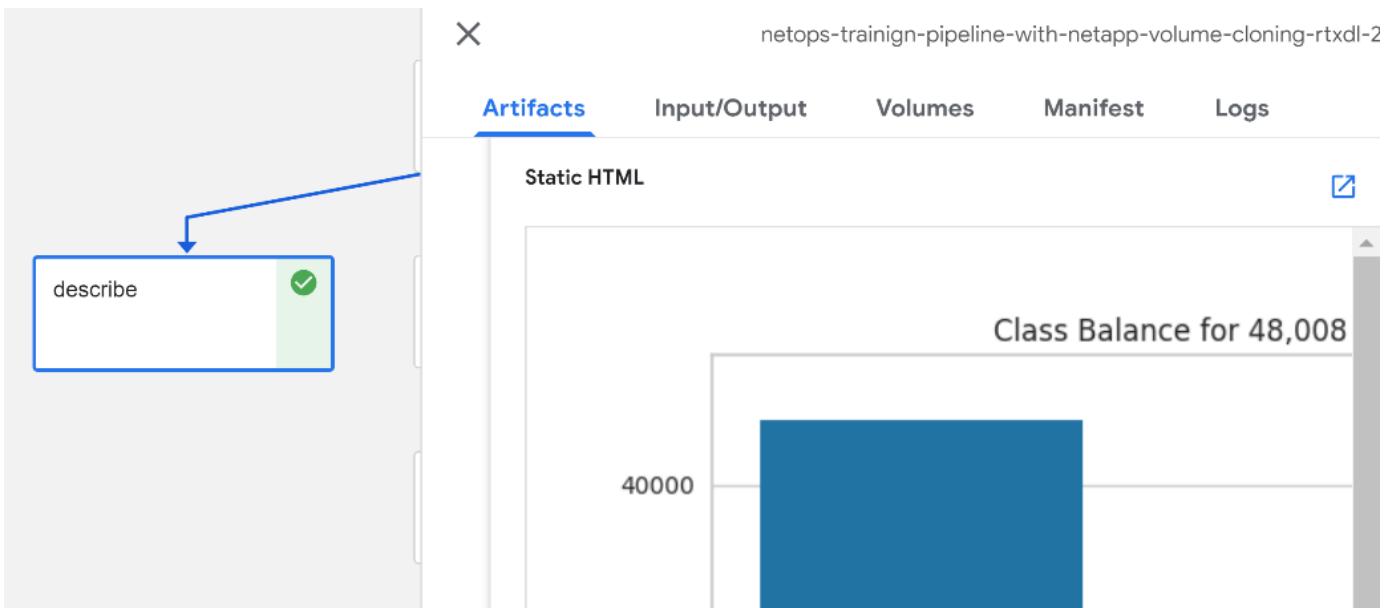
Because we logged the accuracy of training step in every run, we have a record of accuracy for each experiment, as seen in the record of training accuracy.

<input type="checkbox"/>	Run name	Status	Duration	Pipeline Version	Recurring ...	Start time	accuracy
<input type="checkbox"/>	xgb_pipeline 2020-03-24 18-51-...	✓	0:08:43	[View pipeline]	-	3/24/2020, 2:51:09 PM	0.985
<input type="checkbox"/>	xgb_pipeline 2020-03-19 13-31-...	✓	0:08:14	[View pipeline]	-	3/19/2020, 9:31:19 AM	0.980
<input type="checkbox"/>	xgb_pipeline 2020-03-18 12-56-...	✓	0:08:11	[View pipeline]	-	3/18/2020, 8:56:08 AM	0.990
<input type="checkbox"/>	xgb_pipeline 2020-03-17 19-49-...	✓	0:08:03	[View pipeline]	-	3/17/2020, 3:49:31 PM	0.985
<input type="checkbox"/>	xgb_pipeline 2020-03-17 18-34-...	✓	0:05:54	[View pipeline]	-	3/17/2020, 2:34:56 PM	0.980
<input type="checkbox"/>	xgb_pipeline 2020-03-17 17-34-...	✓	0:04:48	[View pipeline]	-	3/17/2020, 1:34:16 PM	0.982
<input type="checkbox"/>	xgb_pipeline 2020-03-17 17-01-...	✓	0:05:25	[View pipeline]	-	3/17/2020, 1:01:58 PM	0.987
<input type="checkbox"/>	xgb_pipeline 2020-03-16 16-47-...	✓	0:06:08	[View pipeline]	-	3/16/2020, 12:47:19 ...	0.983
<input type="checkbox"/>	xgb_pipeline 2020-03-16 13-57-...	✓	0:05:18	[View pipeline]	-	3/16/2020, 9:57:03 AM	0.980

If you select the Snapshot step, you can see the name of the Snapshot copy that was used to run this experiment.



The described step has visual artifacts to explore the metrics we used. You can expand to view the full plot as seen in the following image.



The MLRun API database also tracks inputs, outputs, and artifacts for each run organized by project. An example of inputs, outputs, and artifacts for each run can be seen in the following image.

The screenshot shows the MLRun UI interface. At the top, there's a dark header with the MLRun UI logo and a navigation bar with tabs for 'Projects', 'Jobs', 'Artifacts', and 'Logs'. Below the header, there are three project cards:

- NetApp**: Shows a green status indicator and a 'Jobs' tab.
- default**: Shows a green status indicator and a 'Jobs' tab.
- describe**: Shows a green status indicator and a 'Jobs' tab.

For each job, we store additional details.

The screenshot shows a detailed view of the 'describe' project card. On the left is a sidebar with a list of jobs:

- deploy-model** • 24 Mar, 14:56:03 ...bcbe38e
- xgb\_train** • 24 Mar, 14:53:18 ...5c85949
- data-prep** • 24 Mar, 14:52:46 ...126dc73
- describe** • 24 Mar, 14:52:45 ...c2a460e
- deploy-features-function** • 24 Mar, 14:52:43 ...50d8b83
- NetApp\_Cloud\_Volume\_Sna** • 24 Mar, 14:51:22 ...3108eb2

The main area shows the 'describe' job details:

describe				
24 Mar, 14:52:45 •				
Info	Inputs	Artifacts	Results	Logs
UID	66ef22187efb4ad89e8da8433c2a460e			
Start time	24 Mar, 14:52:45			
Parameters	Completed •			
Results	<a href="#">class_label...</a> ▾ <a href="#">key: summary</a> <a href="#">label_colu...</a> ▾			

There is more information about MLRun than we can cover in this document. AI artifacts, including the definition of the steps and functions, can be saved to the API database, versioned, and invoked individually or as a full project. Projects can also be saved and pushed to Git for later use. We encourage you to learn more at the [MLRun GitHub site](#).

Next: Deploy Grafana Dashboard

### Deploy Grafana Dashboard

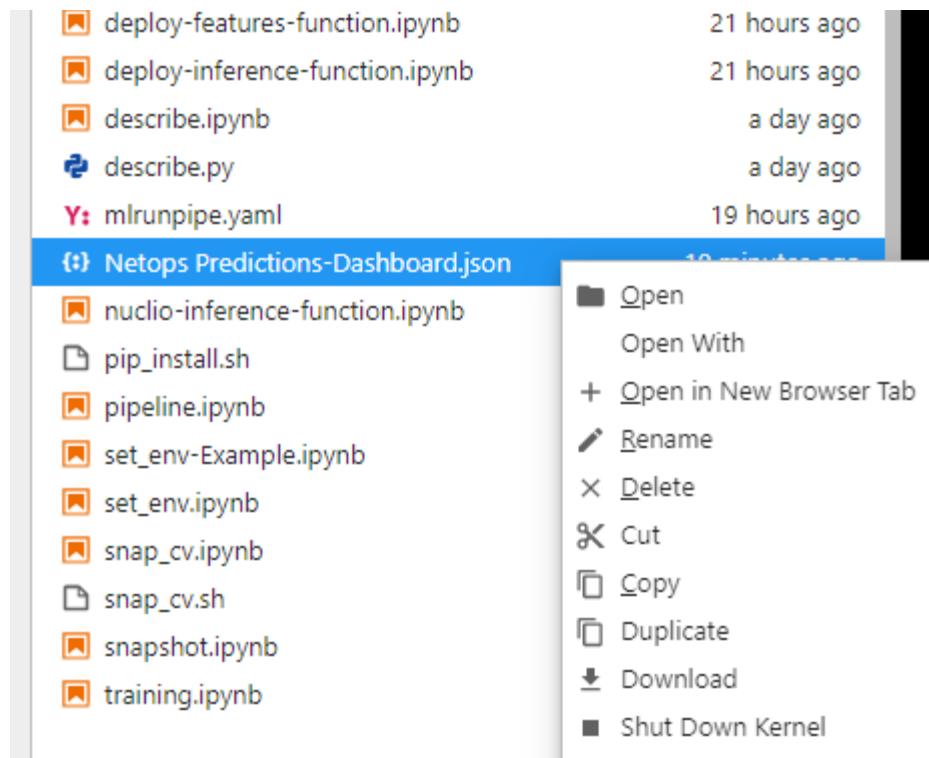
After everything is deployed, we run inferences on new data. The models predict failure on network device equipment. The results of the prediction are stored in an Iguazio TimeSeries table. You can visualize the results with Grafana in the platform integrated with Iguazio's security and data access policy.

You can deploy the dashboard by importing the provided JSON file into the Grafana interfaces in the cluster.

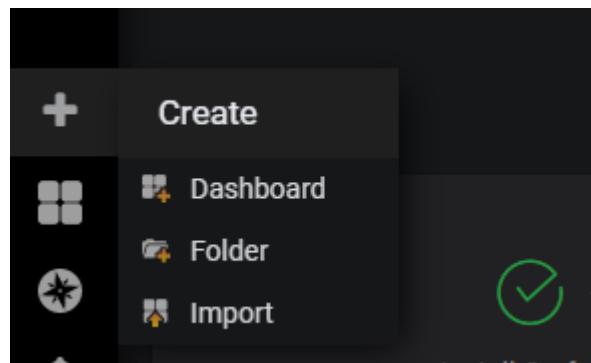
1. To verify that the Grafana service is running, look under Services.

Services							
Name	Running User	Version	CPU (cores)	Memory	AF	H	I
docker-registry Type: Docker Registry	root	2.7.1	96μ		1.67 GB		H1
framesd Type: V3IO Frame	root	0.6.10	369μ		795.19 MB		H1
grafana Type: Grafana	root	6.6.0	1m		38.39 MB		
jupyter Type: Jupyter Note	admin	1.0.2	81m		3.27 GB		
log-forwarder Type: Log forward	root	6.7.2	0		0 bytes		

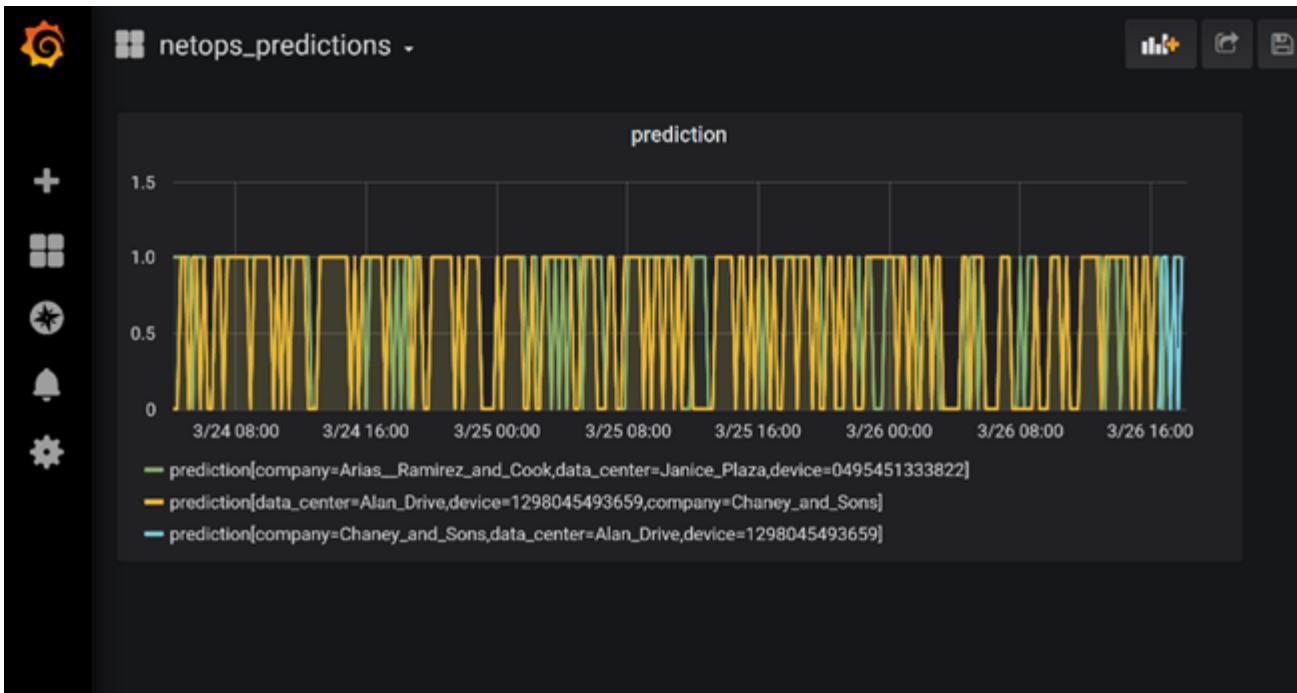
2. If it is not present, deploy an instance from the Services section:
  - a. Click New Service.
  - b. Select Grafana from the list.
  - c. Accept the defaults.
  - d. Click Next Step.
  - e. Enter your user ID.
  - f. Click Save Service.
  - g. Click Apply Changes at the top.
3. To deploy the dashboard, download the file [NetopsPredictions-Dashboard.json](#) through the Jupyter interface.



4. Open Grafana from the Services section and import the dashboard.



5. Click Upload \*.json File and select the file that you downloaded earlier (`NetopsPredictions-Dashboard.json`). The dashboard displays after the upload is completed.



## Deploy Cleanup Function

When you generate a lot of data, it is important to keep things clean and organized. To do so, deploy the cleanup function with the [cleanup.ipynb](#) notebook.

## Benefits

NetApp and Iguazio speed up and simplify the deployment of AI and ML applications by building in essential frameworks, such as Kubeflow, Apache Spark, and TensorFlow, along with orchestration tools like Docker and Kubernetes. By unifying the end-to-end data pipeline, NetApp and Iguazio reduce the latency and complexity inherent in many advanced computing workloads, effectively bridging the gap between development and operations. Data scientists can run queries on large datasets and securely share data and algorithmic models with authorized users during the training phase. After the containerized models are ready for production, you can easily move them from development environments to operational environments.

[Next: Conclusion](#)

## Conclusion

When building your own AI/ML pipelines, configuring the integration, management, security, and accessibility of the components in an architecture is a challenging task. Giving developers access and control of their environment presents another set of challenges.

The combination of NetApp and Iguazio brings these technologies together as managed services to accelerate technology adoption and improve the time to market for new AI/ML applications.

[Next: Where to Find Additional Information](#)

## Where to Find Additional Information

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

resources:

- NetApp AI Control Plane:
  - NetApp AI Control Plane Technical Report  
<https://www.netapp.com/us/media/tr-4798.pdf>
- NetApp persistent storage for containers:
  - NetApp Trident  
<https://netapp.io/persistent-storage-provisioner-for-kubernetes/>
- ML framework and tools:
  - TensorFlow: An Open-Source Machine Learning Framework for Everyone <https://www.tensorflow.org/>
  - Docker  
<https://docs.docker.com>
  - Kubernetes  
<https://kubernetes.io/docs/home/>
  - Kubeflow  
<http://www.kubeflow.org/>
  - Jupyter Notebook Server  
<http://www.jupyter.org/>
- Iguazio Data Science Platform
  - Iguazio Data Science Platform Documentation  
<https://www.iguazio.com/docs/>
  - Nuclio serverless function  
<https://nuclio.io/>
  - MLRun opensource pipeline orchestration framework  
<https://www.iguazio.com/open-source/mlrun/>
- NVIDIA DGX-1 systems
  - NVIDIA DGX-1 systems  
<https://www.nvidia.com/en-us/data-center/dgx-1/>
  - NVIDIA Tesla V100 Tensor core GPU  
<https://www.nvidia.com/en-us/data-center/tesla-v100/>
  - NVIDIA GPU Cloud

<https://www.nvidia.com/en-us/gpu-cloud/>

- NetApp AFF systems

- AFF datasheet

<https://www.netapp.com/us/media/ds-3582.pdf>

- NetApp Flash Advantage for AFF

<https://www.netapp.com/us/media/ds-3733.pdf>

- ONTAP 9.x documentation

<https://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>

- NetApp FlexGroup technical report

<https://www.netapp.com/us/media/tr-4557.pdf>

- NetApp ONTAP AI

- ONTAP AI with DGX-1 and Cisco Networking Design Guide

<https://www.netapp.com/us/media/nva-1121-design.pdf>

- ONTAP AI with DGX-1 and Cisco Networking Deployment Guide

<https://www.netapp.com/us/media/nva-1121-deploy.pdf>

- ONTAP AI with DGX-1 and Mellanox Networking Design Guide

<https://www.netapp.com/us/media/nva-1138-design.pdf>

- ONTAP AI networking

- Cisco Nexus 3232C Series Switches

<https://www.cisco.com/c/en/us/products/switches/nexus-3232c-switch/index.html>

- Mellanox Scale-Out SN2000 Ethernet Switch Series

[https://www.mellanox.com/page/products\\_dyn?product\\_family=251&mtag=sn2000](https://www.mellanox.com/page/products_dyn?product_family=251&mtag=sn2000)

## Use Cases

### Azure Kubernetes Service, Azure NetApp Files, RAPIDS AI, and Dask

#### TR-4904: Azure Kubernetes Service, Azure NetApp Files, RAPIDS AI, and Dask

Rick Huang, Verron Martina, Muneer Ahmad, NetApp

The work of a data scientist should be focused on the training and tuning of machine learning (ML) and artificial intelligence (AI) models. However, according to research by Google, data scientists spend approximately 80% of their time figuring out how to make their models work with enterprise applications and run at scale.

To manage end-to-end AI/ML projects, a wider understanding of enterprise components is needed. Although DevOps have taken over the definition, integration, and deployment, these types of components, ML operations target a similar flow that includes AI/ML projects. To get an idea of what an end-to-end AI/ML pipeline touches in the enterprise, see the following list of required components:

- Storage
- Networking
- Databases
- File systems
- Containers
- Continuous integration and continuous deployment (CI/CD) pipeline
- Integrated development environment (IDE)
- Security
- Data access policies
- Hardware
- Cloud
- Virtualization
- Data science toolsets and libraries

#### Target audience

The world of data science touches multiple disciplines in IT and business:

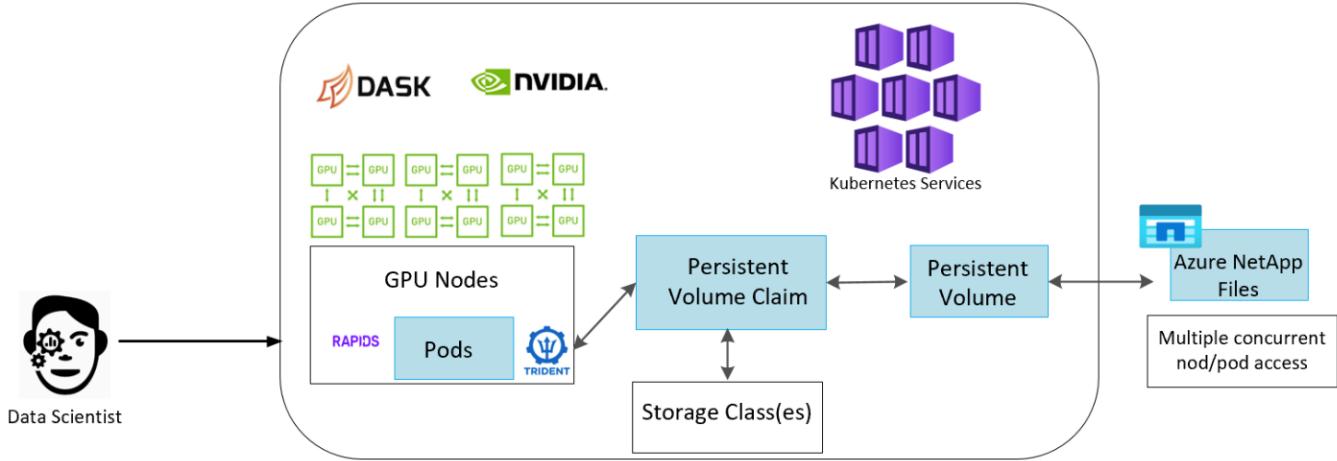
- The data scientist needs the flexibility to use their tools and libraries of choice.
- The data engineer needs to know how the data flows and where it resides.
- A DevOps engineer needs the tools to integrate new AI/ML applications into their CI/CD pipelines.
- Cloud administrators and architects need to be able to set up and manage Azure resources.
- Business users want to have access to AI/ML applications.

In this technical report, we describe how Azure NetApp Files, RAPIDS AI, Dask, and Azure help each of these roles bring value to business.

#### Solution overview

This solution follows the lifecycle of an AI/ML application. We start with the work of data scientists to define the different steps needed to prepare data and train models. By leveraging RAPIDS on Dask, we perform distributed training across the Azure Kubernetes Service (AKS) cluster to drastically reduce the training time when compared to the conventional Python scikit-learn approach. To complete the full cycle, we integrate the pipeline with Azure NetApp Files.

Azure NetApp Files provides various performance tiers. Customers can start with a Standard tier and scale out and scale up to a high-performance tier nondisruptively without moving any data. This capability enables data scientists to train models at scale without any performance issues, avoiding any data silos across the cluster, as shown in figure below.



[Next: Technology overview.](#)

## Technology overview

[Previous: Introduction.](#)

### Microsoft and NetApp

Since May 2019, Microsoft has delivered an Azure native, first-party portal service for enterprise NFS and SMB file services based on NetApp ONTAP technology. This development is driven by a strategic partnership between Microsoft and NetApp and further extends the reach of world-class ONTAP data services to Azure.

### Azure NetApp Files

The Azure NetApp Files service is an enterprise-class, high-performance, metered file storage service. Azure NetApp Files supports any workload type and is highly available by default. You can select service and performance levels and set up Snapshot copies through the service. Azure NetApp Files is an Azure first-party service for migrating and running the most demanding enterprise-file workloads in the cloud, including databases, SAP, and high-performance computing applications with no code changes.

This reference architecture gives IT organizations the following advantages:

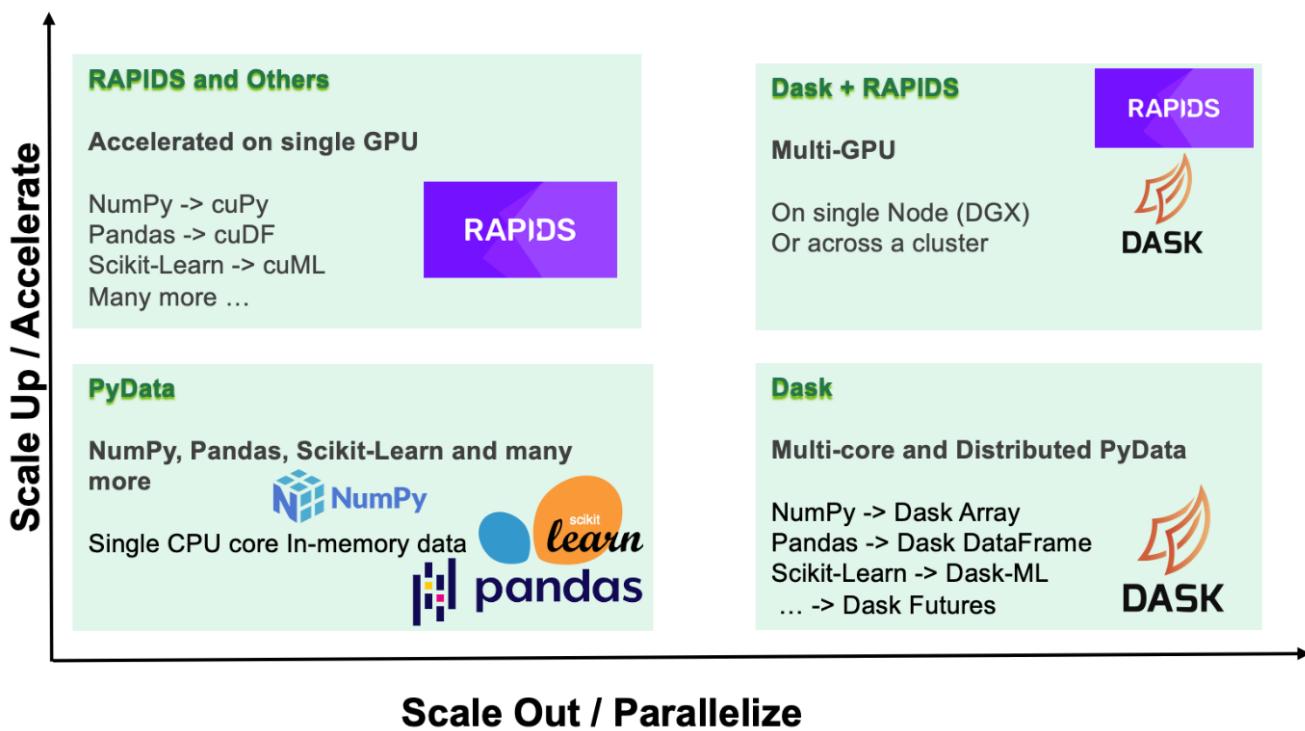
- Eliminates design complexities
- Enables independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage tiers for various performance and cost points

### Dask and NVIDIA RAPIDS overview

Dask is an open-source, parallel computing tool that scales Python libraries on multiple machines and provides faster processing of large amounts of data. It provides an API similar to single-threaded conventional Python libraries, such as Pandas, Numpy, and scikit-learn. As a result, native Python users are not forced to change much in their existing code to use resources across the cluster.

NVIDIA RAPIDS is a suite of open-source libraries that makes it possible to run end-to-end ML and data analytics workflows entirely on GPUs. Together with Dask, it enables you to easily scale from GPU workstation (scale up) to multinode, multi-GPU clusters (scale out).

For deploying Dask on a cluster, you could use Kubernetes for resource orchestration. You could also scale up or scale down the worker nodes as per the process requirement, which in-turn can help to optimize the cluster resource consumption, as shown in the following figure.



[Next: Software requirements.](#)

## Software requirements

[Previous: Technology overview.](#)

The following table lists the software requirements needed for this solution.

Software	Version
Azure Kubernetes Service	1.18.14
RAPIDS and Dask container image	Repository: "rapidsai/rapidsai" Tag: 0.17-cuda11.0-runtime-ubuntu18.04
NetApp Trident	20.01.1
Helm	3.0.0

[Next: Cloud resource requirements.](#)

## Cloud resource requirements

[Previous: Software requirements.](#)

### Configure Azure NetApp Files

Configure Azure NetApp Files as described in [QuickStart: Set up Azure NetApp Files and create an NFS volume.](#)

You can proceed past the section “Create NFS volume for Azure NetApp Files” because you are going to create volumes through Trident. Before continuing, complete the following steps:

1. Register for Azure NetApp Files and NetApp Resource Provider (through the Azure Shell) ([link](#)).
2. Create an account in Azure NetApp Files ([link](#)).
3. Set up a capacity pool (a minimum 4TB Standard or Premium, depending on your need) ([link](#)). The following table lists the network configuration requirements for setting up in the cloud. The Dask cluster and Azure NetApp Files must be in the same Azure Virtual Network (VNet) or a peered VNet.

Resources	Type/version
Azure Kubernetes Service	1.18.14
Agent node	3x Standard_DS2_v2
GPU node	3x Standard_NC6s_v3
Azure NetApp Files	Standard capacity pool
Capacity in TB	4

Next: [Click-through rate prediction use case summary](#).

## Click-through rate prediction use case summary

Previous: [Cloud resource requirements](#).

This use case is based on the publicly available [Terabyte Click Logs](#) dataset from [Criteo AI Lab](#). With the recent advances in ML platforms, a lot of attention is now on learning at scale. In this context, Terabyte Click Logs is now the dataset of reference in assessing the scalability of ML platforms and algorithms. By predicting the click-through rate, an advertiser can select the most plausible visitors who are most likely to respond to the ads, analyzing their browsing history, and showing the most relevant ads based on the interests of the user.

This solution provided in this technical report highlights the following benefits:

- Azure NetApp Files advantages in distributed or large-scale training
- RAPIDS CUDA-enabled data processing (cuDF, cuPy, and so on) and ML algorithms (cuML)
- Dask parallel computing framework for distributed training

The collaborative relationship between Dask and Azure NetApp Files demonstrates the drastic improvement in random forest model training time by two orders of magnitude. This improvement is comparable to the conventional Pandas approach when dealing with real-world click logs data with 45GB of text (on average) each day.

Next: [Install and set up the aks cluster](#).

## Setup

### Install and set up the AKS cluster

Previous: [Click-through rate prediction use case summary](#).

To install and set up the AKS cluster, see the webpage [Create an AKS Cluster](#) and then complete the following steps:

1. When selecting the type of node (system [CPU] or worker [GPU] nodes), select the following:
  - a. Primary system nodes should be Standard DS2v2 ([agentpool](#) default three nodes).
  - b. Then add the worker node Standard\_NC6s\_v3 pool (three nodes minimum) for the user group (for GPU nodes) named [gpupool](#).

Name	Mode	OS type	Node count	Node size
<input type="checkbox"/> agentpool	System	Linux	3	Standard_DS2_v2
<input type="checkbox"/> gpupool	User	Linux	3	Standard_NC6s_v

2. Deployment takes 5 to 10 minutes. After it is complete, click Connect to Cluster.
3. To connect to the newly created AKS cluster, install the following from your local environment (laptop/pc):
  - a. The Kubernetes command-line tool using the [instructions provided for your specific OS](#)
  - b. The Azure CLI as described in the document, [Install the Azure CLI](#)
4. To access the AKS cluster from the terminal, enter `az login` and enter the credentials.
5. Run the following two commands:

```
az account set --subscription xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
aks get-credentials --resource-group resourcegroup --name aksclustername
```

6. Enter [Azure CLI: kubectl get nodes](#).
7. If all six nodes are up and running, as shown in the following example, your AKS cluster is ready and connected to your local environment

```
verronmartina@verron-mac-0 ~ % kubectl get nodes
NAME                  STATUS   ROLES   AGE     VERSION
aks-agentpool-34613062-vmss000000  Ready    agent   22m    v1.18.14
aks-agentpool-34613062-vmss000001  Ready    agent   22m    v1.18.14
aks-agentpool-34613062-vmss000002  Ready    agent   22m    v1.18.14
aks-gpupool-34613062-vmss000000  Ready    agent   20m    v1.18.14
aks-gpupool-34613062-vmss000001  Ready    agent   20m    v1.18.14
aks-gpupool-34613062-vmss000002  Ready    agent   20m    v1.18.14
verronmartina@verron-mac-0 ~ %
```

[Next: Create a delegated subnet for Azure NetApp Files.](#)

[Create a delegated subnet for Azure NetApp Files](#)

[Previous: Install and set up the AKS cluster.](#)

To create a delegated subnet for Azure NetApp Files, complete the following steps:

1. Navigate to Virtual Networks within the Azure portal. Find your newly created virtual network. It should have a prefix such as [aks-vnet](#).
2. Click the name of the VNet.

Microsoft Azure Search resources, services, and docs (G+ /) ...

Dashboard > Virtual networks X

seanlucelive (Default Directory)

+ Add Manage view Refresh Export to CSV Open query | Assign tags | Feedback

Filter by name... Subscription == AzureSub01 Resource group == all X Location == all X Add filter

Showing 1 to 5 of 5 records. No grouping List view

Name ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
aks-vnet-22885919	MC_sluce.rg_TridentDemo_eastus2	East US 2	AzureSub01

3. Click Subnets and click +Subnet from the top toolbar.

Microsoft Azure Search resources, services, and docs (G+ /)

Dashboard > Virtual networks > aks-vnet-22885919

< aks-vnet-22885919 | Subnets X

Virtual network

Search (Ctrl+/) + Subnet Refresh Delete

Search subnets

Name ↑↓	IPv4 ↑↓	IPv6 (many availab... ↑↓	Delegated to ↑↓	Security group ↑↓
aks-subnet	10.240.0.0/16 (65530 av...	-	-	aks-agentpool-2288591... ***

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Address space Connected devices

4. Provide the subnet with a name such as **ANF.sn** and, under the Subnet Delegation heading, select **Microsoft.Netapp/volumes**. Do not change anything else. Click OK.

## Add subnet

X

Name \*

ANF.sn



Subnet address range \* ⓘ

10.0.0.0/24

10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

Add IPv6 address space ⓘ

NAT gateway ⓘ

None



Network security group

None



Route table

None



### SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ

0 selected



### SUBNET DELEGATION

Delegate subnet to a service ⓘ

Microsoft.Netapp/volumes



OK

Cancel

Azure NetApp Files volumes are allocated to the application cluster and are consumed as persistent volume claims (PVCs) in Kubernetes. In turn, this process provides you the flexibility to map them to different services, such as Jupyter notebooks, serverless functions, and so on.

Users of services can consume storage from the platform in many ways. As this technical report discusses NFSs, the main benefits of Azure NetApp Files are:

- Providing users with the ability to use Snapshot copies.
- Enabling users to store large quantities of data on Azure NetApp Files volumes.
- Using the performance benefits of Azure NetApp Files volumes when running their models on large sets of files.

Next: Peer AKS vnet and Azure NetApp Files vnet.

## Peer AKS VNet and Azure NetApp Files VNet

Previous: [Create a delegated subnet for Azure NetApp Files.](#)

To peer the AKS VNet to the Azure NetApp Files VNet, complete the following steps:

1. Enter Virtual Networks in the search field.
2. Select `vnet aks-vnet-name`. Click it and enter Peerings in the search field.
3. Click +Add.
4. Enter the following descriptors:
  - a. The peering link name is `aks-vnet-name_to_anf`.
  - b. subscriptionID and Azure NetApp Files VNet as the VNet peering partner.
  - c. Leave all the nonasterisk sections with the default values.
5. Click Add.

For more information, see [Create, change, or delete a virtual network peering](#).

Next: [Install Trident](#).

## Install Trident

Previous: [Peer AKS VNet and Azure NetApp Files VNet](#).

To install Trident using Helm, complete the following steps:

1. Install Helm (for installation instructions, visit the [source](#)).
2. Download and extract the Trident 20.01.1 installer.

```
$ wget  
$ tar -xf trident-installer-21.01.1.tar.gz
```

3. Change the directory to `trident-installer`.

```
$ cd trident-installer
```

4. Copy `tridentctl` to a directory in your system `$PATH`.

```
$ sudo cp ./tridentctl /usr/local/bin
```

5. Install Trident on the Kubernetes (K8s) cluster with Helm ([source](#)):

- a. Change the directory to the `helm` directory.

```
$ cd helm
```

b. Install Trident.

```
$ helm install trident trident-operator-21.01.1.tgz --namespace  
trident --create-namespace
```

c. Check the status of Trident pods.

```
$ kubectl -n trident get pods
```

If all the pods are up and running, then Trident is installed and you can move forward.

6. Set up the Azure NetApp Files backend and storage class for AKS.

a. Create an Azure Service Principle.

The service principal is how Trident communicates with Azure to manipulate your Azure NetApp Files resources.

```
$ az ad sp create-for-rbac --name ""
```

The output should look like the following example:

```
{  
  "appId": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",  
  "displayName": "netapptrident",  
  "name": "",  
  "password": "xxxxxxxxxxxxxxxx.xxxxxxxxxxxxxx",  
  "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"  
}
```

7. Create a Trident backend json file, example name [anf-backend.json](#).

8. Using your preferred text editor, complete the following fields inside the [anf-backend.json](#) file:

```
{
    "version": 1,
    "storageDriverName": "azure-netapp-files",
    "subscriptionID": "fakec765-4774-fake-ae98-a721add4fake",
    "tenantID": "fakef836-edc1-fake-bff9-b2d865eefake",
    "clientID": "fake0f63-bf8e-fake-8076-8de91e57fake",
    "clientSecret": "SECRET",
    "location": "westeurope",
    "serviceLevel": "Standard",
    "virtualNetwork": "anf-vnet",
    "subnet": "default",
    "nfsMountOptions": "vers=3,proto=tcp",
    "limitVolumeSize": "500Gi",
    "defaults": {
        "exportRule": "0.0.0.0/0",
        "size": "200Gi"
    }
}
```

9. Substitute the following fields:

- **subscriptionID**. Your Azure subscription ID.
- **tenantID**. Your Azure Tenant ID from the output of `az ad sp` in the previous step.
- **clientID**. Your appID from the output of `az ad sp` in the previous step.
- **clientSecret**. Your password from the output of `az ad sp` in the previous step.

10. Instruct Trident to create the Azure NetApp Files backend in the `trident` namespace using `anf-backend.json` as the configuration file:

```
$tridentctl create backend -f anf-backend.json -n trident
```

NAME	STORAGE DRIVER	UUID	STATE	VOLUMES
azurenappfiles_86181	azure-netapp-files	2ca85462-59ac-4946-be05-c03f5575a2ad	online	0

11. Create a storage class. Kubernetes users provision volumes by using PVCs that specify a storage class by name. Instruct K8s to create a storage class `azurenappfiles` that references the Trident backend created in the previous step.

12. Create a YAML (`anf-storage-class.yaml`) file for storage class and copy.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: azurenetaappfiles
  provisioner: netapp.io/trident
  parameters:
    backendType: "azure-netapp-files"
$kubectl create -f anf-storage-class.yaml
```

13. Verify that the storage class was created.

```
kubectl get sc azurenetaappfiles
```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
azurenetaappfiles	csi.trident.netapp.io	Delete	Immediate	false	98s

Next: [Set up Dask with RAPIDS deployment on AKS using Helm](#).

#### **Set up Dask with RAPIDS deployment on AKS using Helm**

[Previous: Install Trident.](#)

To set up Dask with RAPIDS deployment on AKS using Helm, complete the following steps:

1. Create a namespace for installing Dask with RAPIDS.

```
kubectl create namespace rapids-dask
```

2. Create a PVC to store the click-through rate dataset:

- a. Save the following YAML content to a file to create a PVC.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-criteo-data
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1000Gi
  storageClassName: azurenetaappfiles
```

- b. Apply the YAML file to your Kubernetes cluster.

```
kubectl -n rapids-dask apply -f <your yaml file>
```

3. Clone the `rapidsai` git repository (<https://github.com/rapidsai/helm-chart>).

```
git clone https://github.com/rapidsai/helm-chart helm-chart
```

4. Modify `values.yaml` and include the PVC created earlier for workers and Jupyter workspace.

- a. Go to the `rapidsai` directory of the repository.

```
cd helm-chart/rapidsai
```

- b. Update the `values.yaml` file and mount the volume using PVC.

```
dask:  
...  
worker:  
  name: worker  
...  
  mounts:  
    volumes:  
      - name: data  
        persistentVolumeClaim:  
          claimName: pvc-criteo-data  
    volumeMounts:  
      - name: data  
        mountPath: /data  
...  
jupyter:  
  name: jupyter  
...  
  mounts:  
    volumes:  
      - name: data  
        persistentVolumeClaim:  
          claimName: pvc-criteo-data  
    volumeMounts:  
      - name: data  
        mountPath: /data  
...
```

5. Go to the repository's home directory and deploy Dask with three worker nodes on AKS using Helm.

```
cd ..  
helm dep update rapidsai  
helm install rapids-dask --namespace rapids-dask rapidsai
```

[Next: Azure NetApp Files performance tiers.](#)

#### Azure NetApp Files performance tiers

[Previous: Set up Dask with RAPIDS deployment on AKS using Helm.](#)

You can change the service level of an existing volume by moving the volume to another capacity pool that uses the service level you want for the volume. This solution enables customers to start with a small dataset and small number of GPUs in Standard Tier and scale out or scale up to Premium Tier as the amount of data and GPUs increase. The Premium Tier offers four times the throughput per terabyte as the Standard Tier, and scale up is performed without having to move any data to change the service level of a volume.

#### Dynamically change the service level of a volume

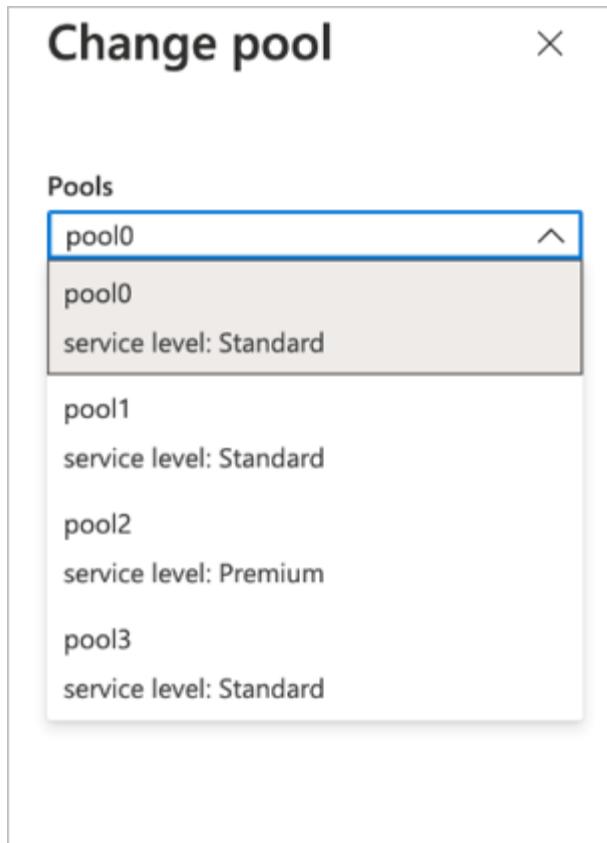
To dynamically change the service level of a volume, complete the following steps:

1. On the Volumes page, right-click the volume whose service level you want to change. Select Change Pool.

NFSv3	10.28.254.4:/norootfor...	Standard	pool0	...
NFSv4.1	NAS-735a.docs.lab:/fo...	Premium		
NFSv4.1	NAS-735a.docs.lab:/kr...	Premium		
NFSv3	10.28.254.4:/moveme0...	Premium		
NFSv3	10.28.254.4:/placeholder...	Premium		

A context menu is open over the second row of the table. The menu items are: Resize (with a resize icon), Edit (with a pencil icon), Change pool (with an upward arrow icon), and Delete (with a trash bin icon). A dashed blue rectangle highlights the "Change pool" option.

2. In the Change Pool window, select the capacity pool to which you want to move the volume.



3. Click OK.

## Automate performance tier change

The following options are available to automate performance tier changes:

- Dynamic Service Level change is still in Public Preview at this time and not enabled by default. To enable this feature on the Azure Subscription, see this documentation about how to [Dynamically change the service level of a volume](#).
- Azure CLI volume pool change commands are provided in [volume pool change documentation](#) and in the following example:

```
az netappfiles volume pool-change -g mygroup --account-name myaccname  
--pool-name mypoolname --name myvolname --new-pool-resource-id  
mynewresourceid
```

- PowerShell: The [Set-AzNetAppFilesVolumePool cmdlet](#) changes the pool of an Azure NetApp Files volume and is shown in the following example:

```

Set-AzNetAppFilesVolumePool
-ResourceGroupName "MyRG"
-AccountName "MyAnfAccount"
-PoolName "MyAnfPool"
-Name "MyAnfVolume"
-NewPoolResourceId 7d6e4069-6c78-6c61-7bf6-c60968e45fbf

```

[Next: Libraries for data processing and model training.](#)

## Click through rate prediction data processing and model training

### Libraries for data processing and model training

[Previous: Azure NetApp Files performance tiers.](#)

The following table lists the libraries and frameworks that were used to build this task. All these components have been fully integrated with Azure's role-based access and security controls.

Libraries/framework	Description
Dask cuML	For ML to work on GPU, the <a href="#">cuML library</a> provides access to the RAPIDS cuML package with Dask. RAPIDS cuML implements popular ML algorithms, including clustering, dimensionality reduction, and regression approaches, with high-performance GPU-based implementations, offering speed-ups of up to 100x over CPU-based approaches.
Dask cuDF	cuDF includes various other functions supporting GPU-accelerated extract, transform, load (ETL), such as data subsetting, transformations, one-hot encoding, and more. The RAPIDS team maintains a <a href="#">dask-cudf library</a> that includes helper methods to use Dask and cuDF.
Scikit Learn	Scikit-learn provides dozens of built-in machine learning algorithms and models, called estimators. Each <a href="#">estimator</a> can be fitted to some data using its <a href="#">fit</a> method.

We used two notebooks to construct the ML pipelines for comparison; one is the conventional Pandas scikit-learn approach, and the other is distributed training with RAPIDS and Dask. Each notebook can be tested individually to see the performance in terms of time and scale. We cover each notebook individually to demonstrate the benefits of distributed training using RAPIDS and Dask.

[Next: Load Criteo Click Logs day 15 in Pandas and train a scikit-learn random forest model.](#)

### Load Criteo Click Logs day 15 in Pandas and train a scikit-learn random forest model

[Previous: Libraries for data processing and model training.](#)

This section describes how we used Pandas and Dask DataFrames to load Click Logs data from the Criteo

Terabyte dataset. The use case is relevant in digital advertising for ad exchanges to build users' profiles by predicting whether ads will be clicked or if the exchange isn't using an accurate model in an automated pipeline.

We loaded day 15 data from the Click Logs dataset, totaling 45GB. Running the following cell in Jupyter notebook [CTR-PandasRF-collated.ipynb](#) creates a Pandas DataFrame that contains the first 50 million rows and generates a scikit-learn random forest model.

```
%%time
import pandas as pd
import numpy as np
header = ['col'+str(i) for i in range (1,41)] #note that according to
criteo, the first column in the dataset is Click Through (CT). Consist of
40 columns
first_row_taken = 50_000_000 # use this in pd.read_csv() if your compute
resource is limited.
# total number of rows in day15 is 20B
# take 50M rows
"""
Read data & display the following metrics:
1. Total number of rows per day
2. df loading time in the cluster
3. Train a random forest model
"""
df = pd.read_csv(file, nrows=first_row_taken, delimiter='\t',
names=header)
# take numerical columns
df_sliced = df.iloc[:, 0:14]
# split data into training and Y
Y = df_sliced.pop('col1') # first column is binary (click or not)
# change df_sliced data types & fillna
df_sliced = df_sliced.astype(np.float32).fillna(0)
from sklearn.ensemble import RandomForestClassifier
# Random Forest building parameters
# n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
rf_model = RandomForestClassifier(max_depth=max_depth,
n_estimators=n_trees)
rf_model.fit(df_sliced, Y)
```

To perform prediction by using a trained random forest model, run the following paragraph in this notebook. We took the last one million rows from day 15 as the test set to avoid any duplication. The cell also calculates accuracy of prediction, defined as the percentage of occurrences the model accurately predicts whether a user clicks an ad or not. To review any unfamiliar components in this notebook, see the [official scikit-learn documentation](#).

```

# testing data, last 1M rows in day15
test_file = '/data/day_15_test'
with open(test_file) as g:
    print(g.readline())

# DataFrame processing for test data
test_df = pd.read_csv(test_file, delimiter='\t', names=header)
test_df_sliced = test_df.iloc[:, 0:14]
test_Y = test_df_sliced.pop('col1')
test_df_sliced = test_df_sliced.astype(np.float32).fillna(0)
# prediction & calculating error
pred_df = rf_model.predict(test_df_sliced)
from sklearn import metrics
# Model Accuracy
print("Accuracy:",metrics.accuracy_score(test_Y, pred_df))

```

[Next: Load Day 15 in Dask and train a Dask cuML random forest model.](#)

[Load Day 15 in Dask and train a Dask cuML random forest model](#)

[Previous: Load Criteo Click Logs day 15 in Pandas and train a scikit-learn random forest model.](#)

In a manner similar to the previous section, load Criteo Click Logs day 15 in Pandas and train a scikit-learn random forest model. In this example, we performed DataFrame loading with Dask cuDF and trained a random forest model in Dask cuML. We compared the differences in training time and scale in the section [“Training time comparison.”](#)

### criteo\_dask\_RF.ipynb

This notebook imports `numpy`, `cuml`, and the necessary `dask` libraries, as shown in the following example:

```

import cuml
from dask.distributed import Client, progress, wait
import dask_cudf
import numpy as np
import cudf
from cuml.dask.ensemble import RandomForestClassifier as cumlDaskRF
from cuml.dask.common import utils as dask_utils

```

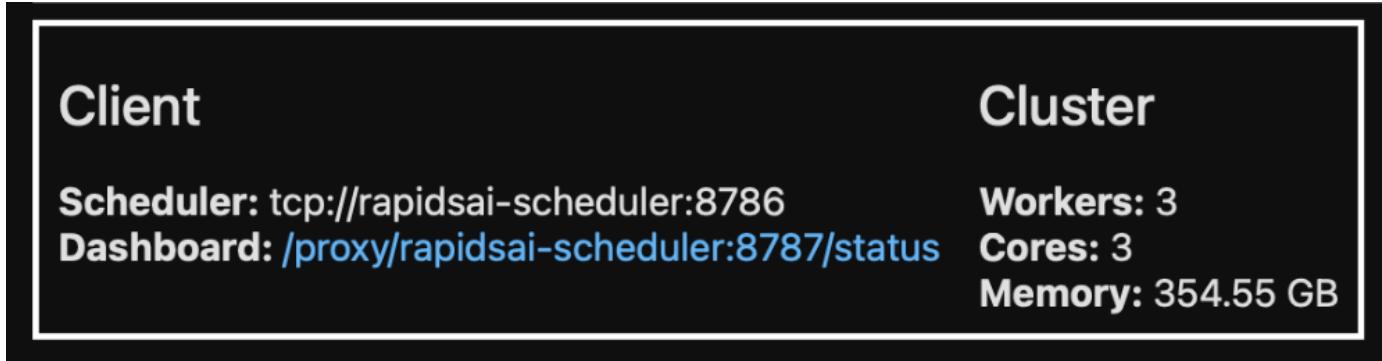
Initiate Dask Client().

```
client = Client()
```

If your cluster is configured correctly, you can see the status of worker nodes.

```
client
workers = client.has_what().keys()
n_workers = len(workers)
n_streams = 8 # Performance optimization
```

In our AKS cluster, the following status is displayed:



Note that Dask employs the lazy execution paradigm: rather than executing the processing code instantly, Dask builds a Directed Acyclic Graph (DAG) of execution instead. DAG contains a set of tasks and their interactions that each worker needs to run. This layout means the tasks do not run until the user tells Dask to execute them in one way or another. With Dask you have three main options:

- **Call `compute()` on a DataFrame.** This call processes all the partitions and then returns results to the scheduler for final aggregation and conversion to cuDF DataFrame. This option should be used sparingly and only on heavily reduced results unless your scheduler node runs out of memory.
- **Call `persist()` on a DataFrame.** This call executes the graph, but, instead of returning the results to the scheduler node, it maintains them across the cluster in memory so the user can reuse these intermediate results down the pipeline without the need for rerunning the same processing.
- **Call `head()` on a DataFrame.** Just like with cuDF, this call returns 10 records back to the scheduler node. This option can be used to quickly check if your DataFrame contains the desired output format, or if the records themselves make sense, depending on your processing and calculation.

Therefore, unless the user calls either of these actions, the workers sit idle waiting for the scheduler to initiate the processing. This lazy execution paradigm is common in modern parallel and distributed computing frameworks such as Apache Spark.

The following paragraph trains a random forest model by using Dask cuML for distributed GPU-accelerated computing and calculates model prediction accuracy.

```

Adsf

# Random Forest building parameters
n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
cuml_model = cumlDaskRF(max_depth=max_depth, n_estimators=n_trees,
n_bins=n_bins, n_streams=n_streams, verbose=True, client=client)
cuml_model.fit(gdf_sliced_small, Y)
# Model prediction
pred_df = cuml_model.predict(gdf_test)
# calculate accuracy
cu_score = cuml.metrics.accuracy_score( test_y, pred_df )

```

[Next: Monitor Dask using native Task Streams dashboard.](#)

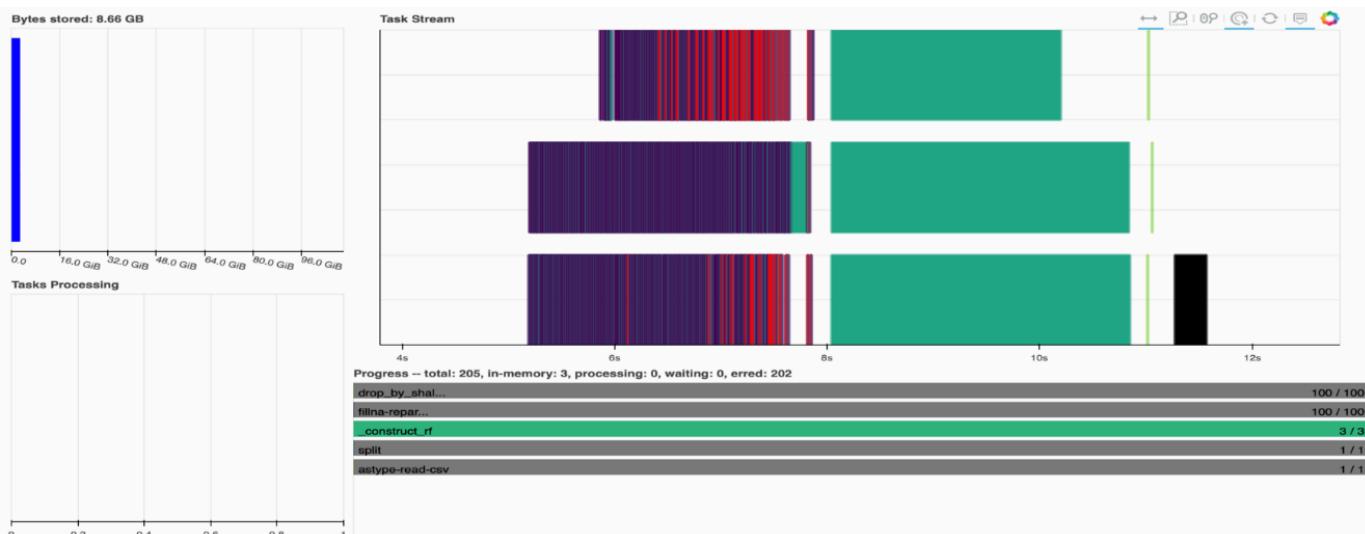
#### Monitor Dask using native Task Streams dashboard

[Previous: Load Day 15 in Dask and train a Dask cuML random forest model.](#)

The [Dask distributed scheduler](#) provides live feedback in two forms:

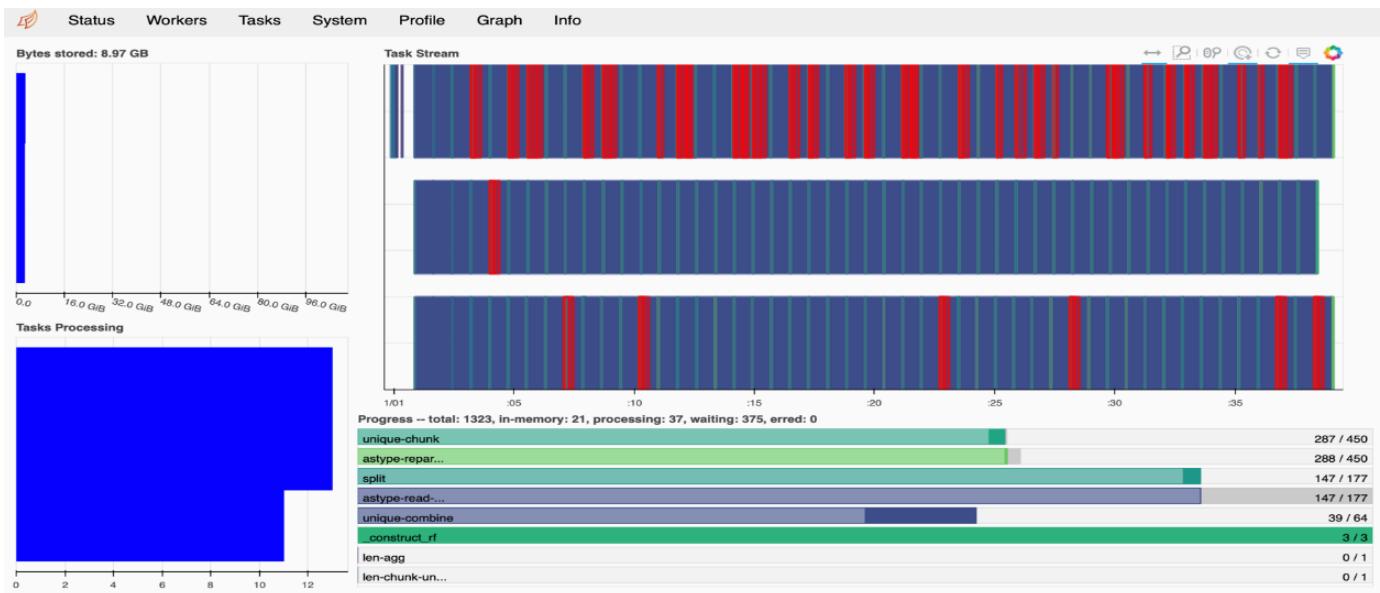
- An interactive dashboard containing many plots and tables with live information
- A progress bar suitable for interactive use in consoles or notebooks

In our case, the following figure shows how you can monitor the task progress, including Bytes Stored, the Task Stream with a detailed breakdown of the number of streams, and Progress by task names with associated functions executed. In our case, because we have three worker nodes, there are three main chunks of stream and the color codes denote different tasks within each stream.



You have the option to analyze individual tasks and examine the execution time in milliseconds or identify any obstacles or hindrances. For example, the following figure shows the Task Streams for the random forest model fitting stage. There are considerably more functions being executed, including unique chunk for DataFrame processing, `_construct_rf` for fitting the random forest, and so on. Most of the time was spent on

DataFrame operations due to the large size (45GB) of one day of data from the Criteo Click Logs.



[Next: Training time comparison.](#)

#### Training time comparison

[Previous: Monitor Dask using native Task Streams dashboard.](#)

This section compares the model training time using conventional Pandas compared to Dask. For Pandas, we loaded a smaller amount of data due to the nature of slower processing time to avoid memory overflow. Therefore, we interpolated the results to offer a fair comparison.

The following table shows the raw training time comparison when there is significantly less data used for the Pandas random forest model (50 million rows out of 20 billion per day15 of the dataset). This sample is only using less than 0.25% of all available data. Whereas for Dask-cuML we trained the random forest model on all 20 billion available rows. The two approaches yielded comparable training time.

Approach	Training time
Scikit-learn: Using only 50M rows in day15 as the training data	47 minutes and 21 seconds
RAPIDS-Dask: Using all 20B rows in day15 as the training data	1 hour, 12 minutes, and 11 seconds

If we interpolate the training time results linearly, as shown in the following table, there is a significant advantage to using distributed training with Dask. It would take the conventional Pandas scikit-learn approach 13 days to process and train 45GB of data for a single day of click logs, whereas the RAPIDS-Dask approach processes the same amount of data 262.39 times faster.

Approach	Training time
Scikit-learn: Using all 20B rows in day15 as the training data	13 days, 3 hours, 40 minutes, and 11 seconds
RAPIDS-Dask: Using all 20B rows in day15 as the training data	1 hour, 12 minutes, and 11 seconds

In the previous table, you can see that by using RAPIDS with Dask to distribute the data processing and model training across multiple GPU instances, the run time is significantly shorter compared to conventional Pandas DataFrame processing with scikit-learn model training. This framework enables scaling up and out in the cloud as well as on-premises in a multinode, multi-GPU cluster.

[Next: Monitor Dask and RAPIDS with Prometheus and Grafana.](#)

### **Monitor Dask and RAPIDS with Prometheus and Grafana**

[Previous: Training time comparison.](#)

After everything is deployed, run inferences on new data. The models predict whether a user clicks an ad based on browsing activities. The results of the prediction are stored in a Dask cuDF. You can monitor the results with Prometheus and visualize in Grafana dashboards.

For more information, see this [RAPIDS AI Medium post](#).

[Next: Dataset and Model Versioning using NetApp DataOps Toolkit.](#)

### **Dataset and model versioning using NetApp DataOps Toolkit**

[Previous: Monitor Dask and RAPIDS with Prometheus and Grafana.](#)

The NetApp DataOps Toolkit for Kubernetes abstracts storage resources and Kubernetes workloads up to the data-science workspace level. These capabilities are packaged in a simple, easy-to-use interface that is designed for data scientists and data engineers. Using the familiar form of a Python program, the Toolkit enables data scientists and engineers to provision and destroy JupyterLab workspaces in just seconds. These workspaces can contain terabytes, or even petabytes, of storage capacity, enabling data scientists to store all their training datasets directly in their project workspaces. Gone are the days of separately managing workspaces and data volumes.

For more information, visit the Toolkit's [GitHub repository](#).

[Next: Conclusion.](#)

## **Conclusion**

[Previous: Dataset and Model Versioning using NetApp DataOps Toolkit.](#)

Azure NetApp Files, RAPIDS, and Dask speed up and simplify the deployment of large-scale ML processing and training, integrated with orchestration tools such as Docker and Kubernetes. By unifying the end-to-end data pipeline, this solution reduces the latency and complexity inherent in many advanced computing workloads, effectively bridging the gap between development and operations. Data scientists can run queries on large datasets and securely share data and algorithmic models with other users during the training phase.

When building your own AI/ML pipelines, configuring the integration, management, security, and accessibility of the components in an architecture is a challenging task. Giving developers access and control of their environment presents another set of challenges.

The combination of NetApp and NVIDIA brings these technologies together as managed services to accelerate technology adoption and improve the time to market for new AI/ML applications. These advanced services are also delivered in a cloud native environment which can be easily ported for on-premises deployment.

[Next: Where to find additional information.](#)

## Where to find additional information

Previous: Conclusion.

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

- Azure NetApp Files:

- Solutions architecture page for Azure NetApp Files

<https://docs.microsoft.com/azure/azure-netapp-files/azure-netapp-files-solution-architectures>

- Trident persistent storage for containers:

- Azure NetApp Files and Trident

<https://netapptrident.readthedocs.io/en/stablev20.07/kubernetes/operations/tasks/backends/anf.html>

- Dask and RAPIDS:

- Dask

<https://docs.dask.org/en/latest/>

- Install Dask

<https://docs.dask.org/en/latest/install.html>

- Dask API

<https://docs.dask.org/en/latest/api.html>

- Dask Machine Learning

<https://examples.dask.org/machine-learning.html>

- Dask Distributed Diagnostics

<https://docs.dask.org/en/latest/diagnostics-distributed.html>

- ML framework and tools:

- TensorFlow: An Open-Source Machine Learning Framework for Everyone

<https://www.tensorflow.org/>

- Docker

<https://docs.docker.com>

- Kubernetes

<https://kubernetes.io/docs/home/>

- Kubeflow

<http://www.kubeflow.org/>

- Jupyter Notebook Server

<http://www.jupyter.org/>

Next: [Version history.](#)

## Version history

Previous: [Where to find additional information.](#)

Version	Date	Document version history
Version 1.0	August 2021	Initial release.

## TR-4896: Distributed training in Azure: Lane detection - Solution design

Muneer Ahmad and Verron Martina, NetApp  
Ronen Dar, RUN:AI

Since May 2019, Microsoft delivers an Azure native, first-party portal service for enterprise NFS and SMB file services based on NetApp ONTAP technology. This development is driven by a strategic partnership between Microsoft and NetApp and further extends the reach of world-class ONTAP data services to Azure.

NetApp, a leading cloud data services provider, has teamed up with RUN: AI, a company virtualizing AI infrastructure, to allow faster AI experimentation with full GPU utilization. The partnership enables teams to speed up AI by running many experiments in parallel, with fast access to data, and leveraging limitless compute resources. RUN: AI enables full GPU utilization by automating resource allocation, and the proven architecture of Azure NetApp Files enables every experiment to run at maximum speed by eliminating data pipeline obstructions.

NetApp and RUN: AI have joined forces to offer customers a future-proof platform for their AI journey in Azure. From analytics and high-performance computing (HPC) to autonomous decisions (where customers can optimize their IT investments by only paying for what they need, when they need it), the alliance between NetApp and RUN: AI offers a single unified experience in the Azure Cloud.

## Solution overview

In this architecture, the focus is on the most computationally intensive part of the AI or machine learning (ML) distributed training process of lane detection. Lane detection is one of the most important tasks in autonomous driving, which helps to guide vehicles by localization of the lane markings. Static components like lane markings guide the vehicle to drive on the highway interactively and safely.

Convolutional Neural Network (CNN)-based approaches have pushed scene understanding and segmentation to a new level. Although it doesn't perform well for objects with long structures and regions that could be occluded (for example, poles, shade on the lane, and so on). Spatial Convolutional Neural Network (SCNN) generalizes the CNN to a rich spatial level. It allows information propagation between neurons in the same layer, which makes it best suited for structured objects such as lanes, poles, or truck with occlusions. This compatibility is because the spatial information can be reinforced, and it preserves smoothness and continuity.

Thousands of scene images need to be injected in the system to allow the model learn and distinguish the various components in the dataset. These images include weather, daytime or nighttime, multilane highway roads, and other traffic conditions.

For training, there is a need for good quality and quantity of data. Single GPU or multiple GPUs can take days

to weeks to complete the training. Data-distributed training can speed up the process by using multiple and multinode GPUs. Horovod is one such framework that grants distributed training but reading data across clusters of GPUs could act as a hindrance. Azure NetApp Files provides ultrafast, high throughput and sustained low latency to provide scale-out/scale-up capabilities so that GPUs are leveraged to the best of their computational capacity. Our experiments verified that all the GPUs across the cluster are used more than 96% on average for training the lane detection using SCNN.

## Target audience

Data science incorporates multiple disciplines in IT and business, therefore multiple personas are part of our targeted audience:

- Data scientists need the flexibility to use the tools and libraries of their choice.
- Data engineers need to know how the data flows and where it resides.
- Autonomous driving use-case experts.
- Cloud administrators and architects to set up and manage cloud (Azure) resources.
- A DevOps engineer needs the tools to integrate new AI/ML applications into their continuous integration and continuous deployment (CI/CD) pipelines.
- Business users want to have access to AI/ML applications.

In this document, we describe how Azure NetApp Files, RUN: AI, and Microsoft Azure help each of these roles bring value to business.

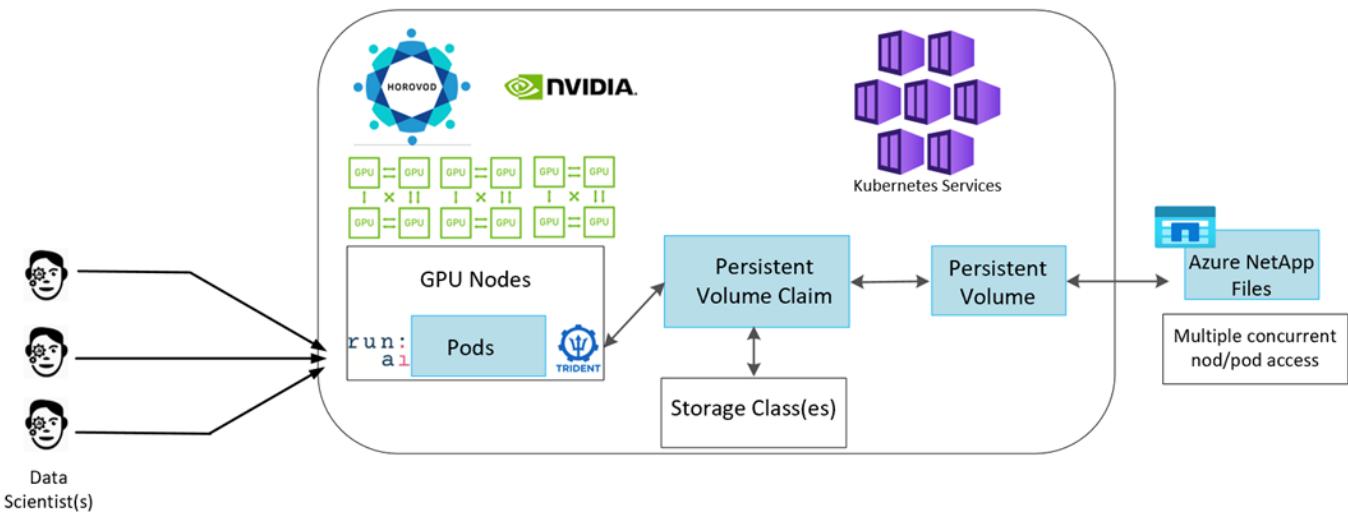
## Solution technology

This section covers the technology requirements for the lane detection use case by implementing a distributed training solution at scale that fully runs in the Azure cloud. The figure below provides an overview of the solution architecture.

The elements used in this solution are:

- Azure Kubernetes Service (AKS)
- Azure Compute SKUs with NVIDIA GPUs
- Azure NetApp Files
- RUN: AI
- NetApp Trident

Links to all the elements mentioned here are listed in the [Additional information](#) section.



### Cloud resources and services requirements

The following table lists the hardware components that are required to implement the solution. The cloud components that are used in any implementation of the solution might vary based on customer requirements.

Cloud	Quantity
AKS	Minimum of three system nodes and three GPU worker nodes
Virtual machine (VM) SKU system nodes	Three Standard_DS2_v2
VM SKU GPU worker nodes	Three Standard_NC6s_v3
Azure NetApp Files	4TB standard tier

### Software requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any implementation of the solution might vary based on customer requirements.

Software	Version or other information
AKS - Kubernetes version	1.18.14
RUN:AI CLI	v2.2.25
RUN:AI Orchestration Kubernetes Operator version	1.0.109
Horovod	0.21.2
NetApp Trident	20.01.1
Helm	3.0.0

### Lane detection – Distributed training with RUN:AI

This section provides details on setting up the platform for performing lane detection distributed training at scale using the RUN: AI orchestrator. We discuss installation of all the solution elements and running the distributed training job on the said platform. ML versioning is completed by using NetApp SnapshotTM linked with RUN: AI experiments for achieving data and model reproducibility. ML versioning plays a crucial role in

tracking models, sharing work between team members, reproducibility of results, rolling new model versions to production, and data provenance. NetApp ML version control (Snapshot) can capture point-in-time versions of the data, trained models, and logs associated with each experiment. It has rich API support making it easy to integrate with the RUN: AI platform; you just have to trigger an event based on the training state. You also have to capture the state of the whole experiment without changing anything in the code or the containers running on top of Kubernetes (K8s).

Finally, this technical report wraps up with performance evaluation on multiple GPU-enabled nodes across AKS.

#### Distributed training for lane detection use case using the TuSimple dataset

In this technical report, distributed training is performed on the TuSimple dataset for lane detection. Horovod is used in the training code for conducting data distributed training on multiple GPU nodes simultaneously in the Kubernetes cluster through AKS. Code is packaged as container images for TuSimple data download and processing. Processed data is stored on persistent volumes allocated by NetApp Trident plug-in. For the training, one more container image is created, and it uses the data stored on persistent volumes created during downloading the data.

To submit the data and training job, use RUN: AI for orchestrating the resource allocation and management. RUN: AI allows you to perform Message Passing Interface (MPI) operations which are needed for Horovod. This layout allows multiple GPU nodes to communicate with each other for updating the training weights after every training mini batch. It also enables monitoring of training through the UI and CLI, making it easy to monitor the progress of experiments.

NetApp Snapshot is integrated within the training code and captures the state of data and the trained model for every experiment. This capability enables you to track the version of data and code used, and the associated trained model generated.

#### AKS setup and installation

For setup and installation of the AKS cluster go to [Create an AKS Cluster](#). Then, follow these series of steps:

1. When selecting the type of nodes (whether it be system (CPU) or worker (GPU) nodes), select the following:
  - a. Add primary system node named `agentpool` at the `Standard_DS2_v2` size. Use the default three nodes.
  - b. Add worker node `gpupool` with `the Standard_NC6s_v3` pool size. Use three nodes minimum for GPU nodes.

Name	Mode	OS type	Node count	Node size
<input type="checkbox"/> agentpool	System	Linux	3	Standard_DS2_v2
<input type="checkbox"/> gpupool	User	Linux	3	Standard_NC6s_v3



Deployment takes 5–10 minutes.

2. After deployment is complete, click Connect to Cluster. To connect to the newly created AKS cluster, install the Kubernetes command-line tool from your local environment (laptop/PC). Visit [Install Tools](#) to install it as

per your OS.

3. [Install Azure CLI on your local environment](#).
4. To access the AKS cluster from the terminal, first enter `az login` and put in the credentials.
5. Run the following two commands:

```
az account set --subscription xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  
aks get-credentials --resource-group resourcegroup --name aksclustername
```

6. Enter this command in the Azure CLI:

```
kubectl get nodes
```



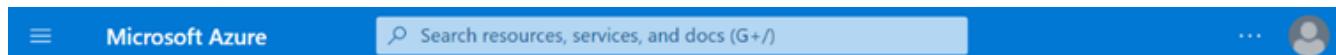
If all six nodes are up and running as seen here, your AKS cluster is ready and connected to your local environment.

```
verronmartina@verron-mac-0 ~ % kubectl get nodes  
NAME                      STATUS   ROLES      AGE     VERSION  
aks-agentpool-34613062-vmss000000  Ready    agent      22m    v1.18.14  
aks-agentpool-34613062-vmss000001  Ready    agent      22m    v1.18.14  
aks-agentpool-34613062-vmss000002  Ready    agent      22m    v1.18.14  
aks-gpupool-34613062-vmss000000  Ready    agent      20m    v1.18.14  
aks-gpupool-34613062-vmss000001  Ready    agent      20m    v1.18.14  
aks-gpupool-34613062-vmss000002  Ready    agent      20m    v1.18.14  
verronmartina@verron-mac-0 ~ %
```

#### Create a delegated subnet for Azure NetApp Files

To create a delegated subnet for Azure NetApp Files, follow this series of steps:

1. Navigate to Virtual networks within the Azure portal. Find your newly created virtual network. It should have a prefix such as aks-vnet, as seen here. Click the name of the virtual network.



Dashboard >

## Virtual networks ✖

seanluclive (Default Directory)

+ Add ⚙️ Manage view ▼ ⟳ Refresh ⬇️ Export to CSV 🔗 Open query | 🏷️ Assign tags | 💡 Feedback

Filter by name...

Subscription == AzureSub01

Resource group == all ✖

Location == all ✖

➕ 💡 Add filter

Showing 1 to 5 of 5 records.

No grouping ▼

List view ▼

<span style="color: blue;">checkbox</span> Name ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓	...
<span style="color: blue;">checkbox</span> <a href="#">aks-vnet-22885919</a>	MC_sluce.rg_TridentDemo_eastus2	East US 2	AzureSub01	<span style="color: grey;">...</span>

2. Click Subnets and select +Subnet from the top toolbar.

This screenshot shows the 'Subnets' page for the virtual network 'aks-vnet-22885919'. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Address space, Connected devices, and Subnets (which is highlighted with a red box). The main content area has a search bar and a toolbar with '+ Subnet' (highlighted with a red box), 'Gateway subnet', 'Refresh', 'Manage users', and 'Delete'. A table lists existing subnets: 'aks-subnet' with IP range '10.240.0.0/16 (65530 av...)' and security group 'aks-agentpool-2288591...'.

Name ↑↓	IPv4 ↑↓	IPv6 (many availab... ↑↓	Delegated to ↑↓	Security group ↑↓
aks-subnet	10.240.0.0/16 (65530 av...)	-	-	aks-agentpool-2288591...

3. Provide the subnet with a name such as `ANF.sn` and under the Subnet Delegation heading, select Microsoft.NetApp/volumes. Do not change anything else. Click OK.

## Add subnet

X

Name \*

ANF.sn



Subnet address range \* ⓘ

10.0.0.0/24

10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

Add IPv6 address space ⓘ

NAT gateway ⓘ

None



Network security group

None



Route table

None



### SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ

0 selected



### SUBNET DELEGATION

Delegate subnet to a service ⓘ

Microsoft.Netapp/volumes



OK

Cancel

Azure NetApp Files volumes are allocated to the application cluster and are consumed as persistent volume claims (PVCs) in Kubernetes. In turn, this allocation provides us the flexibility to map volumes to different services, be it Jupyter notebooks, serverless functions, and so on

Users of services can consume storage from the platform in many ways. The main benefits of Azure NetApp Files are:

- Provides users with the ability to use snapshots.
- Enables users to store large quantities of data on Azure NetApp Files volumes.
- Procure the performance benefits of Azure NetApp Files volumes when running their models on large sets of files.

## Azure NetApp Files setup

To complete the setup of Azure NetApp Files, you must first configure it as described in [Quickstart: Set up Azure NetApp Files and create an NFS volume](#).

However, you may omit the steps to create an NFS volume for Azure NetApp Files as you will create volumes through Trident. Before continuing, be sure that you have:

1. [Registered for Azure NetApp Files and NetApp Resource Provider \(through the Azure Cloud Shell\)](#).
2. [Created an account in Azure NetApp Files](#).
3. [Set up a capacity pool \(minimum 4TiB Standard or Premium depending on your needs\)](#).

## Peering of AKS virtual network and Azure NetApp Files virtual network

Next, peer the AKS virtual network (VNet) with the Azure NetApp Files VNet by following these steps:

1. In the search box at the top of the Azure portal, type virtual networks.
2. Click VNet aks- vnet-name, then enter Peerings in the search field.
3. Click +Add and enter the information provided in the table below:

Field	Value or description
Peering link name	aks-vnet-name_to_anf
SubscriptionID	Subscription of the Azure NetApp Files VNet to which you're peering
VNet peering partner	Azure NetApp Files VNet



Leave all the nonasterisk sections on default

4. Click ADD or OK to add the peering to the virtual network.

For more information, visit [Create, change, or delete a virtual network peering](#).

## Trident

Trident is an open-source project that NetApp maintains for application container persistent storage. Trident has been implemented as an external provisioner controller that runs as a pod itself, monitoring volumes and completely automating the provisioning process.

NetApp Trident enables smooth integration with K8s by creating and attaching persistent volumes for storing training datasets and trained models. This capability makes it easier for data scientists and data engineers to use K8s without the hassle of manually storing and managing datasets. Trident also eliminates the need for data scientists to learn managing new data platforms as it integrates the data management-related tasks through the logical API integration.

## Install Trident

To install Trident software, complete the following steps:

1. [First install helm](#).
2. Download and extract the Trident 21.01.1 installer.

```
wget  
https://github.com/NetApp/trident/releases/download/v21.01.1/trident-  
installer-21.01.1.tar.gz  
tar -xf trident-installer-21.01.1.tar.gz
```

3. Change the directory to `trident-installer`.

```
cd trident-installer
```

4. Copy `tridentctl` to a directory in your system `$PATH`.

```
cp ./tridentctl /usr/local/bin
```

5. Install Trident on K8s cluster with Helm:

- a. Change directory to helm directory.

```
cd helm
```

- b. Install Trident.

```
helm install trident trident-operator-21.01.1.tgz --namespace trident  
--create-namespace
```

- c. Check the status of Trident pods the usual K8s way:

```
kubectl -n trident get pods
```

- d. If all the pods are up and running, Trident is installed and you are good to move forward.

#### **Set up Azure NetApp Files back-end and storage class**

To set up Azure NetApp Files back-end and storage class, complete the following steps:

1. Switch back to the home directory.

```
cd ~
```

2. Clone the [project repository](#) `lane-detection-SCNN-horovod`.

3. Go to the `trident-config` directory.

```
cd ./lane-detection-SCNN-horovod/trident-config
```

4. Create an Azure Service Principle (the service principle is how Trident communicates with Azure to access your Azure NetApp Files resources).

```
az ad sp create-for-rbac --name
```

The output should look like the following example:

```
{  
    "appId": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",  
    "displayName": "netapprtrident",  
    "name": "http://netapprtrident",  
    "password": "xxxxxxxxxxxxxx.xxxxxxxxxxxxxx",  
    "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"  
}
```

5. Create the Trident `backend.json` file.
6. Using your preferred text editor, complete the following fields from the table below inside the `anf-backend.json` file.

Field	Value
subscriptionID	Your Azure Subscription ID
tenantID	Your Azure Tenant ID (from the output of <code>az ad sp</code> in the previous step)
clientID	Your appID (from the output of <code>az ad sp</code> in the previous step)
clientSecret	Your password (from the output of <code>az ad sp</code> in the previous step)

The file should look like the following example:

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "fakec765-4774-fake-ae98-a721add4fake",
  "tenantID": "fakef836-edc1-fake-bff9-b2d865eefake",
  "clientID": "fake0f63-bf8e-fake-8076-8de91e57fake",
  "clientSecret": "SECRET",
  "location": "westeurope",
  "serviceLevel": "Standard",
  "virtualNetwork": "anf-vnet",
  "subnet": "default",
  "nfsMountOptions": "vers=3,proto=tcp",
  "limitVolumeSize": "500Gi",
  "defaults": {
    "exportRule": "0.0.0.0/0",
    "size": "200Gi"
  }
}
```

7. Instruct Trident to create the Azure NetApp Files back- end in the `trident` namespace, using `anf-backend.json` as the configuration file as follows:

```
tridentctl create backend -f anf-backend.json -n trident
```

8. Create the storage class:

- a. K8 users provision volumes by using PVCs that specify a storage class by name. Instruct K8s to create a storage class `azurenappfiles` that will reference the Azure NetApp Files back end created in the previous step using the following:

```
kubectl create -f anf-storage-class.yaml
```

- b. Check that storage class is created by using the following command:

```
kubectl get sc azurenappfiles
```

The output should look like the following example:

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
azurenappfiles	csi.trident.netapp.io	Delete	Immediate	false	98s

#### Deploy and set up volume snapshot components on AKS

If your cluster does not come pre-installed with the correct volume snapshot components, you may manually install these components by running the following steps:



AKS 1.18.14 does not have pre-installed Snapshot Controller.

1. Install Snapshot Beta CRDs by using the following commands:

```
kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml  
kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml  
kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Install Snapshot Controller by using the following documents from GitHub:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml  
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

3. Set up K8s **volumesnapshotclass**: Before creating a volume snapshot, a **volume snapshot class** must be set up. Create a volume snapshot class for Azure NetApp Files, and use it to achieve ML versioning by using NetApp Snapshot technology. Create **volumesnapshotclass netapp-csi-snapclass** and set it to default `volumesnapshotclass` as such:

```
kubectl create -f netapp-volume-snapshot-class.yaml
```

The output should look like the following example:

```
volumesnapshotclass.snapshot.storage.k8s.io/netapp-csi-snapclass created
```

4. Check that the volume Snapshot copy class was created by using the following command:

```
kubectl get volumesnapshotclass
```

The output should look like the following example:

NAME	DRIVER	DELETIONPOLICY	AGE
netapp-csi-snapclass	csi.trident.netapp.io	Delete	63s

## RUN:AI installation

To install RUN:AI, complete the following steps:

1. [Install RUN:AI cluster on AKS](#).
2. Go to app.runai.ai, click create New Project, and name it lane-detection. It will create a namespace on a K8s cluster starting with `runai-` followed by the project name. In this case, the namespace created would be `runai-lane-detection`.

The screenshot shows the 'New Project' creation interface. On the left, there's a sidebar with tabs: 'Basics' (selected), 'Node Affinity', and 'Time Limit'. The main area is titled 'Basics' and contains the following fields:

- 'Project Name' input field containing 'lane-detection'.
- 'Assigned GPUs' input field containing '3'.
- A note 'Over-quota for project' with a checked checkbox labeled 'Allow over-quota'.

At the bottom right are 'Save' and 'Cancel' buttons.

3. [Install RUN:AI CLI](#).
4. On your terminal, set lane-detection as a default RUN: AI project by using the following command:

```
`runai config project lane-detection`
```

The output should look like the following example:

```
Project lane-detection has been set as default project
```

5. Create ClusterRole and ClusterRoleBinding for the project namespace (for example, `lane-detection`) so the default service account belonging to `runai-lane-detection` namespace has permission to perform `volumesnapshot` operations during job execution:

- a. List namespaces to check that `runai-lane-detection` exists by using this command:

```
kubectl get namespaces
```

The output should appear like the following example:

NAME	STATUS	AGE
default	Active	130m
kube-node-lease	Active	130m
kube-public	Active	130m
kube-system	Active	130m
runai	Active	4m44s
runai-lane-detection	Active	13s
trident	Active	102m

6. Create ClusterRole `netappsnapshot` and ClusterRoleBinding `netappsnapshot` using the following commands:

```
`kubectl create -f runai-project-snap-role.yaml`  
`kubectl create -f runai-project-snap-role-binding.yaml`
```

#### Download and process the TuSimple dataset as RUN:AI job

The process to download and process the TuSimple dataset as a RUN: AI job is optional. It involves the following steps:

1. Build and push the docker image, or omit this step if you want to use an existing docker image (for example, `muneer7589/download-tusimple:1.0`)

- a. Switch to the home directory:

```
cd ~
```

- b. Go to the data directory of the project `lane-detection-SCNN-horovod`:

```
cd ./lane-detection-SCNN-horovod/data
```

- c. Modify `build_image.sh` shell script and change docker repository to yours. For example, replace `muneer7589` with your docker repository name. You could also change the docker image name and

TAG (such as `download-tusimple` and `1.0`):

```
#!/bin/bash
#
# A simple script to build the Docker image.
#
# $ build_image.sh
set -ex

IMAGE=muneer7589/download-tusimple
TAG=1.0

# Build image
echo "Building image: "$IMAGE
docker build . -f Dockerfile \
--tag "${IMAGE}:${TAG}"
echo "Finished building image: "$IMAGE

# Push image
echo "Pushing image: "$IMAGE
docker push "${IMAGE}:${TAG}"
echo "Finished pushing image: "$IMAGE
```

d. Run the script to build the docker image and push it to the docker repository using these commands:

```
chmod +x build_image.sh
./build_image.sh
```

2. Submit the RUN: AI job to download, extract, pre-process, and store the TuSimple lane detection dataset in a `pvc`, which is dynamically created by NetApp Trident:

a. Use the following commands to submit the RUN: AI job:

```
runai submit
--name download-tusimple-data
--pvc azurenetaffiles:100Gi:/mnt
--image muneer7589/download-tusimple:1.0
```

- b. Enter the information from the table below to submit the RUN:AI job:

Field	Value or description
-name	Name of the job
-pvc	PVC of the format [StorageClassName]:Size:ContainerMountPath  In the above job submission, you are creating an PVC based on-demand using Trident with storage class azurenetaffiles. Persistent volume capacity here is 100Gi and it's mounted at path /mnt.
-image	Docker image to use when creating the container for this job

The output should look like the following example:

```
The job 'download-tusimple-data' has been submitted successfully
You can run `runai describe job download-tusimple-data -p lane-detection` to check the job status
```

- c. List the submitted RUN:AI jobs.

```
runai list jobs
```

Showing jobs for project lane-detection							
NAME	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER
download-tusimple-data	ContainerCreating	1m	aks-agentpool-34613062-vmss00000a	muneer7589/download-tusimple:1.0	Train	lane-detection	vernonmartina 0 (0)
1 (0)							

- d. Check the submitted job logs.

```
runai logs download-tusimple-data -t 10
```

751150K	.	.	.	.	.	.	.	.	.	.	6%	16.2M	20m37s
751200K	.	.	.	.	.	.	.	.	.	.	6%	11.1M	20m37s
751250K	.	.	.	.	.	.	.	.	.	.	6%	12.5M	20m36s
751300K	.	.	.	.	.	.	.	.	.	.	6%	11.3M	20m36s
751350K	.	.	.	.	.	.	.	.	.	.	6%	15.2M	20m36s
751400K	.	.	.	.	.	.	.	.	.	.	6%	10.5M	20m36s
751450K	.	.	.	.	.	.	.	.	.	.	6%	15.2M	20m36s
751500K	.	.	.	.	.	.	.	.	.	.	6%	14.1M	20m36s
751550K	.	.	.	.	.	.	.	.	.	.	6%	24.3M	20m36s
751600K	.	.	.	.	.	.	.	.	.	.	6%	26.3M	20m36s

- e. List the `pvc` created. Use this `pvc` command for training in the next step.

```
kubectl get pvc | grep download-tusimple-data
```

The output should look like the following example:

```
pvc-download-tusimple-data-0    Bound    pvc-bb03b74d-2c17-40c4-a445-79f3de8d16d5    100Gi    RWO    azurenetaappfiles    4m47s
```

- f. Check the job in RUN: AI UI (or [app.run.ai](#)).

The screenshot shows the RUN: AI UI interface with the title 'run: ai | Jobs'. At the top right, it displays 'Cluster: runalaks' and 'Muneer.Ahmad@netapp.com'. A 'NetApp' logo is also present. Below the header is a search bar with 'Filter and Search' placeholder text and a magnifying glass icon. The main area is a table with the following columns: Job Name, Status, User, Project, Total Run Time, Creation Time, Type, GPU Utilization, Used CPU, and a small circular icon. The table lists several jobs:

Job Name	Status	User	Project	Total Run Time	Creation Time	Type	GPU Utilization	Used CPU	Icon
download-tusimple-data	Running	verronma...	lane-detection	00:07:11	03/03/21, 2:51PM	Train	-	0.00	C
build1	Deleted	root	lane-detection	00:01:56	03/01/21, 10:18...	Interactive	-	-	-
download-tusimple-data	Deleted	root	lane-detection	-	03/01/21, 9:58AM	Train	-	-	-
download-tusimple-data	Deleted	root	lane-detection	-	03/01/21, 10:03...	Train	-	-	-
download-tusimple-data	Deleted	root	lane-detection	00:02:55	03/01/21, 10:24...	Train	-	-	-
download-tusimple-data	Deleted	root	lane-detection	-	03/01/21, 10:30...	Train	-	-	-
download-tusimple-data	Deleted	root	lane-detection	00:13:17	03/01/21, 11:41...	Train	-	-	-
download-tusimple-data-1	Deleted	verronma...	lane-detection	-	02/26/21, 5:30PM	Train	-	-	-

### Perform distributed lane detection training using Horovod

Performing distributed lane detection training using Horovod is an optional process. However, here are the steps involved:

1. Build and push the docker image, or skip this step if you want to use the existing docker image (for example, [muneer7589/dist-lane-detection:3.1](#)):

- a. Switch to home directory.

```
cd ~
```

- b. Go to the project directory [lane-detection-SCNN-horovod](#).

```
cd ./lane-detection-SCNN-horovod
```

- c. Modify the [build\\_image.sh](#) shell script and change docker repository to yours (for example, replace [muneer7589](#) with your docker repository name). You could also change the docker image name and TAG ([dist-lane-detection](#) and [3.1](#), for example).

```

#!/bin/bash
#
# A simple script to build the distributed Docker image.
#
# $ build_image.sh
set -ex

IMAGE=muneer7589/dist-lane-detection
TAG=3.0

# Build image
echo "Building image: "$IMAGE
docker build . -f Dockerfile \
--tag "${IMAGE}:${TAG}"
echo "Finished building image: "$IMAGE

# Push image
echo "Pushing image: "$IMAGE
docker push "${IMAGE}:${TAG}"
echo "Finished pushing image: "$IMAGE

```

- d. Run the script to build the docker image and push to the docker repository.

```

chmod +x build_image.sh
./build_image.sh

```

2. Submit the RUN: AI job for carrying out distributed training (MPI):

- Using submit of RUN: AI for automatically creating PVC in the previous step (for downloading data) only allows you to have RWO access, which does not allow multiple pods or nodes to access the same PVC for distributed training. Update the access mode to ReadWriteMany and use the Kubernetes patch to do so.
- First, get the volume name of the PVC by running the following command:

```
kubectl get pvc | grep download-tusimple-data
```

```

root@ai-w-gpu-2:/mnt/ai_data/anf_runai/lane-detection-SCNN-horovod# kubectl get pvc | grep download-tusimple-data
pvc-download-tusimple-data-0 Bound pvc-bb03b74d-2c17-40c4-a445-79f3de8d16d5 100Gi RWX azurenetaffiles 2d4h

```

- Patch the volume and update access mode to ReadWriteMany (replace volume name with yours in the following command):

```

kubectl patch pv pvc-bb03b74d-2c17-40c4-a445-79f3de8d16d5 -p
'{"spec":{"accessModes":["ReadWriteMany"]}}'

```

- d. Submit the RUN: AI MPI job for executing the distributed training` job using information from the table below:

```
runai submit-mpi
--name dist-lane-detection-training
--large-shm
--processes=3
--gpu 1
--pvc pvc-download-tusimple-data-0:/mnt
--image muneer7589/dist-lane-detection:3.1
-e USE_WORKERS="true"
-e NUM_WORKERS=4
-e BATCH_SIZE=33
-e USE_VAL="false"
-e VAL_BATCH_SIZE=99
-e ENABLE_SNAPSHOT="true"
-e PVC_NAME="pvc-download-tusimple-data-0"
```

Field	Value or description
name	Name of the distributed training job
large shm	Mount a large /dev/shm device  It is a shared file system mounted on RAM and provides large enough shared memory for multiple CPU workers to process and load batches into CPU RAM.
processes	Number of distributed training processes
gpu	Number of GPUs/processes to allocate for the job  In this job, there are three GPU worker processes (--processes=3), each allocated with a single GPU (--gpu 1)
pvc	Use existing persistent volume (pvc-download-tusimple-data-0) created by previous job (download-tusimple-data) and it is mounted at path /mnt
image	Docker image to use when creating the container for this job
Define environment variables to be set in the container	
USE_WORKERS	Setting the argument to true turns on multi-process data loading
NUM_WORKERS	Number of data loader worker processes
BATCH_SIZE	Training batch size

Field	Value or description
USE_VAL	Setting the argument to true allows validation
VAL_BATCH_SIZE	Validation batch size
ENABLE_SNAPSHOT	Setting the argument to true enables taking data and trained model snapshots for ML versioning purposes
PVC_NAME	Name of the pvc to take a snapshot of. In the above job submission, you are taking a snapshot of pvc-download-tusimple-data-0, consisting of dataset and trained models

The output should look like the following example:

```
The job 'dist-lane-detection-training' has been submitted successfully
You can run `runai describe job dist-lane-detection-training -p lane-detection` to check the job status.
```

- e. List the submitted job.

```
runai list jobs
```

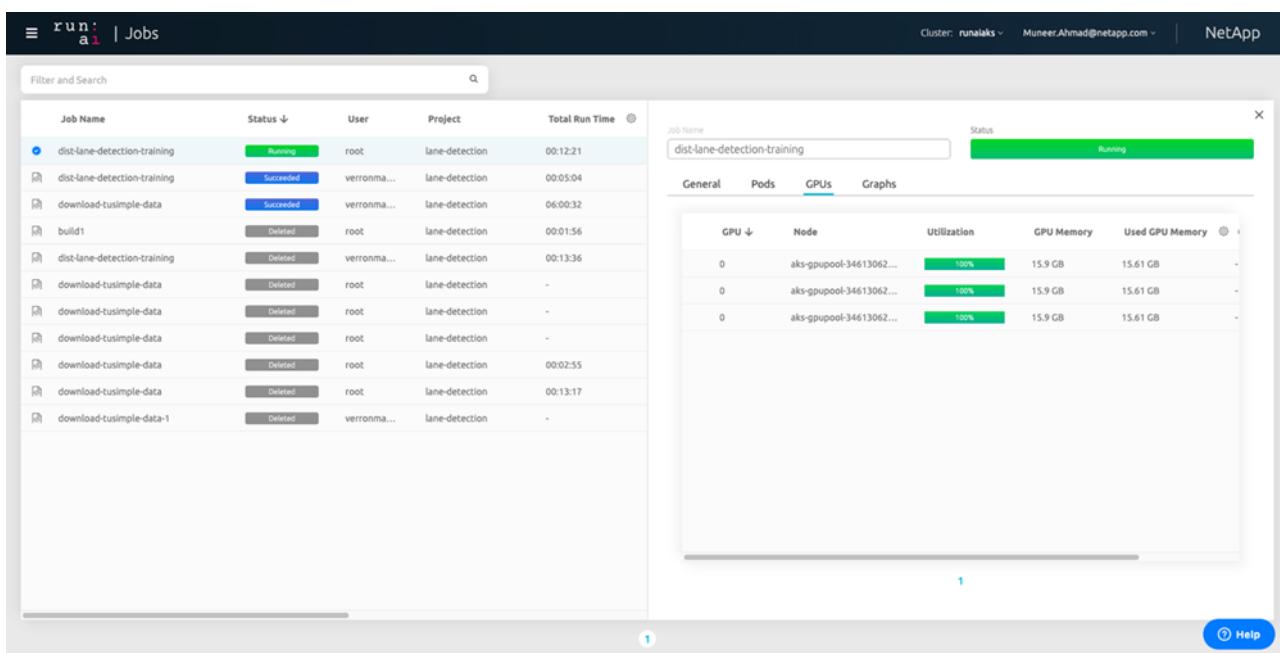
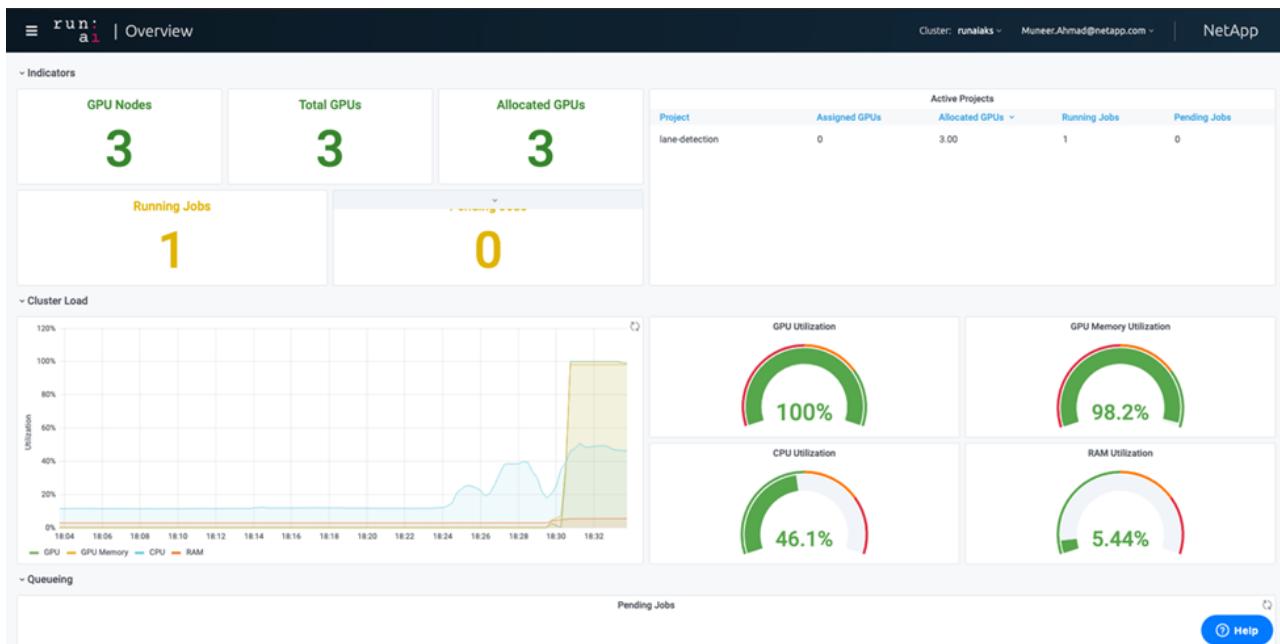
NAME	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUs Allocated (Requested)	PODs
download-tusimple-data	Succeeded	1d		muineer7589/download-tusimple:1.0	Train	lane-detection	verronmartina	- (0)	0 (0)
dist-lane-detection-training	Init:0/1	2m	<multiple>	muineer7589/dist-lane-detection:3.1	Train	lane-detection	root	3 (3)	4 (0)

- f. Submitted job logs:

```
runai logs dist-lane-detection-training
```

```
root@ai-w-gpu-2:~/runai# runai logs dist-lane-detection-training
Running with 3 workers
2021-03-04 17:29:23.158449: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully opened dynamic library libcudart.so.10.1
+ POD_NAME=dist-lane-detection-training-worker-0
+ [ d = - ]
+ shift
+ /opt/kube/kubectl cp /opt/kube/hosts dist-lane-detection-training-worker-0:/etc/hosts_of_nodes
+ POD_NAME=dist-lane-detection-training-worker-2
+ [ d = - ]
+ shift
+ /opt/kube/kubectl cp /opt/kube/hosts dist-lane-detection-training-worker-2:/etc/hosts_of_nodes
+ POD_NAME=dist-lane-detection-training-worker-1
```

- g. Check training job in RUN: AI GUI (or app.runai.ai): RUN: AI Dashboard, as seen in the figures below. The first figure details three GPUs allocated for the distributed training job spread across three nodes on AKS, and the second RUN:AI jobs:



- h. After the training is finished, check the NetApp Snapshot copy that was created and linked with RUN: AI job.

```
runai logs dist-lane-detection-training --tail 1
```

```
[1,0]<stdout>:Snapshot snap-pvc-download-tusimple-data-0-dist-lane-detection-training-launcher-2021-03-05-16-23-42 created in namespace runai-lane-detection
```

```
kubectl get volumesnapshots | grep download-tusimple-data-0
```

## Restore data from the NetApp Snapshot copy

To restore data from the NetApp Snapshot copy, complete the following steps:

1. Switch to home directory.

```
cd ~
```

2. Go to the project directory `lane-detection-SCNN-horovod`.

```
cd ./lane-detection-SCNN-horovod
```

3. Modify `restore-snapshot-pvc.yaml` and update `dataSource name` field to the Snapshot copy from which you want to restore data. You could also change PVC name where the data will be restored to, in this example its `restored-tusimple`.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: restored-tusimple
spec:
  storageClassName: azurenetappfiles
  dataSource:
    name: snap-pvc-download-tusimple-data-0-dist-lane-detection-training-launcher-2021-03-05-16-23-42
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
```

4. Create a new PVC by using `restore-snapshot-pvc.yaml`.

```
kubectl create -f restore-snapshot-pvc.yaml
```

The output should look like the following example:

```
persistentvolumeclaim/restored-tusimple created
```

5. If you want to use the just restored data for training, job submission remains the same as before; only replace the `PVC_NAME` with the restored `PVC_NAME` when submitting the training job, as seen in the following commands:

```
runai submit-mpi
--name dist-lane-detection-training
--large-shm
--processes=3
--gpu 1
--pvc restored-tusimple:/mnt
--image muneer7589/dist-lane-detection:3.1
-e USE_WORKERS="true"
-e NUM_WORKERS=4
-e BATCH_SIZE=33
-e USE_VAL="false"
-e VAL_BATCH_SIZE=99
-e ENABLE_SNAPSHOT="true"
-e PVC_NAME="restored-tusimple"
```

## Performance evaluation

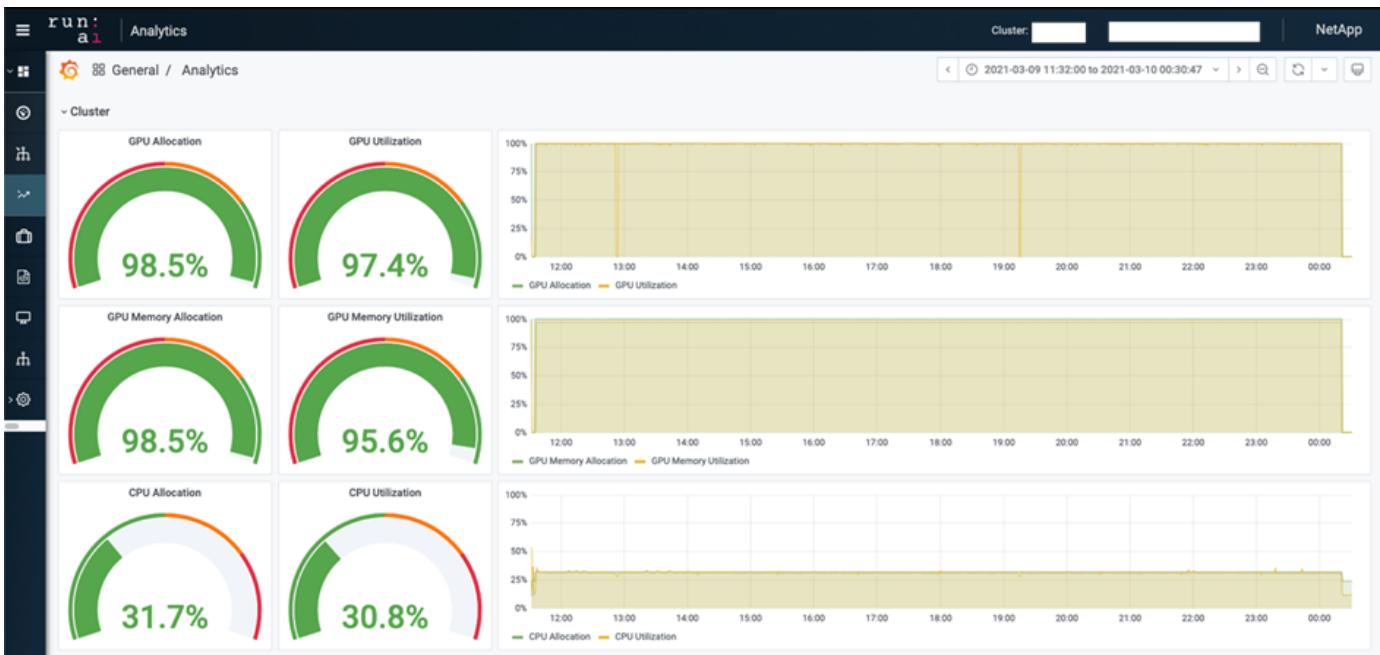
To show the linear scalability of the solution, performance tests have been done for two scenarios: one GPU and three GPUs. GPU allocation, GPU and memory utilization, different single- and three- node metrics have been captured during the training on the TuSimple lane detection dataset. Data is increased five- fold just for the sake of analyzing resource utilization during the training processes.

The solution enables customers to start with a small dataset and a few GPUs. When the amount of data and the demand of GPUs increase, customers can dynamically scale out the terabytes in the Standard Tier and quickly scale up to the Premium Tier to get four times the throughput per terabyte without moving any data. This process is further explained in the section, [Azure NetApp Files service levels](#).

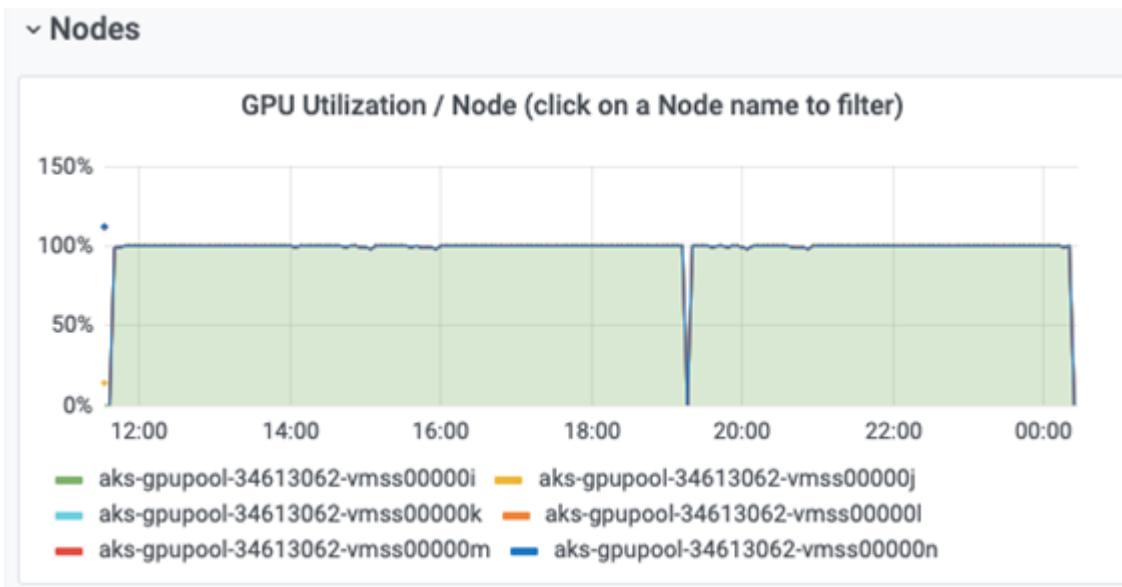
Processing time on one GPU was 12 hours and 45 minutes. Processing time on three GPUs across three nodes was approximately 4 hours and 30 minutes.

The figures shown throughout the remainder of this document illustrate examples of performance and scalability based on individual business needs.

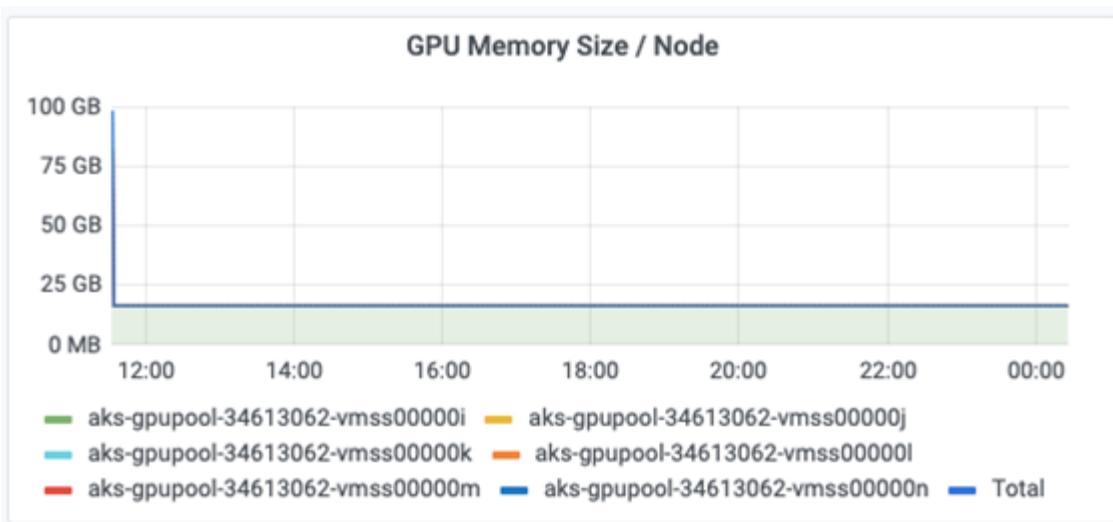
The figure below illustrates 1 GPU allocation and memory utilization.



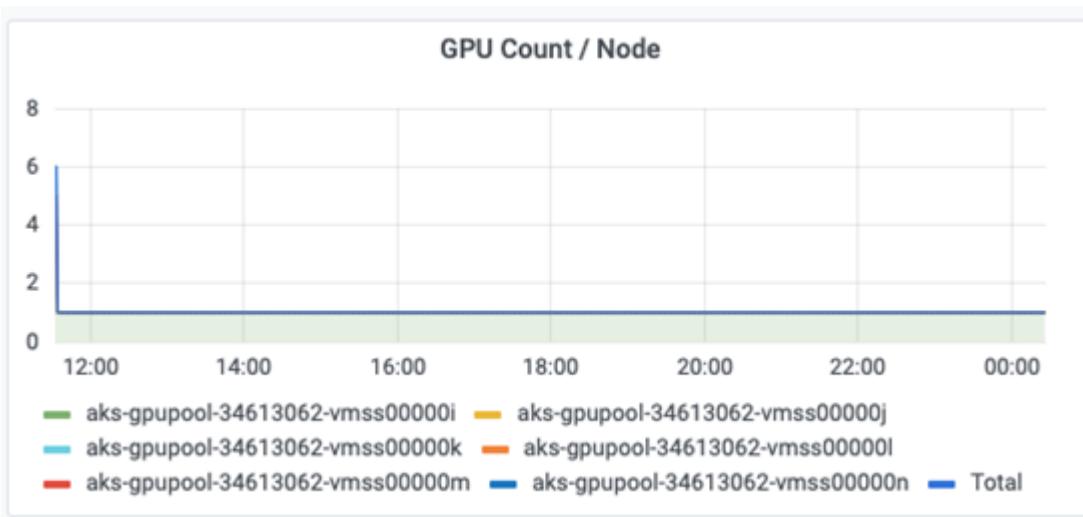
The figure below illustrates single node GPU utilization.



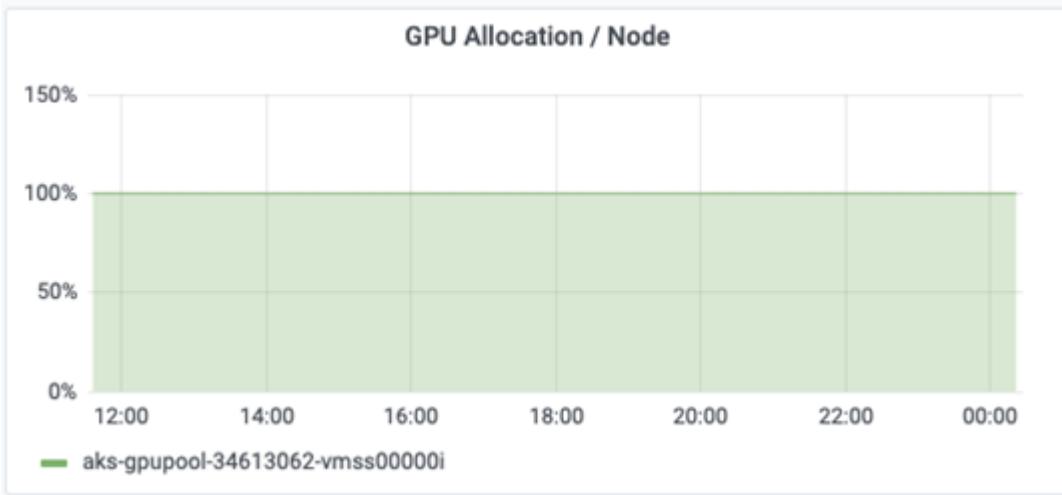
The figure below illustrates single node memory size (16GB).



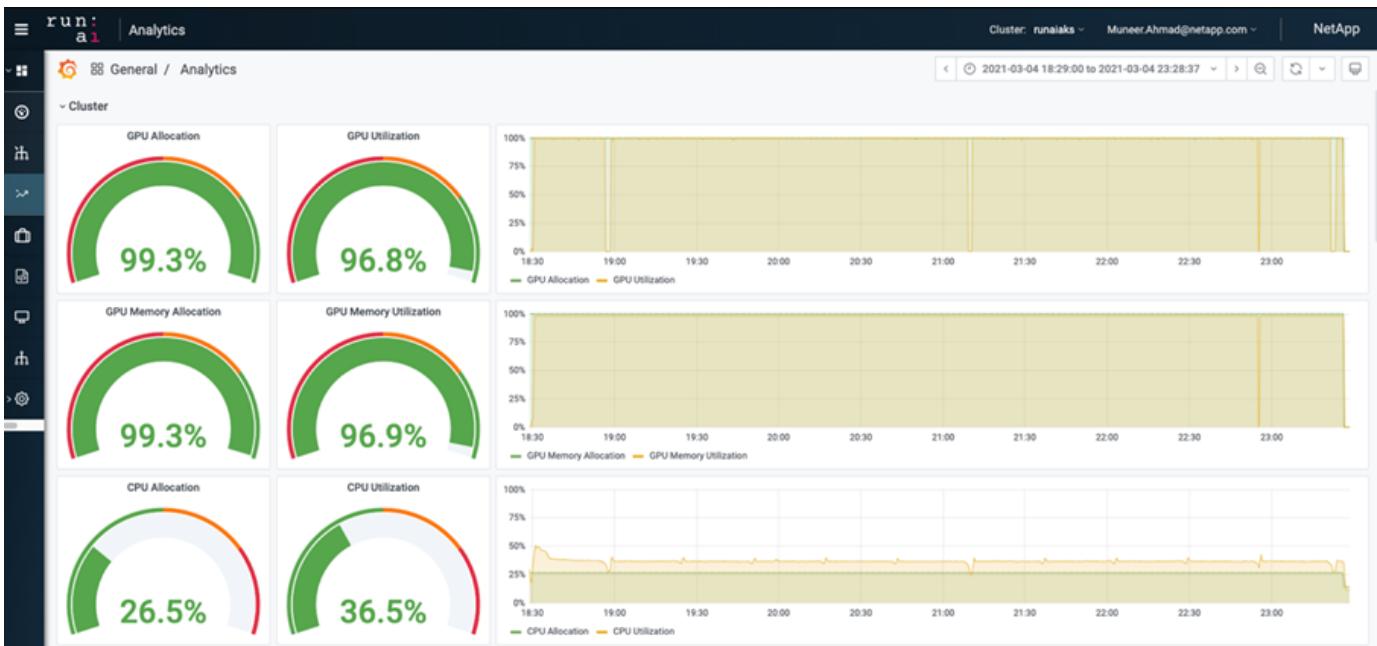
The figure below illustrates single node GPU count (1).



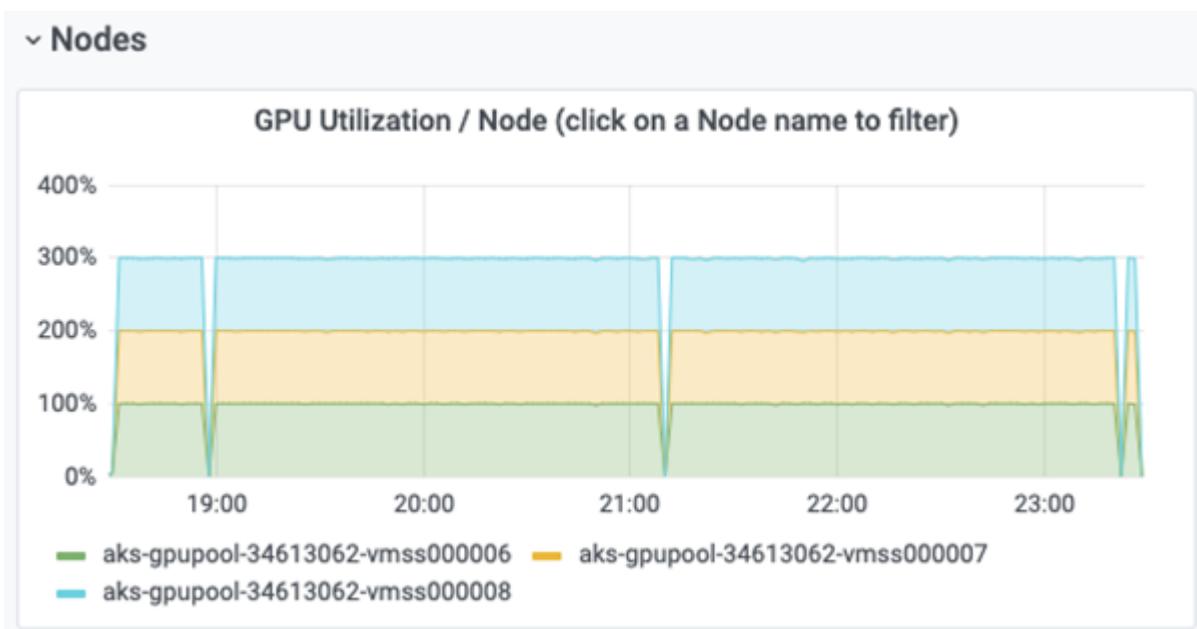
The figure below illustrates single node GPU allocation (%).



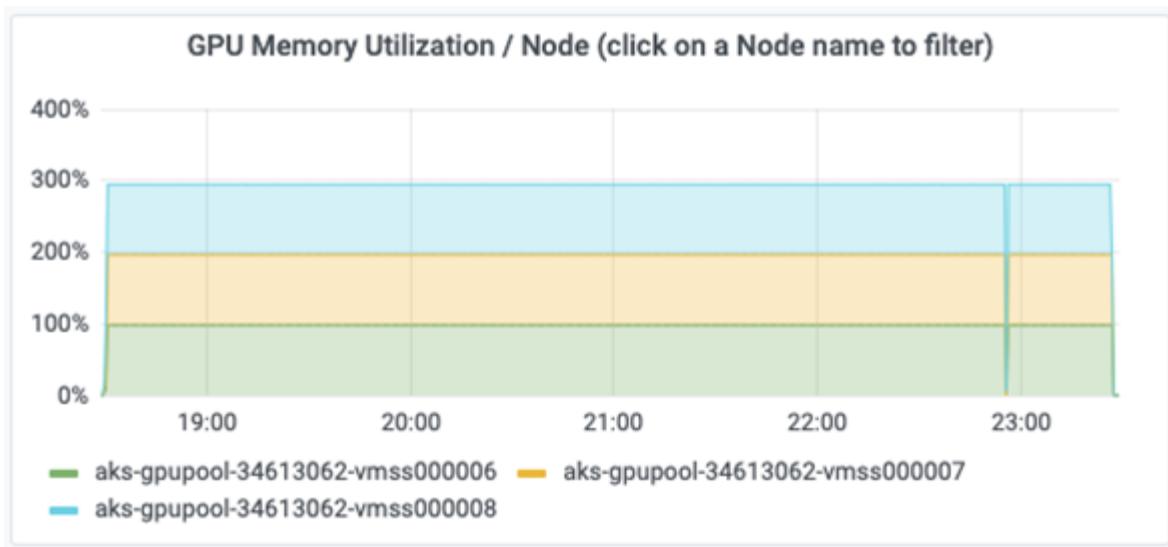
The figure below illustrates three GPUs across three nodes – GPUs allocation and memory.



The figure below illustrates three GPUs across three nodes utilization (%).



The figure below illustrates three GPUs across three nodes memory utilization (%).



## Azure NetApp Files service levels

You can change the service level of an existing volume by moving the volume to another capacity pool that uses the [service level](#) you want for the volume. This existing service-level change for the volume does not require that you migrate data. It also does not affect access to the volume.

### Dynamically change the service level of a volume

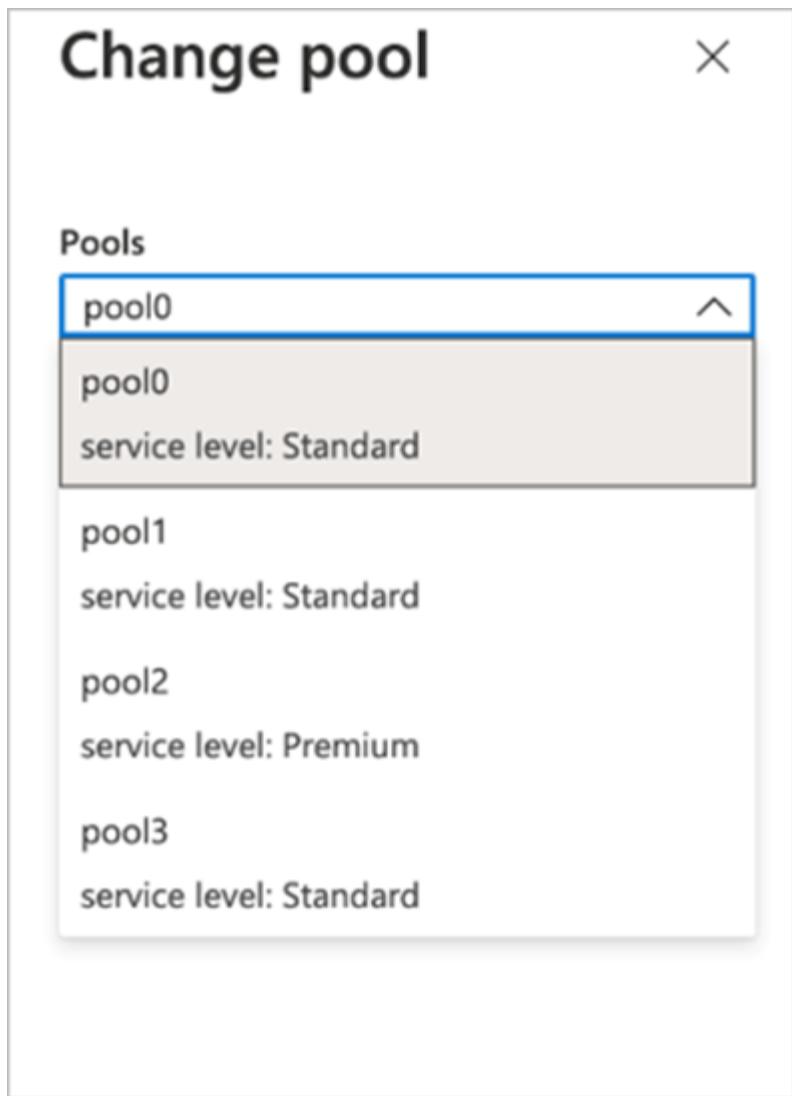
To change the service level of a volume, use the following steps:

1. On the Volumes page, right-click the volume whose service level you want to change. Select Change Pool.

NFSv3	10.28.254.4:/norootfor-	Standard	pool0	...
NFSv4.1	NAS-735a.docs.lab:/fo	Premium		...
NFSv4.1	NAS-735a.docs.lab:/kr	Premium		...
NFSv3	10.28.254.4:/moveme0	Premium		...
NFSv3	10.28.254.4:/placeholder	Premium		...

Resize
  
Edit
  
Change pool
  
Delete

2. In the Change Pool window, select the capacity pool you want to move the volume to. Then, click OK.



## Automate service level change

Dynamic Service Level change is currently still in Public Preview, but it is not enabled by default. To enable this feature on the Azure subscription, follow these steps provided in the document "[Dynamically change the service level of a volume](#)."

- You can also use the following commands for Azure CLI. For more information about changing the pool size of Azure NetApp Files, visit [az netappfiles volume: Manage Azure NetApp Files \(ANF\) volume resources](#).

```
az netappfiles volume pool-change -g mygroup  
--account-name myaccname  
-pool-name mypoolname  
--name myvolname  
--new-pool-resource-id mynewresourceid
```

- The `set- aznetappfilesvolumepool` cmdlet shown here can change the pool of an Azure NetApp Files volume. More information about changing volume pool size and Azure PowerShell can be found by visiting [Change pool for an Azure NetApp Files volume](#).

```
Set-AzNetAppFilesVolumePool  
-ResourceGroupName "MyRG"  
-AccountName "MyAnfAccount"  
-PoolName "MyAnfPool"  
-Name "MyAnfVolume"  
-NewPoolResourceId 7d6e4069-6c78-6c61-7bf6-c60968e45fbf
```

## Conclusion

NetApp and RUN: AI have partnered in the creation of this technical report to demonstrate the unique capabilities of the Azure NetApp Files together with the RUN: AI platform for simplifying orchestration of AI workloads. This technical report provides a reference architecture for streamlining the process of both data pipelines and workload orchestration for distributed lane detection training.

In conclusion, with regard to distributed training at scale (especially in a public cloud environment), the resource orchestration and storage component is a critical part of the solution. Making sure that data managing never hinders multiple GPU processing, therefore results in the optimal utilization of GPU cycles. Thus, making the system as cost effective as possible for large- scale distributed training purposes.

Data fabric delivered by NetApp overcomes the challenge by enabling data scientists and data engineers to connect together on-premises and in the cloud to have synchronous data, without performing any manual intervention. In other words, data fabric smooths the process of managing AI workflow spread across multiple locations. It also facilitates on demand-based data availability by bringing data close to compute and performing analysis, training, and validation wherever and whenever needed. This capability not only enables data integration but also protection and security of the entire data pipeline.

## Additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- Dataset: TuSimple

[https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane\\_detection](https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection)

- Deep Learning Network Architecture: Spatial Convolutional Neural Network

<https://arxiv.org/abs/1712.06080>

- Distributed deep learning training framework: Horovod

<https://horovod.ai/>

- RUN: AI container orchestration solution: RUN: AI product introduction

<https://docs.run.ai/home/components/>

- RUN: AI installation documentation

<https://docs.run.ai/Administrator/Cluster-Setup/cluster-install/#step-3-install-runai>

<https://docs.run.ai/Administrator/Researcher-Setup/cli-install/#runai-cli-installation>

- Submitting jobs in RUN: AI CLI

<https://docs.run.ai/Researcher/cli-reference/runai-submit/>

<https://docs.run.ai/Researcher/cli-reference/runai-submit-mpi/>

- Azure Cloud resources: Azure NetApp Files

<https://docs.microsoft.com/azure/azure-netapp-files/>

- Azure Kubernetes Service

<https://azure.microsoft.com/services/kubernetes-service/-features>

- Azure VM SKUs

<https://azure.microsoft.com/services/virtual-machines/>

- Azure VM with GPU SKUs

<https://docs.microsoft.com/azure/virtual-machines/sizes-gpu>

- NetApp Trident

<https://github.com/NetApp/trident/releases>

- Data Fabric powered by NetApp

<https://www.netapp.com/data-fabric/what-is-data-fabric/>

- NetApp Product Documentation

<https://www.netapp.com/support-and-training/documentation/>

## TR-4841: Hybrid Cloud AI Operating System with Data Caching

Rick Huang, David Arnette, NetApp

Yochay Ettun, cnvrg.io

The explosive growth of data and the exponential growth of ML and AI have converged to create a zettabyte economy with unique development and implementation challenges.

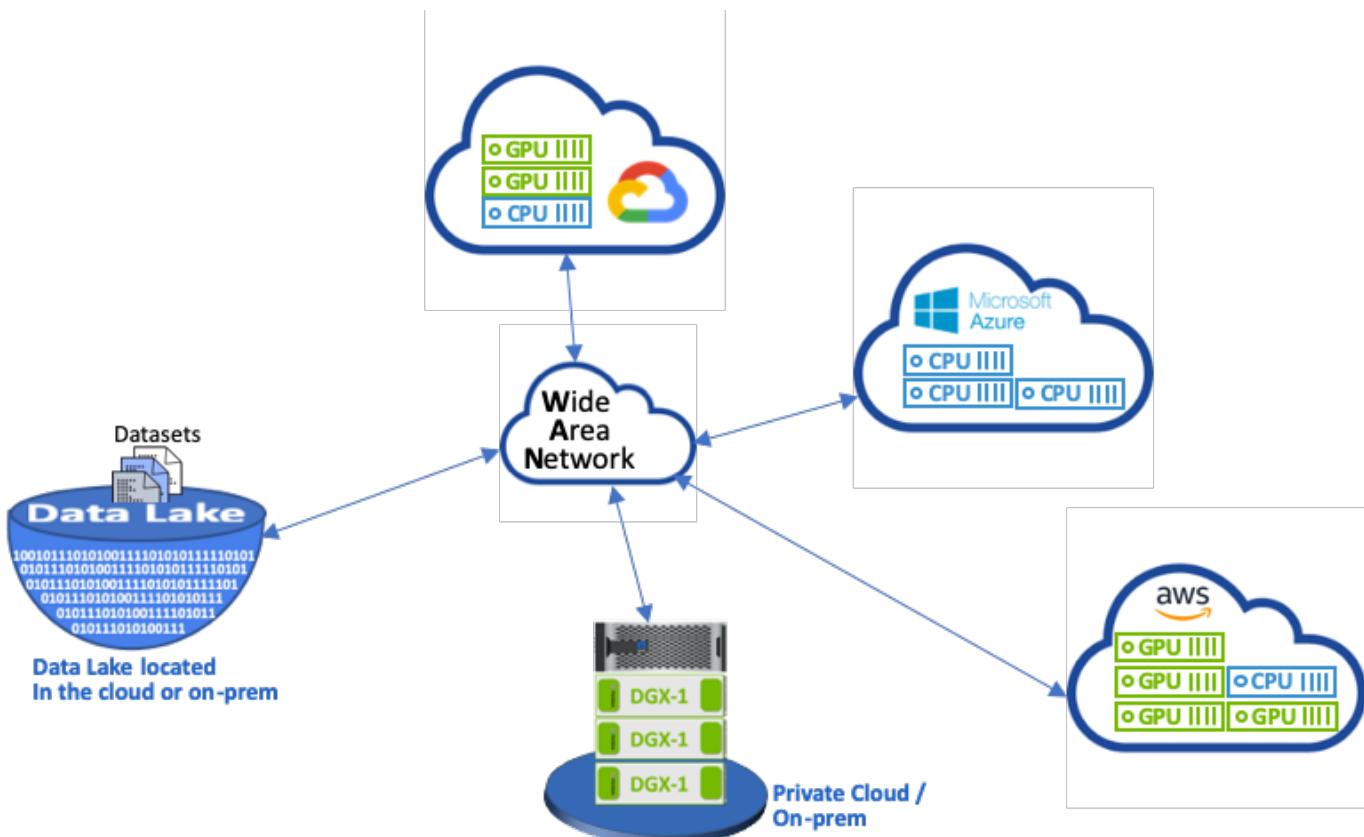
Although it is a widely known that ML models are data-hungry and require high-performance data storage proximal to compute resources, in practice, it is not so straight forward to implement this model, especially with hybrid cloud and elastic compute instances. Massive quantities of data are usually stored in low-cost data lakes, where high-performance AI compute resources such as GPUs cannot efficiently access it. This problem is aggravated in a hybrid-cloud infrastructure where some workloads operate in the cloud and some are located on-premises or in a different HPC environment entirely.

In this document, we present a novel solution that allows IT professionals and data engineers to create a truly hybrid cloud AI platform with a topology-aware data hub that enables data scientists to instantly and automatically create a cache of their datasets in proximity to their compute resources, wherever they are located. As a result, not only can high-performance model training be accomplished, but additional benefits are created, including the collaboration of multiple AI practitioners, who have immediate access to dataset caches, versions, and lineages within a dataset version hub.

[Next: Use Case Overview and Problem Statement](#)

## Use Case Overview and Problem Statement

Datasets and dataset versions are typically located in a data lake, such as NetApp StorageGrid object-based storage, which offers reduced cost and other operational advantages. Data scientists pull these datasets and engineer them in multiple steps to prepare them for training with a specific model, often creating multiple versions along the way. As the next step, the data scientist must pick optimized compute resources (GPUs, high-end CPU instances, an on-premises cluster, and so on) to run the model. The following figure depicts the lack of dataset proximity in an ML compute environment.



However, multiple training experiments must run in parallel in different compute environments, each of which require a download of the dataset from the data lake, which is an expensive and time-consuming process. Proximity of the dataset to the compute environment (especially for a hybrid cloud) is not guaranteed. In addition, other team members that run their own experiments with the same dataset must go through the same arduous process. Beyond the obvious slow data access, challenges include difficulties tracking dataset versions, dataset sharing, collaboration, and reproducibility.

### Customer Requirements

Customer requirements can vary in order to achieve high- performance ML runs while efficiently using resources; for example, customers might require the following:

- Fast access to datasets from each compute instance executing the training model without incurring expensive downloads and data access complexities
- The use any compute instance (GPU or CPU) in the cloud or on-premises without concern for the location

of the datasets

- Increased efficiency and productivity by running multiple training experiments in parallel with different compute resources on the same dataset without unnecessary delays and data latency
- Minimized compute instance costs
- Improved reproducibility with tools to keep records of the datasets, their lineage, versions, and other metadata details
- Enhanced sharing and collaboration so that any authorized member of the team can access the datasets and run experiments

To implement dataset caching with NetApp ONTAP data management software, customers must perform the following tasks:

- Configure and set the NFS storage that is closest to the compute resources.
- Determine which dataset and version to cache.
- Monitor the total memory committed to cached datasets and how much NFS storage is available for additional cache commits (for example, cache management).
- Age out of datasets in the cache if they have not been used in certain time. The default is one day; other configuration options are available.

[Next: Solution Overview](#)

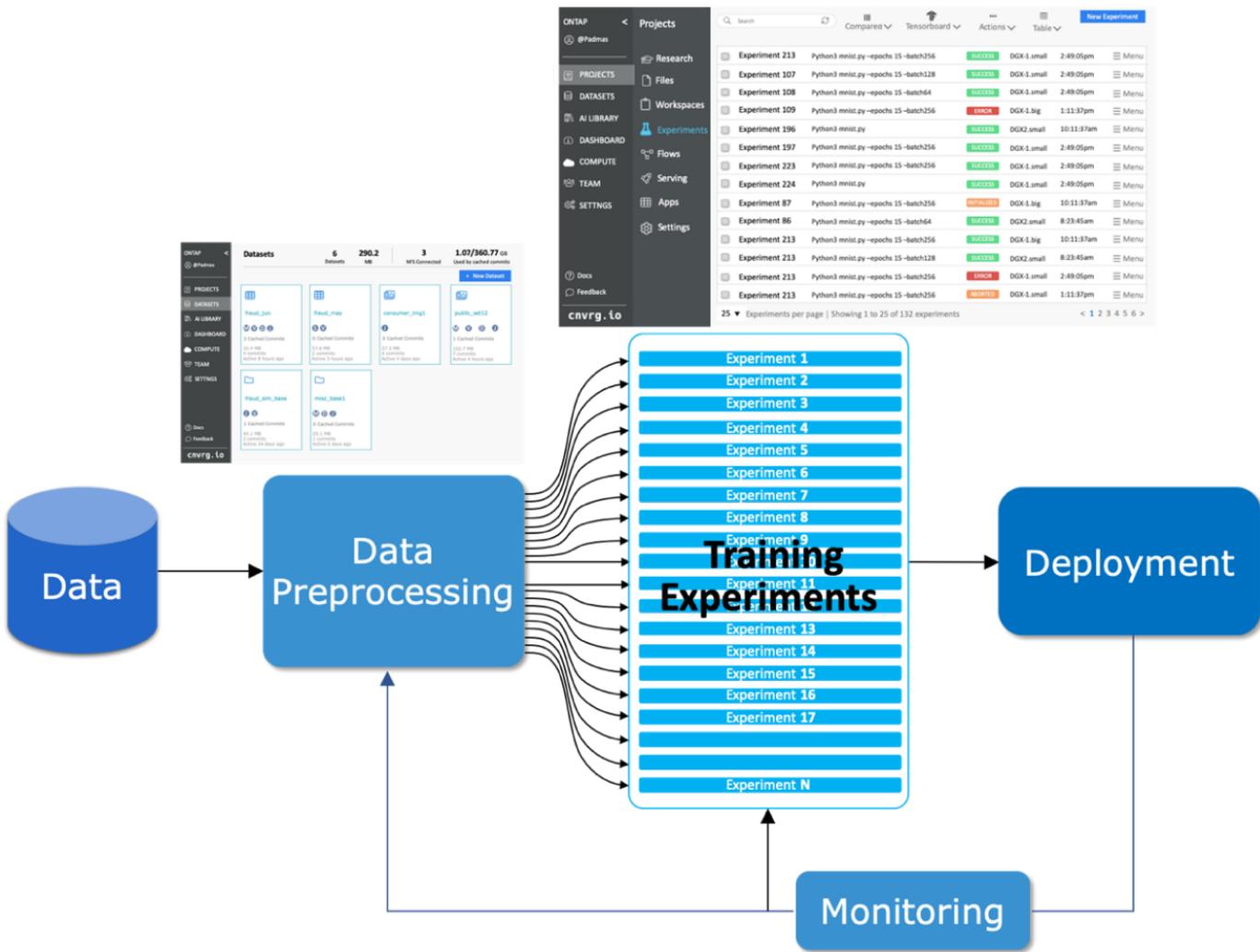
## Solution Overview

This section reviews a conventional data science pipeline and its drawbacks. It also presents the architecture of the proposed dataset caching solution.

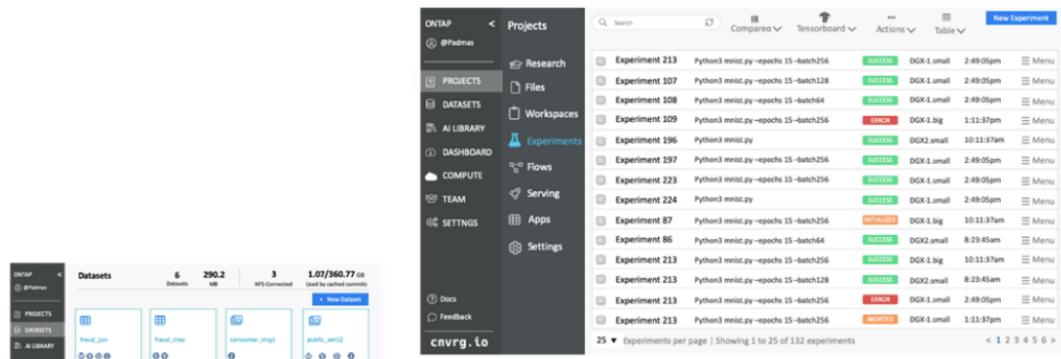
### Conventional Data Science Pipeline and Drawbacks

A typical sequence of ML model development and deployment involves iterative steps that include the following:

- Ingesting data
- Data preprocessing (creating multiple versions of the datasets)
- Running multiple experiments involving hyperparameter optimization, different models, and so on
- Deployment
- Monitoringcnvrg.io has developed a comprehensive platform to automate all tasks from research to deployment. A small sample of dashboard screenshots pertaining to the pipeline is shown in the following figure.



It is very common to have multiple datasets in play from public repositories and private data. In addition, each dataset is likely to have multiple versions resulting from dataset cleanup or feature engineering. A dashboard that provides a dataset hub and a version hub is needed to make sure collaboration and consistency tools are available to the team, as can be seen in the following figure.



Datasets		6 Datasets	290.2 MB	3 NFS Connected	1.07/360.77 GB Used by cached commits
	<a href="#">fraud_jun</a>		<a href="#">fraud_may</a>		<a href="#">consumer_img1</a>
M V D J		S V		J	M V D J
2 Cached Commits		0 Cached Commits		0 Cached Commits	1 Cached Commits
32.4 MB 4 commits Active 8 hours ago		57.6 MB 2 commits Active 3 hours ago		27.3 MB 4 commits Active 4 days ago	102.7 MB 7 commits Active 4 hours ago
	<a href="#">fraud_sim_base</a>		<a href="#">misc_base1</a>		
J V		M D J			
1 Cached Commits		0 Cached Commits			
45.1 MB 2 commits Active 24 days ago		25.1 MB 1 commits Active 2 days ago			

The next step in the pipeline is training, which requires multiple parallel instances of training models, each associated with a dataset and a certain compute instance. The binding of a dataset to a certain experiment with a certain compute instance is a challenge because it is possible that some experiments are performed by GPU instances from Amazon Web Services (AWS), while other experiments are performed by DGX-1 or DGX-2 instances on-premises. Other experiments might be executed in CPU servers in GCP, while the dataset location is not in reasonable proximity to the compute resources performing the training. A reasonable proximity would have full 10GbE or more low-latency connectivity from the dataset storage to the compute instance.

It is a common practice for data scientists to download the dataset to the compute instance performing the training and execute the experiment. However, there are several potential problems with this approach:

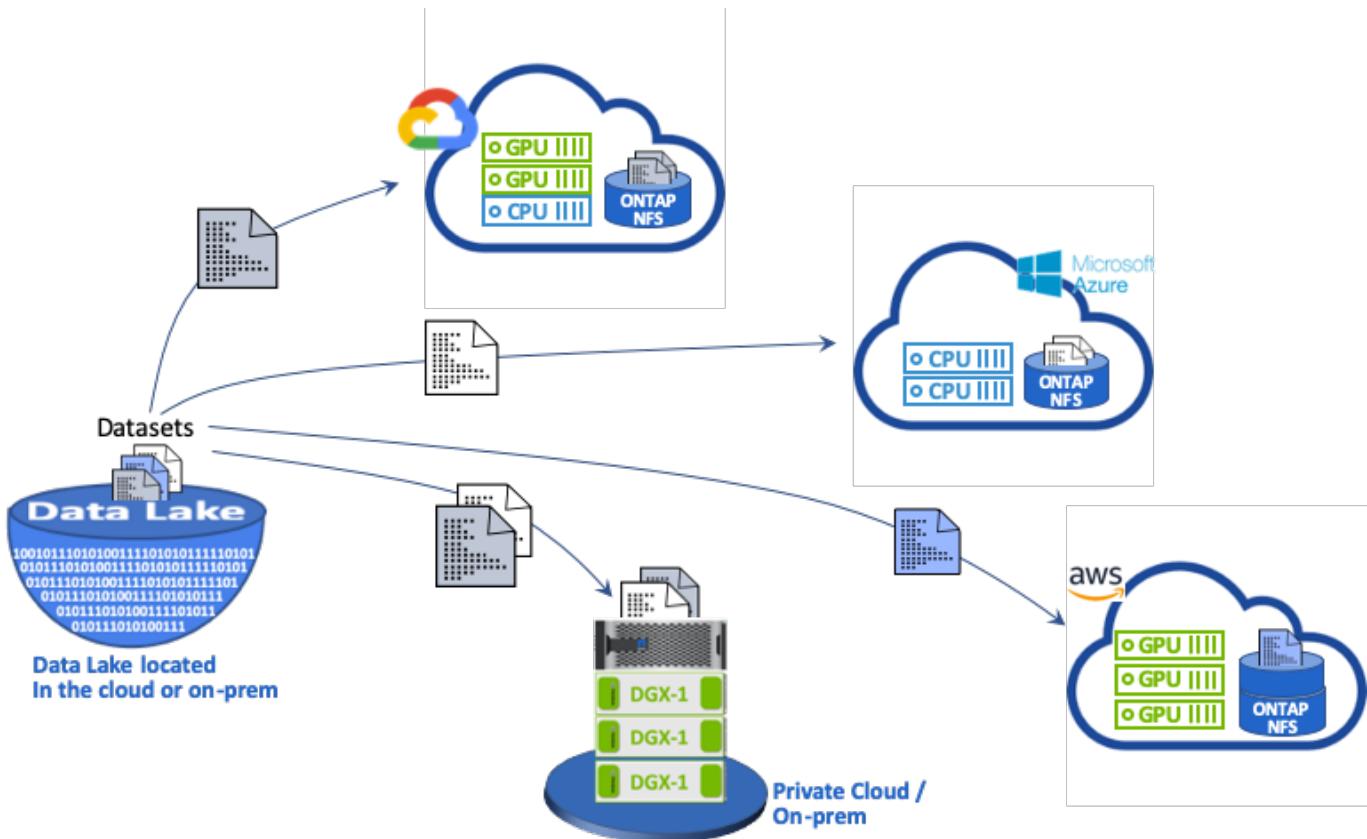
- When the data scientist downloads the dataset to a compute instance, there are no guarantees that the integrated compute storage is high performance (an example of a high-performance system would be the ONTAP AFF A800 NVMe solution).
- When the downloaded dataset resides in one compute node, storage can become a bottleneck when distributed models are executed over multiple nodes (unlike with NetApp ONTAP high-performance distributed storage).
- The next iteration of the training experiment might be performed in a different compute instance due to queue conflicts or priorities, again creating significant network distance from the dataset to the compute location.
- Other team members executing training experiments on the same compute cluster cannot share this dataset; each performs the (expensive) download of the dataset from an arbitrary location.
- If other datasets or versions of the same dataset are needed for the subsequent training jobs, the data scientists must again perform the (expensive) download of the dataset to the compute instance performing the training. NetApp and cnvrg.io have created a new dataset caching solution that eliminates these

hurdles. The solution creates accelerated execution of the ML pipeline by caching hot datasets on the ONTAP high-performance storage system. With ONTAP NFS, the datasets are cached once (and only once) in a data fabric powered by NetApp (such as AFF A800), which is collocated with the compute. As the NetApp ONTAP NFS high-speed storage can serve multiple ML compute nodes, the performance of the training models is optimized, bringing cost savings, productivity, and operational efficiency to the organization.

## Solution Architecture

This solution from NetApp and cnvrg.io provides dataset caching, as shown in the following figure. Dataset caching allows data scientists to pick a desired dataset or dataset version and move it to the ONTAP NFS cache, which lies in proximity to the ML compute cluster. The data scientist can now run multiple experiments without incurring delays or downloads. In addition, all collaborating engineers can use the same dataset with the attached compute cluster (with the freedom to pick any node) without additional downloads from the data lake. The data scientists are offered a dashboard that tracks and monitors all datasets and versions and provides a view of which datasets were cached.

The cnvrg.io platform auto-detects aged datasets that have not been used for a certain time and evicts them from the cache, which maintains free NFS cache space for more frequently used datasets. It is important to note that dataset caching with ONTAP works in the cloud and on-premises, thus providing maximum flexibility.



[Next: Concepts and Components](#)

## Concepts and Components

This section covers concepts and components associated with data caching in an ML workflow.

## Machine Learning

ML is rapidly becoming essential to many businesses and organizations around the world. Therefore, IT and DevOps teams are now facing the challenge of standardizing ML workloads and provisioning cloud, on-premises, and hybrid compute resources that support the dynamic and intensive workflows that ML jobs and pipelines require.

### Container-Based Machine Learning and Kubernetes

Containers are isolated user-space instances that run on top of a shared host operating system kernel. The adoption of containers is rapidly increasing. Containers offer many of the same application sandboxing benefits that virtual machines (VMs) offer. However, because the hypervisor and guest operating system layers that VMs rely on have been eliminated, containers are far more lightweight.

Containers also allow the efficient packaging of application dependencies, run times, and so on directly with an application. The most commonly used container packaging format is the Docker container. An application that has been containerized in the Docker container format can be executed on any machine that can run Docker containers. This is true even if the application's dependencies are not present on the machine, because all dependencies are packaged in the container itself. For more information, visit the [Docker website](#).

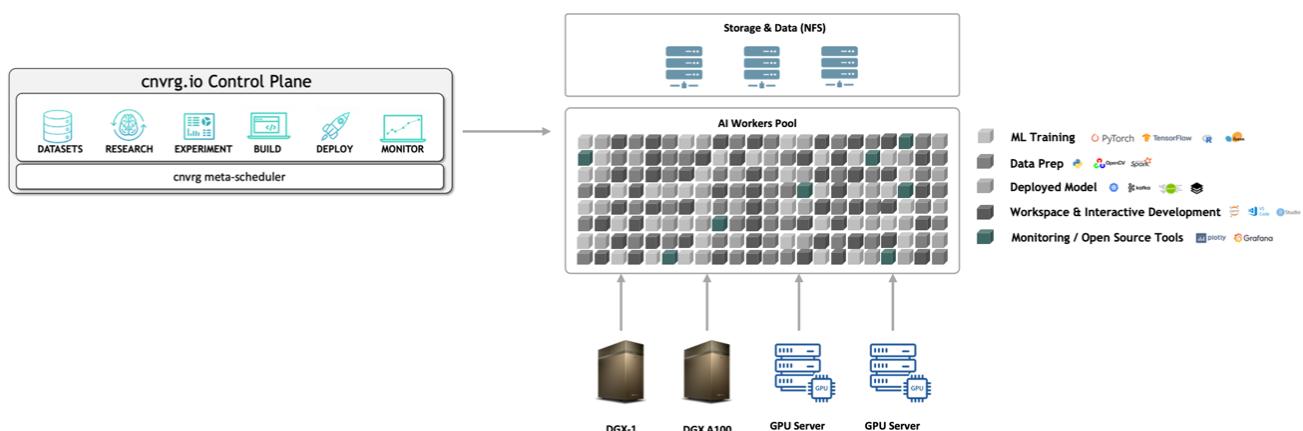
Kubernetes, the popular container orchestrator, allows data scientists to launch flexible, container-based jobs and pipelines. It also enables infrastructure teams to manage and monitor ML workloads in a single managed and cloud-native environment. For more information, visit the [Kubernetes website](#).

### cnvrg.io

cnvrg.io is an AI operating system that transforms the way enterprises manage, scale, and accelerate AI and data science development from research to production. The code-first platform is built by data scientists for data scientists and offers flexibility to run on-premises or in the cloud. With model management, MLOps, and continual ML solutions, cnvrg.io brings top-of-the-line technology to data science teams so they can spend less time on DevOps and focus on the real magic—algorithms. Since using cnvrg.io, teams across industries have gotten more models to production resulting in increased business value.

### cnvrg.io Meta-Scheduler

cnvrg.io has a unique architecture that allows IT and engineers to attach different compute resources to the same control plane and have cnvrg.io manage ML jobs across all resources. This means that IT can attach multiple on-premises Kubernetes clusters, VM servers, and cloud accounts and run ML workloads on all resources, as shown in the following figure.



## **cnvrg.io Data Caching**

cnvrg.io allows data scientists to define hot and cold dataset versions with its data-caching technology. By default, datasets are stored in a centralized object storage database. Then, data scientists can cache a specific data version on the selected compute resource to save time on download and therefore increase ML development and productivity. Datasets that are cached and are not in use for a few days are automatically cleared from the selected NFS. Caching and clearing the cache can be performed with a single click; no coding, IT, or DevOps work is required.

## **cnvrg.io Flows and ML Pipelines**

cnvrg.io Flows is a tool for building production ML pipelines. Each component in a flow is a script/code running on a selected compute with a base docker image. This design enables data scientists and engineers to build a single pipeline that can run both on-premises and in the cloud. cnvrg.io makes sure data, parameters, and artifacts are moving between the different components. In addition, each flow is monitored and tracked for 100% reproducible data science.

## **cnvrg.io CORE**

cnvrg.io CORE is a free platform for the data science community to help data scientists focus more on data science and less on DevOps. CORE's flexible infrastructure gives data scientists the control to use any language, AI framework, or compute environment whether on-premises or in the cloud so they can do what they do best, build algorithms. cnvrg.io CORE can be easily installed with a single command on any Kubernetes cluster.

## **NetApp ONTAP AI**

ONTAP AI is a data center reference architecture for ML and deep learning (DL) workloads that uses NetApp AFF storage systems and NVIDIA DGX systems with Tesla V100 GPUs. ONTAP AI is based on the industry-standard NFS file protocol over 100Gb Ethernet, providing customers with a high-performance ML/DL infrastructure that uses standard data center technologies to reduce implementation and administration overhead. Using standardized network and protocols enables ONTAP AI to integrate into hybrid cloud environments while maintaining operational consistency and simplicity. As a prevalidated infrastructure solution, ONTAP AI reduces deployment time and risk and reduces administration overhead significantly, allowing customers to realize faster time to value.

## **NVIDIA DeepOps**

DeepOps is an open source project from NVIDIA that, by using Ansible, automates the deployment of GPU server clusters according to best practices. DeepOps is modular and can be used for various deployment tasks. For this document and the validation exercise that it describes, DeepOps is used to deploy a Kubernetes cluster that consists of GPU server worker nodes. For more information, visit the [DeepOps website](#).

## **NetApp Trident**

Trident is an open source storage orchestrator developed and maintained by NetApp that greatly simplifies the creation, management, and consumption of persistent storage for Kubernetes workloads. Trident itself a Kubernetes-native application—it runs directly within a Kubernetes cluster. With Trident, Kubernetes users (developers, data scientists, Kubernetes administrators, and so on) can create, manage, and interact with persistent storage volumes in the standard Kubernetes format that they are already familiar with. At the same time, they can take advantage of NetApp advanced data management capabilities and a data fabric that is powered by NetApp technology. Trident abstracts away the complexities of persistent storage and makes it simple to consume. For more information, visit the [Trident website](#).

## **NetApp StorageGRID**

NetApp StorageGRID is a software-defined object storage platform designed to meet these needs by providing simple, cloud-like storage that users can access using the S3 protocol. StorageGRID is a scale-out system designed to support multiple nodes across internet-connected sites, regardless of distance. With the intelligent policy engine of StorageGRID, users can choose erasure-coding objects across sites for geo-resiliency or object replication between remote sites to minimize WAN access latency. StorageGrid provides an excellent private-cloud primary object storage data lake in this solution.

## **NetApp Cloud Volumes ONTAP**

NetApp Cloud Volumes ONTAP data management software delivers control, protection, and efficiency to user data with the flexibility of public cloud providers including AWS, Google Cloud Platform, and Microsoft Azure. Cloud Volumes ONTAP is cloud-native data management software built on the NetApp ONTAP storage software, providing users with a superior universal storage platform that addresses their cloud data needs. Having the same storage software in the cloud and on-premises provides users with the value of a data fabric without having to train IT staff in all-new methods to manage data.

For customers that are interested in hybrid cloud deployment models, Cloud Volumes ONTAP can provide the same capabilities and class-leading performance in most public clouds to provide a consistent and seamless user experience in any environment.

[Next: Hardware and Software Requirements](#)

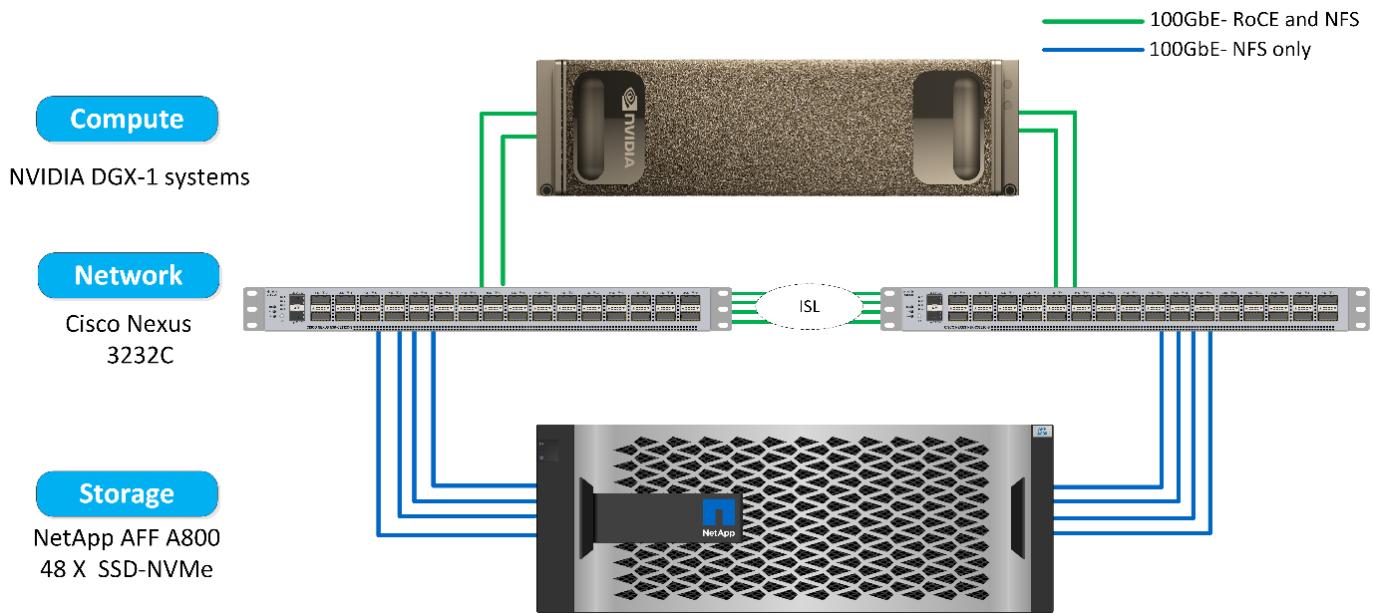
## **Hardware and Software Requirements**

This section covers the technology requirements for the ONTAP AI solution.

### **Hardware Requirements**

Although hardware requirements depend on specific customer workloads, ONTAP AI can be deployed at any scale for data engineering, model training, and production inferencing from a single GPU up to rack-scale configurations for large-scale ML/DL operations. For more information about ONTAP AI, see the [ONTAP AI website](#).

This solution was validated using a DGX-1 system for compute, a NetApp AFF A800 storage system, and Cisco Nexus 3232C for network connectivity. The AFF A800 used in this validation can support as many as 10 DGX-1 systems for most ML/DL workloads. The following figure shows the ONTAP AI topology used for model training in this validation.



To extend this solution to a public cloud, Cloud Volumes ONTAP can be deployed alongside cloud GPU compute resources and integrated into a hybrid cloud data fabric that enables customers to use whatever resources are appropriate for any given workload.

### Software Requirements

The following table shows the specific software versions used in this solution validation.

Component	Version
Ubuntu	18.04.4 LTS
NVIDIA DGX OS	4.4.0
NVIDIA DeepOps	20.02.1
Kubernetes	1.15
Helm	3.1.0
cnvrg.io	3.0.0
NetApp ONTAP	9.6P4

For this solution validation, Kubernetes was deployed as a single-node cluster on the DGX-1 system. For large-scale deployments, independent Kubernetes master nodes should be deployed to provide high availability of management services as well as reserve valuable DGX resources for ML and DL workloads.

[Next: Solution Deployment and Validation Details](#)

### Solution Deployment and Validation Details

The following sections discuss the details of solution deployment and validation.

[Next: ONTAP AI Deployment](#)

## ONTAP AI Deployment

Deployment of ONTAP AI requires the installation and configuration of networking, compute, and storage hardware. Specific instructions for deployment of the ONTAP AI infrastructure are beyond the scope of this document. For detailed deployment information, see [NVA-1121-DEPLOY: NetApp ONTAP AI, Powered by NVIDIA](#).

For this solution validation, a single volume was created and mounted to the DGX-1 system. That mount point was then mounted to the containers to make data accessible for training. For large-scale deployments, NetApp Trident automates the creation and mounting of volumes to eliminate administrative overhead and enable end-user management of resources.

[Next: Kubernetes Deployment](#)

## Kubernetes Deployment

To deploy and configure your Kubernetes cluster with NVIDIA DeepOps, perform the following tasks from a deployment jump host:

1. Download NVIDIA DeepOps by following the instructions on the [Getting Started page](#) on the NVIDIA DeepOps GitHub site.
2. Deploy Kubernetes in your cluster by following the instructions on the [Kubernetes Deployment Guide](#) on the NVIDIA DeepOps GitHub site.



For the DeepOps Kubernetes deployment to work, the same user must exist on all Kubernetes master and worker nodes.

If the deployment fails, change the value of `kubectl_localhost` to `false` in `deepops/config/group_vars/k8s-cluster.yml` and repeat step 2. The `Copy kubectl binary to ansible host` task, which executes only when the value of `kubectl_localhost` is `true`, relies on the fetch Ansible module, which has known memory usage issues. These memory usage issues can sometimes cause the task to fail. If the task fails because of a memory issue, then the remainder of the deployment operation does not complete successfully.

If the deployment completes successfully after you have changed the value of `kubectl_localhost` to `false`, then you must manually copy the `kubectl binary` from a Kubernetes master node to the deployment jump host. You can find the location of the `kubectl binary` on a specific master node by running the `which kubectl` command directly on that node.

[Next: Cnvrge.io Deployment](#)

## cnvrg.io Deployment

### Deploy cnvrg CORE Using Helm

Helm is the easiest way to quickly deploy cnvrg using any cluster, on-premises, Minikube, or on any cloud cluster (such as AKS, EKS, and GKE). This section describes how cnvrg was installed on an on-premises (DGX-1) instance with Kubernetes installed.

### Prerequisites

Before you can complete the installation, you must install and prepare the following dependencies on your

local machine:

- Kubectl
- Helm 3.x
- Kubernetes cluster 1.15+

## Deploy Using Helm

1. To download the most updated cnvrg helm charts, run the following command:

```
helm repo add cnvrg https://helm.cnvrg.io  
helm repo update
```

2. Before you deploy cnvrg, you need the external IP address of the cluster and the name of the node on which you will deploy cnvrg. To deploy cnvrg on an on-premises Kubernetes cluster, run the following command:

```
helm install cnvrg cnvrg/cnvrg --timeout 1500s --wait \ --set  
global.external_ip=<ip_of_cluster> \ --set global.node=<name_of_node>
```

3. Run the `helm install` command. All the services and systems automatically install on your cluster. The process can take up to 15 minutes.
4. The `helm install` command can take up to 10 minutes. When the deployment completes, go to the URL of your newly deployed cnvrg or add the new cluster as a resource inside your organization. The `helm` command informs you of the correct URL.

Thank you for installing cnvrg.io!  
Your installation of cnvrg.io is now available, and can be reached via:  
Talk to our team via email at

5. When the status of all the containers is running or complete, cnvrg has been successfully deployed. It should look similar to the following example output:

NAME	READY	STATUS	RESTARTS	AGE
cnvrg-app-69fbb9df98-6xrgf	1/1	Running	0	2m
cnvrg-sidekiq-b9d54d889-5x4fc	1/1	Running	0	2m
controller-65895b47d4-s96v6	1/1	Running	0	2m
init-app-vs-config-wv9c4	0/1	Completed	0	9m
init-gateway-vs-config-2zbpp	0/1	Completed	0	9m
init-minio-vs-config-cd2rg	0/1	Completed	0	9m
minio-0	1/1	Running	0	2m
postgres-0	1/1	Running	0	2m
redis-695c49c986-kcbt9	1/1	Running	0	2m
seeder-wh655	0/1	Completed	0	2m
speaker-5sghr	1/1	Running	0	2m

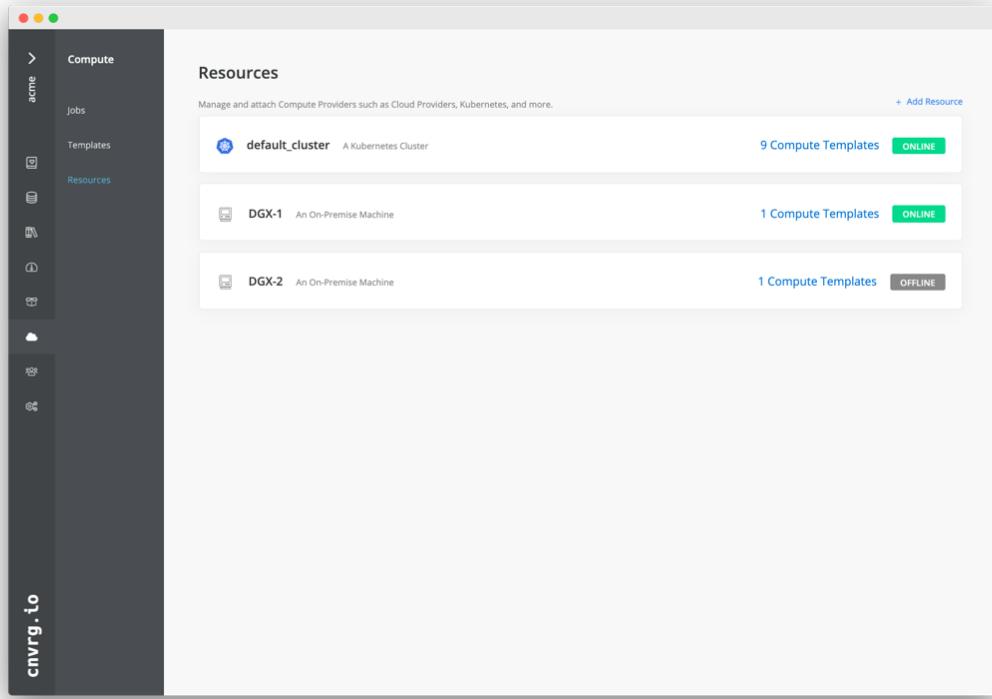
## Computer Vision Model Training with ResNet50 and the Chest X-ray Dataset

cnvrg.io AI OS was deployed on a Kubernetes setup on a NetApp ONTAP AI architecture powered by the NVIDIA DGX system. For validation, we used the NIH Chest X-ray dataset consisting of de-identified images of chest x-rays. The images were in the PNG format. The data was provided by the NIH Clinical Center and is available through the [NIH download site](#). We used a 250GB sample of the data with 627, 615 images across 15 classes.

The dataset was uploaded to the cnvrg platform and was cached on an NFS export from the NetApp AFF A800 storage system.

## Set up the Compute Resources

The cnvrg architecture and meta-scheduling capability allow engineers and IT professionals to attach different compute resources to a single platform. In our setup, we used the same cluster cnvrg that was deployed for running the deep-learning workloads. If you need to attach additional clusters, use the GUI, as shown in the following screenshot.

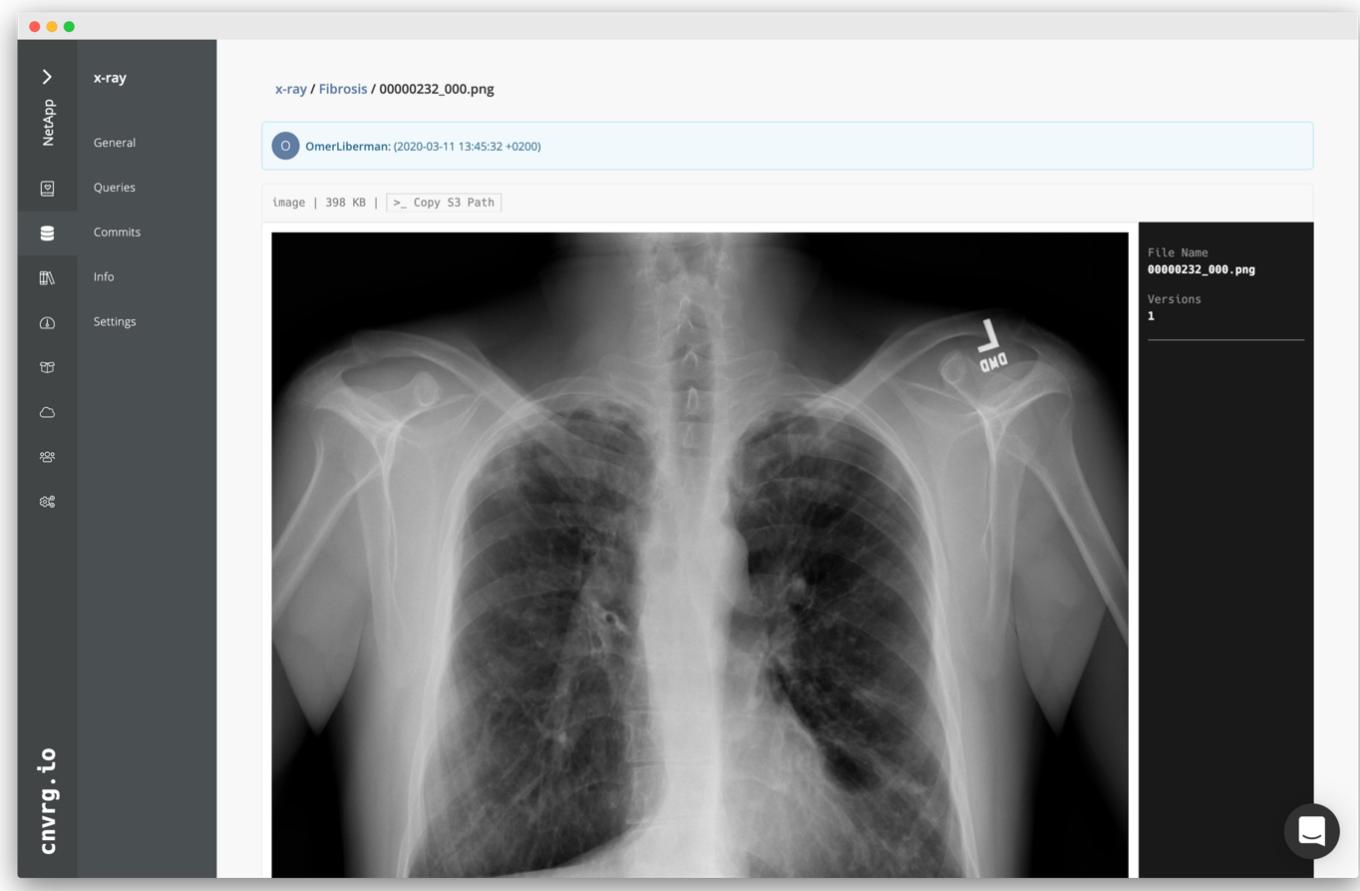


## Load Data

To upload data to the cnvrg platform, you can use the GUI or the cnvrg CLI. For large datasets, NetApp recommends using the CLI because it is a strong, scalable, and reliable tool that can handle a large number of files.

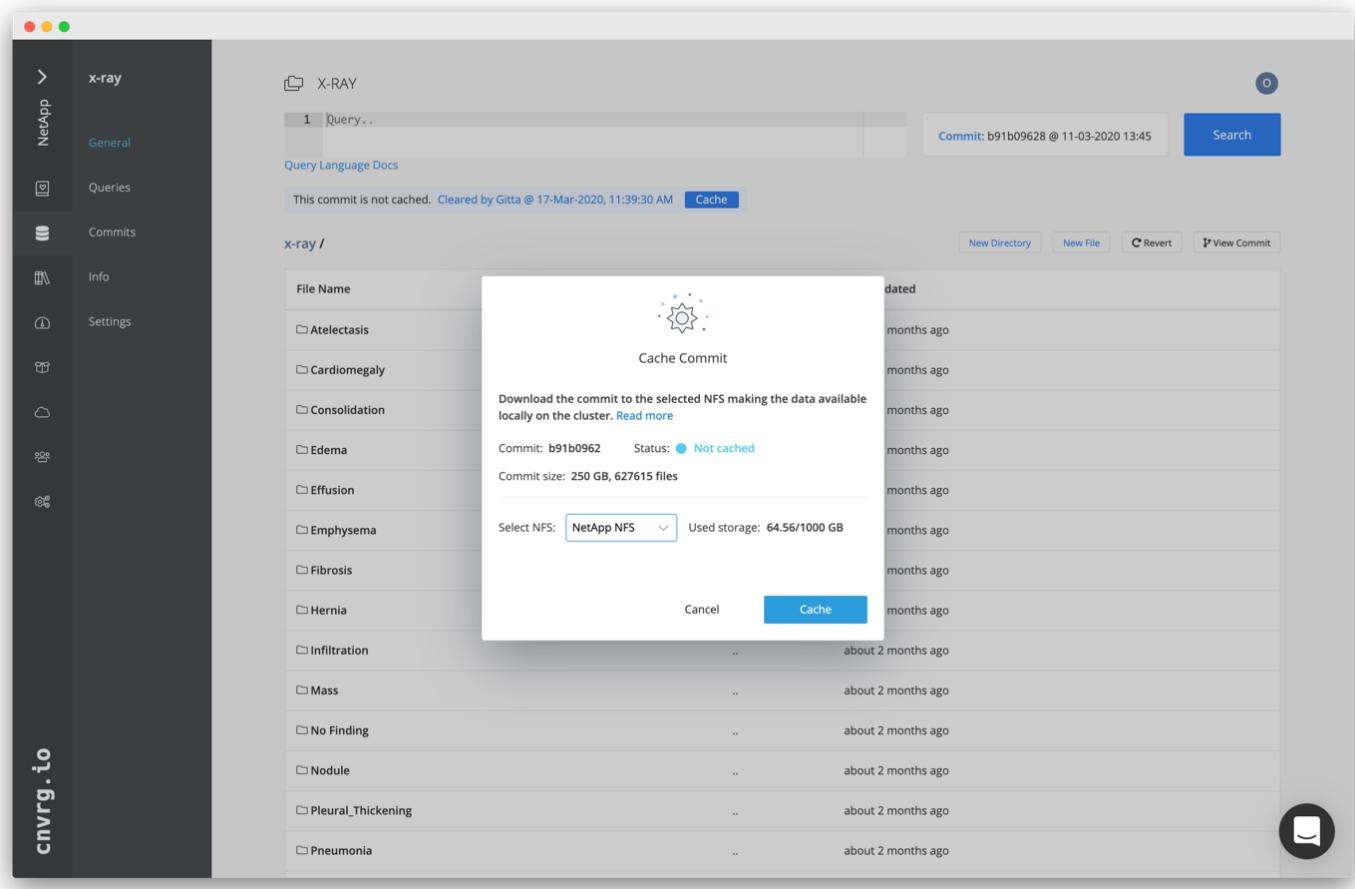
To upload data, complete the following steps:

1. Download the [cnvrg CLI](#).
2. navigate to the x-ray directory.
3. Initialize the dataset in the platform with the `cnvrg data init` command.
4. Upload all contents of the directory to the central data lake with the `cnvrg data sync` command. After the data is uploaded to the central object store (StorageGRID, S3, or others), you can browse with the GUI. The following figure shows a loaded chest X-ray fibrosis image PNG file. In addition, cnvrg versions the data so that any model you build can be reproduced down to the data version.



## Cach Data

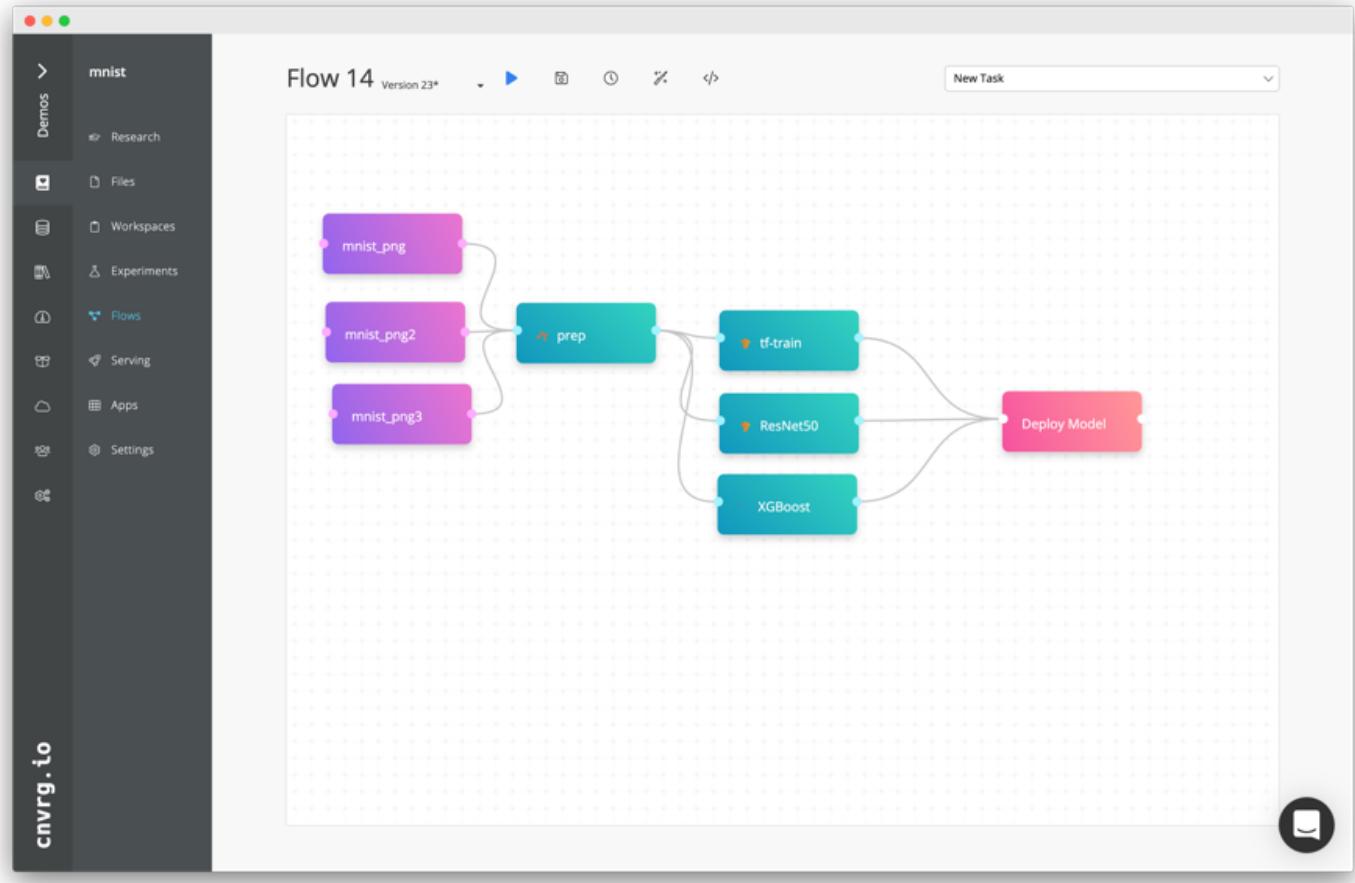
To make training faster and avoid downloading 600k+ files for each model training and experiment, we used the data-caching feature after data was initially uploaded to the central data-lake object store.



After users click Cache, cnvrg downloads the data in its specific commit from the remote object store and caches it on the ONTAP NFS volume. After it completes, the data is available for instant training. In addition, if the data is not used for a few days (for model training or exploration, for example), cnvrg automatically clears the cache.

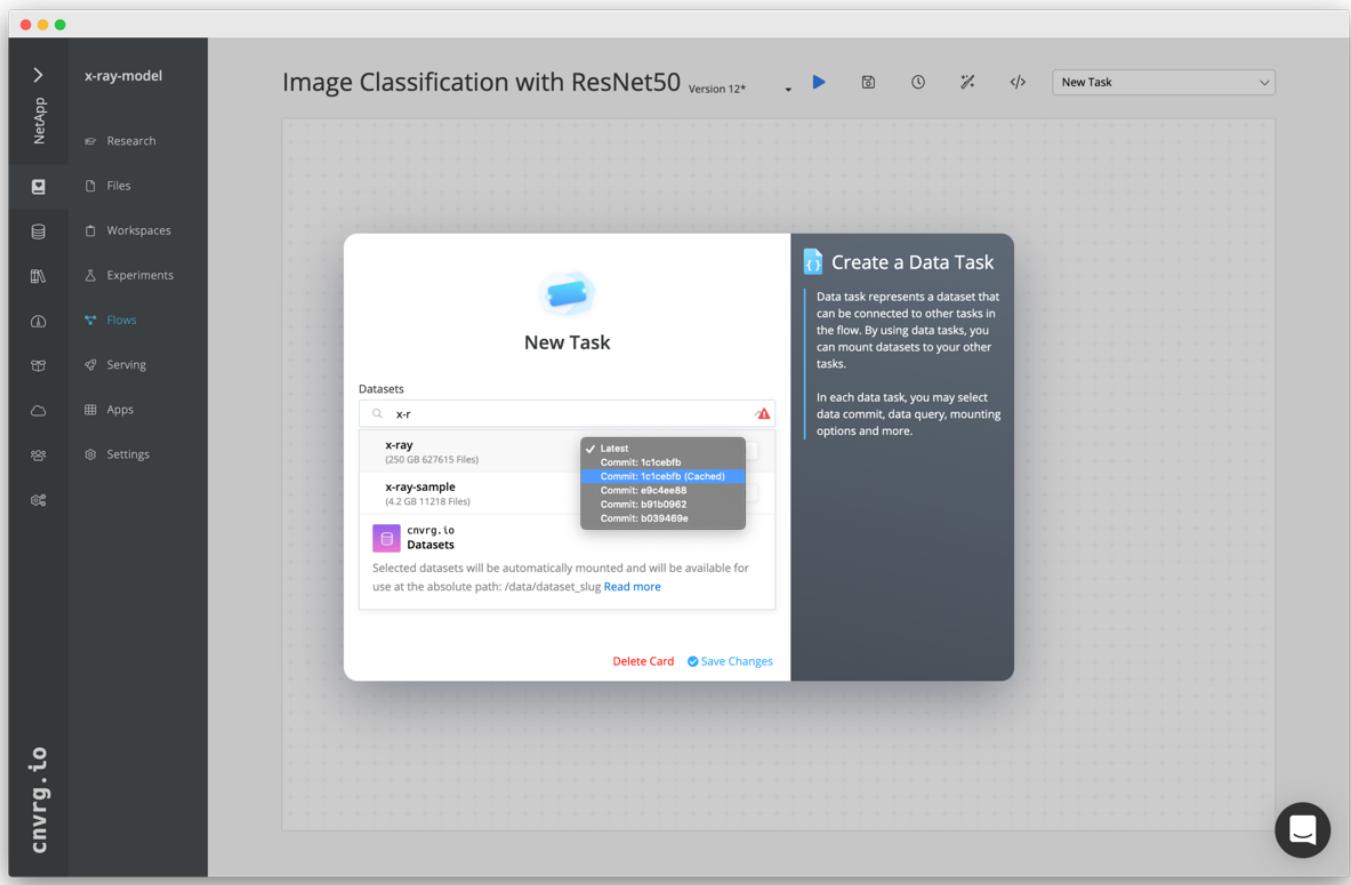
## Build an ML Pipeline with Cached Data

cnvrg flows allows you to easily build production ML pipelines. Flows are flexible, can work for any kind of ML use case, and can be created through the GUI or code. Each component in a flow can run on a different compute resource with a different Docker image, which makes it possible to build hybrid cloud and optimized ML pipelines.



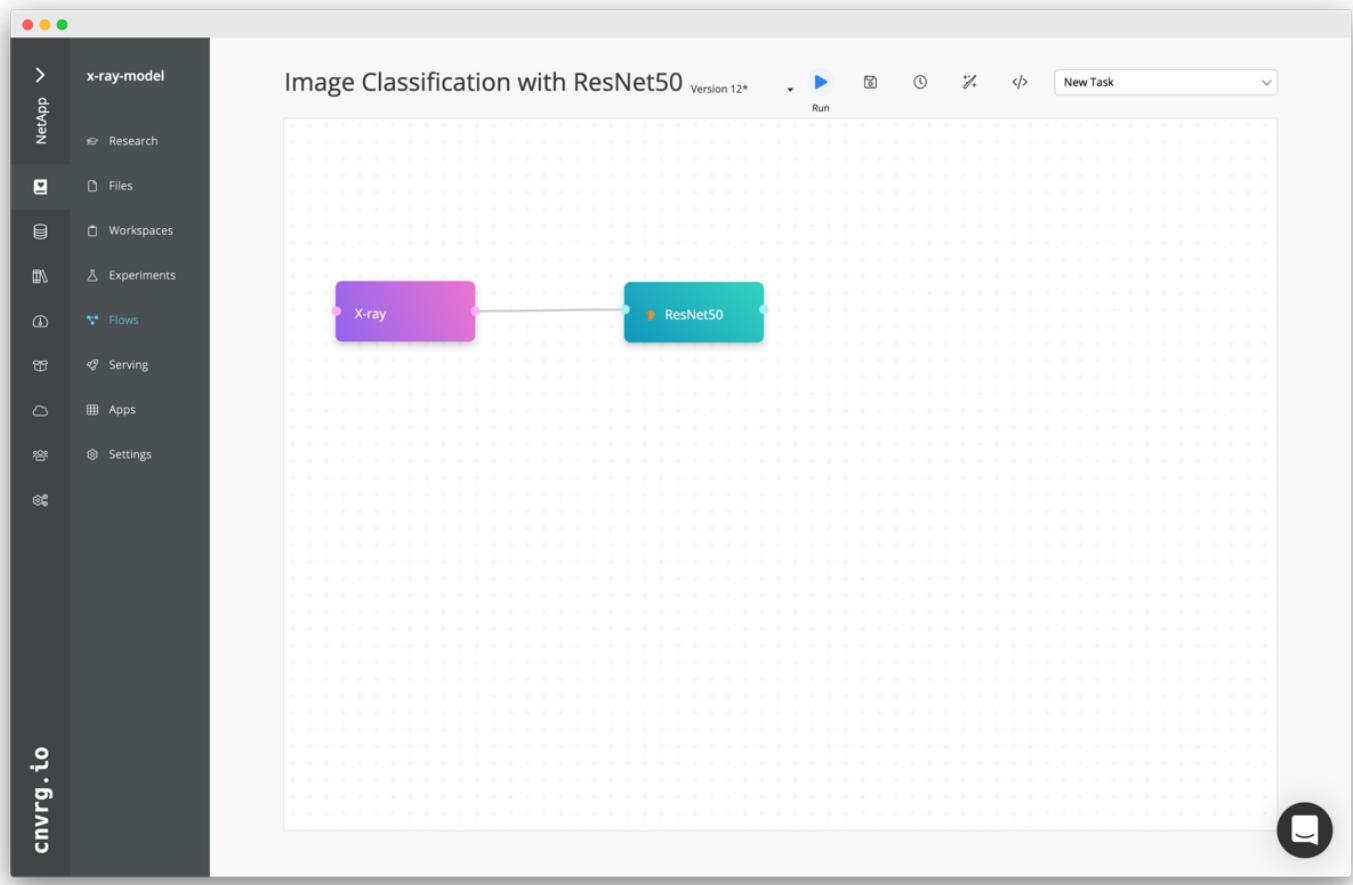
## Building the Chest X-ray Flow: Setting Data

We added our dataset to a newly created flow. When adding the dataset, you can select the specific version (commit) and indicate whether you want the cached version. In this example, we selected the cached commit.



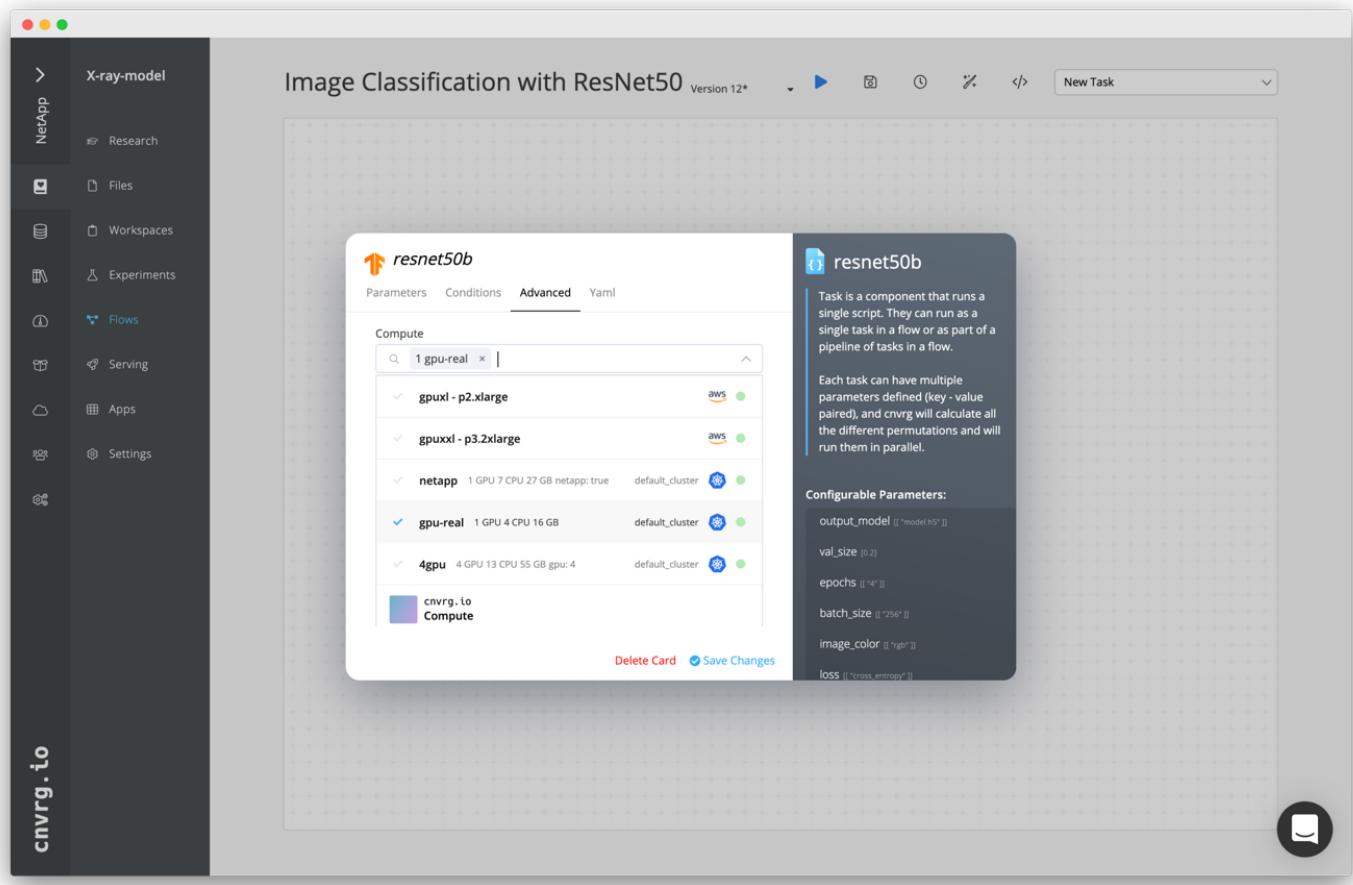
## Building the Chest X-ray Flow: Setting Training Model: ResNet50

In the pipeline, you can add any kind of custom code you want. In cnvrg, there is also the AI library, a reusable ML components collection. In the AI library, there are algorithms, scripts, data sources, and other solutions that can be used in any ML or deep learning flow. In this example, we selected the prebuilt ResNet50 module. We used default parameters such as batch\_size:128, epochs:10, and more. These parameters can be viewed in the AI Library docs. The following screenshot shows the new flow with the X-ray dataset connected to ResNet50.



## Define the Compute Resource for ResNet50

Each algorithm or component in cnvrg flows can run on a different compute instance, with a different Docker image. In our setup, we wanted to run the training algorithm on the NVIDIA DGX systems with the NetApp ONTAP AI architecture. In The following figure, we selected `gpu-real`, which is a compute template and specification for our on-premises cluster. We also created a queue of templates and selected multiple templates. In this way, if the `gpu-real` resource cannot be allocated (if, for example, other data scientists are using it), then you can enable automatic cloud-bursting by adding a cloud provider template. The following screenshot shows the use of `gpu-real` as a compute node for ResNet50.



## Tracking and Monitoring Results

After a flow is executed, cnvrg triggers the tracking and monitoring engine. Each run of a flow is automatically documented and updated in real time. Hyperparameters, metrics, resource usage (GPU utilization, and more), code version, artifacts, logs, and so on are automatically available in the Experiments section, as shown in the following two screenshots.

**X-ray train (ResNet50)**  
by yochz

**Status:** SUCCESS Duration: 33m 54s

**Input:** python3 resnet50.py --data /data/x-ray-sample-splitted --data\_test None --output\_model model.h5 --va... SHOW ALL

**Start Time:** 22-Mar-2020, 3:55:37 PM    **End Time:** 22-Mar-2020, 4:29:22 PM    **Duration:** 33m 45s    **Compute:** gpu-real    **Image:** tensorflow:20.01-tf2-py3

**Start Commit:** c0854e73    **End Commit:** a980dd8e

**CPU**    **Memory**    **Block IO**    **GPU**    **GPU Memory**

**Classes list:** ["No Finding", "Hernia", "Fibrosis", "Pleural\_Thickening", "Mass", "Infiltration", "Effusion", "Cardiomegaly", "Atelectasis", "Edema", "Consolidation", "Touch Bar Shot 2020-03-12 at 7.53.13 PM.png", "Pneumonia", "Pneumothorax", "Nodule", "Emphysemat"]

**Model:** resnet50    **GPUs found:** 1    **tensorflow local version:** 2.0.0

**GridSearch\_ID:** 2461r    **output\_layer\_activation:** softmax    **hidden\_layer\_activation:** relu    **pooling\_height:** 2  
**pooling\_width:** 2    **conv\_height:** 3    **conv\_width:** 3    **image\_height:** 224  
**image\_width:** 224    **optimizer:** adam    **dropout:** 0.3    **image\_color:** rgb  
**batch\_size:** 1024    **steps\_per\_epoch:** 10    **epochs:** 10    **val\_size:** 0.2  
**output\_model:** model.h5    **data\_test:** None    **data:** /data/x-ray-sample-splitted

**loss**

Epoch	Experiment 50 Loss
0	2.30
1	1.75
2	1.70
3	1.75
4	1.70
5	1.68
6	1.65
7	1.63
8	1.61
9	1.60
10	1.58
11	1.56

**Compare Experiments**

**Experiment 59 X Experiment 58 X Experiment 60 X Experiment 61 X Experiment 57 X Refresh**

**loss**

Epoch	Experiment 59 Loss	Experiment 58 Loss	Experiment 60 Loss	Experiment 61 Loss	Experiment 57 Loss
0	0.28	0.28	0.28	0.28	0.28
1	0.08	0.08	0.08	0.08	0.08
2	0.06	0.06	0.06	0.06	0.06
3	0.05	0.05	0.05	0.05	0.05
4	0.04	0.04	0.04	0.04	0.04
5	0.03	0.03	0.03	0.03	0.03
6	0.03	0.03	0.03	0.03	0.03
7	0.02	0.02	0.02	0.02	0.02
8	0.02	0.02	0.02	0.02	0.02
9	0.02	0.02	0.02	0.02	0.02
10	0.02	0.02	0.02	0.02	0.02
11	0.02	0.02	0.02	0.02	0.02

**val\_loss**

Epoch	Experiment 59 val_loss	Experiment 58 val_loss	Experiment 60 val_loss	Experiment 61 val_loss	Experiment 57 val_loss
0	0.06	0.06	0.06	0.06	0.06
1	0.03	0.03	0.03	0.03	0.03
2	0.03	0.03	0.03	0.03	0.03
3	0.03	0.03	0.03	0.03	0.03
4	0.03	0.03	0.03	0.03	0.03
5	0.03	0.03	0.03	0.03	0.03
6	0.03	0.03	0.03	0.03	0.03
7	0.02	0.02	0.02	0.02	0.02
8	0.02	0.02	0.02	0.02	0.02
9	0.02	0.02	0.02	0.02	0.02
10	0.02	0.02	0.02	0.02	0.02
11	0.02	0.02	0.02	0.02	0.02

Next: Conclusion

## Conclusion

NetApp and cnvrg.io have partnered to offer customers a complete data management solution for ML and DL software development. ONTAP AI provides high-performance compute and storage for any scale of operation, and cnvrg.io software streamlines data science workflows and improves resource utilization.

Next: [Acknowledgments](#)

## Acknowledgments

- Mike Oglesby, Technical Marketing Engineer, NetApp
- Santosh Rao, Senior Technical Director, NetApp

Next: [Where to Find Additional Information](#)

## Where to Find Additional Information

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

- Cnvrg.io (<https://cnvrg.io>):
  - Cnvrg CORE (free ML platform)  
<https://cnvrg.io/platform/core>
  - Cnvrg docs  
<https://app.cnvrg.io/docs>
- NVIDIA DGX-1 servers:
  - NVIDIA DGX-1 servers  
<https://www.nvidia.com/en-us/data-center/dgx-1/>
  - NVIDIA Tesla V100 Tensor Core GPU  
<https://www.nvidia.com/en-us/data-center/tesla-v100/>
  - NVIDIA GPU Cloud (NGC)  
<https://www.nvidia.com/en-us/gpu-cloud/>
- NetApp AFF systems:
  - AFF datasheet  
<https://www.netapp.com/us/media/d-3582.pdf>
  - NetApp FlashAdvantage for AFF  
<https://www.netapp.com/us/media/ds-3733.pdf>
  - ONTAP 9.x documentation

<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>

- NetApp FlexGroup technical report

<https://www.netapp.com/us/media/tr-4557.pdf>

- NetApp persistent storage for containers:

- NetApp Trident

<https://netapp.io/persistent-storage-provisioner-for-kubernetes/>

- NetApp Interoperability Matrix:

- NetApp Interoperability Matrix Tool

<http://support.netapp.com/matrix>

- ONTAP AI networking:

- Cisco Nexus 3232C Switches

<https://www.cisco.com/c/en/us/products/switches/nexus-3232c-switch/index.html>

- Mellanox Spectrum 2000 series switches

[http://www.mellanox.com/page/products\\_dyn?product\\_family=251&mtag=sn2000](http://www.mellanox.com/page/products_dyn?product_family=251&mtag=sn2000)

- ML framework and tools:

- DALI

<https://github.com/NVIDIA/DALI>

- TensorFlow: An Open-Source Machine Learning Framework for Everyone

<https://www.tensorflow.org/>

- Horovod: Uber's Open-Source Distributed Deep Learning Framework for TensorFlow

<https://eng.uber.com/horovod/>

- Enabling GPUs in the Container Runtime Ecosystem

<https://devblogs.nvidia.com/gpu-containers-runtime/>

- Docker

<https://docs.docker.com>

- Kubernetes

<https://kubernetes.io/docs/home/>

- NVIDIA DeepOps

<https://github.com/NVIDIA/deepops>

- Kubeflow  
<http://www.kubeflow.org/>
- Jupyter Notebook Server  
<http://www.jupyter.org/>

- Dataset and benchmarks:
  - NIH chest X-ray dataset

<https://nihcc.app.box.com/v/ChestXray-NIHCC>

- Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, Ronald Summers, ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases, IEEE CVPR, pp. 3462-3471, 2017TR-4841-0620

## NVA-1144: NetApp HCI AI Inferencing at the Edge Data Center with H615c and NVIDIA T4

Arvind Ramakrishnan, NetApp

This document describes how NetApp HCI can be designed to host artificial intelligence (AI) inferencing workloads at edge data center locations. The design is based on NVIDIA T4 GPU-powered NetApp HCI compute nodes, an NVIDIA Triton Inference Server, and a Kubernetes infrastructure built using NVIDIA DeepOps. The design also establishes the data pipeline between the core and edge data centers and illustrates implementation to complete the data lifecycle path.

Modern applications that are driven by AI and machine learning (ML) have pushed the limits of the internet. End users and devices demand access to applications, data, and services at any place and any time, with minimal latency. To meet these demands, data centers are moving closer to their users to boost performance, reduce back-and-forth data transfer, and provide cost-effective ways to meet user requirements.

In the context of AI, the core data center is a platform that provides centralized services, such as machine learning and analytics, and the edge data centers are where the real-time production data is subject to inferencing. These edge data centers are usually connected to a core data center. They provide end-user services and serve as a staging layer for data generated by IoT devices that need additional processing and that is too time sensitive to be transmitted back to a centralized core.

This document describes a reference architecture for AI inferencing that uses NetApp HCI as the base platform.

### Customer Value

NetApp HCI offers differentiation in the hyperconverged market for this inferencing solution, including the following advantages:

- A disaggregated architecture allows independent scaling of compute and storage and lowers the virtualization licensing costs and performance tax on independent NetApp HCI storage nodes.
- NetApp Element storage provides quality of service (QoS) for each storage volume, which provides guaranteed storage performance for workloads on NetApp HCI. Therefore, adjacent workloads do not negatively affect inferencing performance.
- A data fabric powered by NetApp allows data to be replicated from core to edge to cloud data centers, which moves data closer to where application needs it.

- With a data fabric powered by NetApp and NetApp FlexCache software, AI deep learning models trained on NetApp ONTAP AI can be accessed from NetApp HCI without having to export the model.
- NetApp HCI can host inference servers on the same infrastructure concurrently with multiple workloads, either virtual-machine (VM) or container-based, without performance degradation.
- NetApp HCI is certified as NVIDIA GPU Cloud (NGC) ready for NVIDIA AI containerized applications.
- NGC-ready means that the stack is validated by NVIDIA, is purpose built for AI, and enterprise support is available through NGC Support Services.
- With its extensive AI portfolio, NetApp can support the entire spectrum of AI use cases from edge to core to cloud, including ONTAP AI for training and inferencing, Cloud Volumes Service and Azure NetApp Files for training in the cloud, and inferencing on the edge with NetApp HCI.

[Next: Use Cases](#)

## Use Cases

Although all applications today are not AI driven, they are evolving capabilities that allow them to access the immense benefits of AI. To support the adoption of AI, applications need an infrastructure that provides them with the resources needed to function at an optimum level and support their continuing evolution.

For AI-driven applications, edge locations act as a major source of data. Available data can be used for training when collected from multiple edge locations over a period of time to form a training dataset. The trained model can then be deployed back to the edge locations where the data was collected, enabling faster inferencing without the need to repeatedly transfer production data to a dedicated inferencing platform.

The NetApp HCI AI inferencing solution, powered by NetApp H615c compute nodes with NVIDIA T4 GPUs and NetApp cloud-connected storage systems, was developed and verified by NetApp and NVIDIA. NetApp HCI simplifies the deployment of AI inferencing solutions at edge data centers by addressing areas of ambiguity, eliminating complexities in the design and ending guesswork.

This solution gives IT organizations a prescriptive architecture that:

- Enables AI inferencing at edge data centers
- Optimizes consumption of GPU resources
- Provides a Kubernetes-based inferencing platform for flexibility and scalability
- Eliminates design complexities

Edge data centers manage and process data at locations that are very near to the generation point. This proximity increases the efficiency and reduces the latency involved in handling data. Many vertical markets have realized the benefits of an edge data center and are heavily adopting this distributed approach to data processing.

The following table lists the edge verticals and applications.

Vertical	Applications
Medical	Computer-aided diagnostics assist medical staff in early disease detection
Oil and gas	Autonomous inspection of remote production facilities, video, and image analytics

<b>Vertical</b>	<b>Applications</b>
Aviation	Air traffic control assistance and real-time video feed analytics
Media and entertainment	Audio/video content filtering to deliver family-friendly content
Business analytics	Brand recognition to analyze brand appearance in live-streamed televised events
E-Commerce	Smart bundling of supplier offers to find ideal merchant and warehouse combinations
Retail	Automated checkout to recognize items a customer placed in cart and facilitate digital payment
Smart city	Improve traffic flow, optimize parking, and enhance pedestrian and cyclist safety
Manufacturing	Quality control, assembly-line monitoring, and defect identification
Customer service	Customer service automation to analyze and triage inquiries (phone, email, and social media)
Agriculture	Intelligent farm operation and activity planning, to optimize fertilizer and herbicide application

## Target Audience

The target audience for the solution includes the following groups:

- Data scientists
- IT architects
- Field consultants
- Professional services
- IT managers
- Anyone else who needs an infrastructure that delivers IT innovation and robust data and application services at edge locations

[Next: Architecture](#)

## Architecture

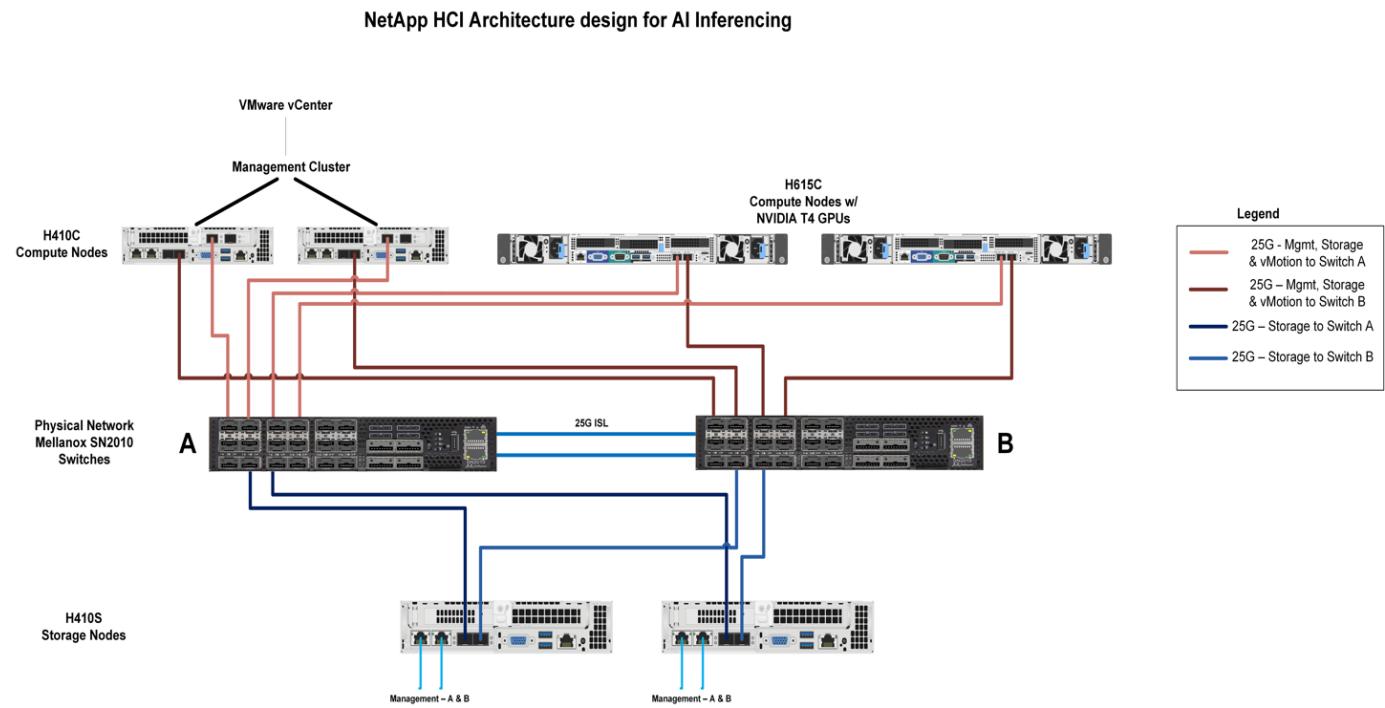
### Solution Technology

This solution is designed with a NetApp HCI system that contains the following components:

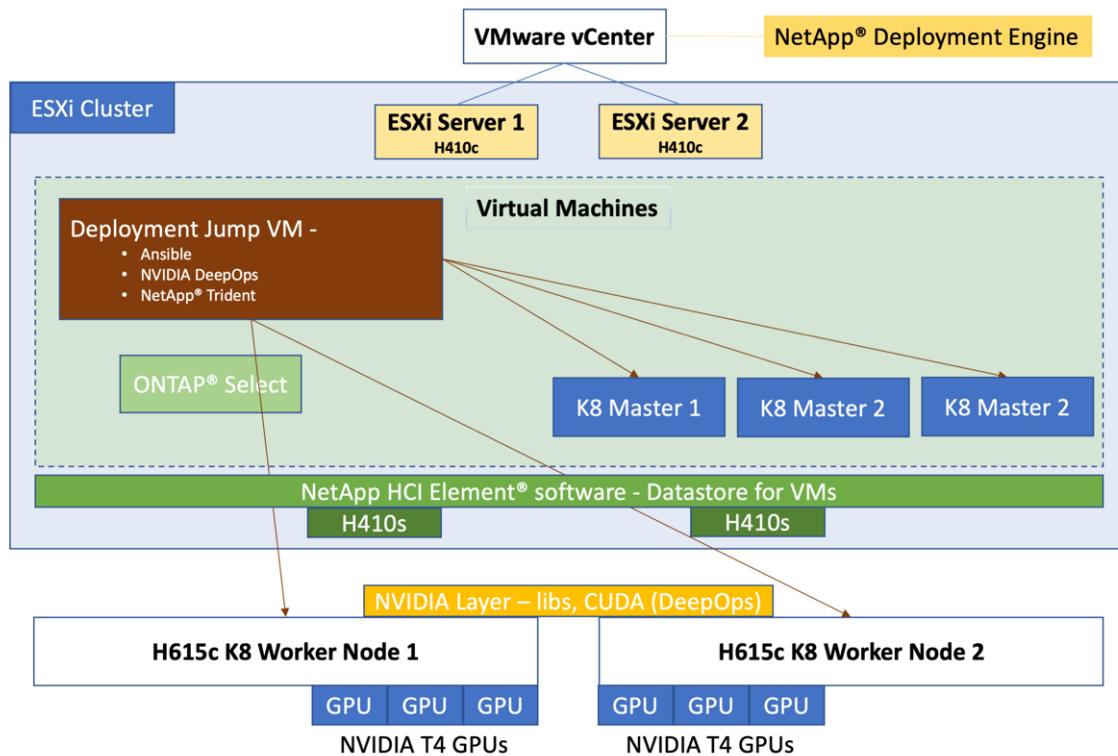
- Two H615c compute nodes with NVIDIA T4 GPUs
- Two H410c compute nodes
- Two H410s storage nodes
- Two Mellanox SN2010 10GbE/25GbE switches

## Architectural Diagram

The following diagram illustrates the solution architecture for the NetApp HCI AI inferencing solution.



The following diagram illustrates the virtual and physical elements of this solution.



A VMware infrastructure is used to host the management services required by this inferencing solution. These services do not need to be deployed on a dedicated infrastructure; they can coexist with any existing workloads. The NetApp Deployment Engine (NDE) uses the H410c and H410s nodes to deploy the VMware infrastructure.

After NDE has completed the configuration, the following components are deployed as VMs in the virtual infrastructure:

- **Deployment Jump VM.** Used to automate the deployment of NVIDIA DeepOps. See [NVIDIA DeepOps](#) and storage management using NetApp Trident.
- **ONTAP Select.** An instance of ONTAP Select is deployed to provide NFS file services and persistent storage to the AI workload running on Kubernetes.
- **Kubernetes Masters.** During deployment, three VMs are installed and configured with a supported Linux distribution and configured as Kubernetes master nodes. After the management services have been set up, two H615c compute nodes with NVIDIA T4 GPUs are installed with a supported Linux distribution. These two nodes function as the Kubernetes worker nodes and provide the infrastructure for the inferencing platform.

## Hardware Requirements

The following table lists the hardware components that are required to implement the solution. The hardware components that are used in any particular implementation of the solution might vary based on customer requirements.

Layer	Product Family	Quantity	Details
Compute	H615c	2	3 NVIDIA Tesla T4 GPUs per node
	H410c	2	Compute nodes for management infrastructure
Storage	H410s	2	Storage for OS and workload
Network	Mellanox SN2010	2	10G/25G switches

## Software Requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any particular implementation of the solution might vary based on customer requirements.

Layer	Software	Version
Storage	NetApp Element software	12.0.0.333
	ONTAP Select	9.7
	NetApp Trident	20.07
NetApp HCI engine	NDE	1.8
Hypervisor	Hypervisor	VMware vSphere ESXi 6.7U1
	Hypervisor Management System	VMware vCenter Server 6.7U1
Inferencing Platform	NVIDIA DeepOps	20.08
	NVIDIA GPU Operator	1.1.7
	Ansible	2.9.5

<b>Layer</b>	<b>Software</b>	<b>Version</b>
	Kubernetes	1.17.9
	Docker	Docker CE 18.09.7
	CUDA Version	10.2
	GPU Device Plugin	0.6.0
	Helm	3.1.2
	NVIDIA Tesla Driver	440.64.00
	NVIDIA Triton Inference Server	2.1.0 – NGC Container v20.07
K8 Master VMs	Linux	<p>Any supported distribution across NetApp IMT, NVIDIA DeepOps, and GPUOperator</p> <p>Ubuntu 18.04.4 LTS was used in this solution</p> <p>Kernel version: 4.15</p>
Host OS/ K8 Worker Nodes	Linux	<p>Any supported distribution across NetApp IMT, NVIDIA DeepOps, and GPUOperator</p> <p>Ubuntu 18.04.4 LTS was used in this solution</p> <p>Kernel version: 4.15</p>

[Next: Design Considerations](#)

## Design Considerations

### Network Design

The switches used to handle the NetApp HCI traffic require a specific configuration for successful deployment.

Consult the NetApp HCI Network Setup Guide for the physical cabling and switch details. This solution uses a two-cable design for compute nodes. Optionally, compute nodes can be configured in a six-node cable design affording options for deployment of compute nodes.

The diagram under [Architecture](#) depicts the network topology of this NetApp HCI solution with a two-cable design for the compute nodes.

### Compute Design

The NetApp HCI compute nodes are available in two form factors, half-width and full-width, and in two rack unit sizes, 1 RU and 2 RU. The 410c nodes used in this solution are half-width and 1 RU and are housed in a chassis that can hold a maximum of four such nodes. The other compute node that is used in this solution is the H615c, which is a full-width node, 1 RU in size. The H410c nodes are based on Intel Skylake processors, and the H615c nodes are based on the second-generation Intel Cascade Lake processors. NVIDIA GPUs can be added to the H615c nodes, and each node can host a maximum of three NVIDIA Tesla T4 16GB GPUs.

The H615c nodes are the latest series of compute nodes for NetApp HCI and the second series that can support GPUs. The first model to support GPUs is the H610c node (full width, 2RU), which can support two

NVIDIA Tesla M10 GPUs.

In this solution, H615c nodes are preferred over H610c nodes because of the following advantages:

- Reduced data center footprint, critical for edge deployments
- Support for a newer generation of GPUs designed for faster inferencing
- Reduced power consumption
- Reduced heat dissipation

#### NVIDIA T4 GPUs

The resource requirements of inferencing are nowhere close to those of training workloads. In fact, most modern hand-held devices are capable of handling small amounts of inferencing without powerful resources like GPUs. However, for mission-critical applications and data centers that are dealing with a wide variety of applications that demand very low inferencing latencies while subject to extreme parallelization and massive input batch sizes, the GPUs play a key role in reducing inference time and help to boost application performance.

The NVIDIA Tesla T4 is an x16 PCIe Gen3 single-slot low-profile GPU based on the Turing architecture. The T4 GPUs deliver universal inference acceleration that spans applications such as image classification and tagging, video analytics, natural language processing, automatic speech recognition, and intelligent search. The breadth of the Tesla T4's inferencing capabilities enables it to be used in enterprise solutions and edge devices.

These GPUs are ideal for deployment in edge infrastructures due to their low power consumption and small PCIe form factor. The size of the T4 GPUs enables the installation of two T4 GPUs in the same space as a double-slot full-sized GPU. Although they are small, with 16GB memory, the T4s can support large ML models or run inference on multiple smaller models simultaneously.

The Turing-based T4 GPUs include an enhanced version of Tensor Cores and support a full range of precisions for inferencing FP32, FP16, INT8, and INT4. The GPU includes 2,560 CUDA cores and 320 Tensor Cores, delivering up to 130 tera operations per second (TOPS) of INT8 and up to 260 TOPS of INT4 inferencing performance. When compared to CPU-based inferencing, the Tesla T4, powered by the new Turing Tensor Cores, delivers up to 40 times higher inference performance.

The Turing Tensor Cores accelerate the matrix-matrix multiplication at the heart of neural network training and inferencing functions. They particularly excel at inference computations in which useful and relevant information can be inferred and delivered by a trained deep neural network based on a given input.

The Turing GPU architecture inherits the enhanced Multi-Process Service (MPS) feature that was introduced in the Volta architecture. Compared to Pascal-based Tesla GPUs, MPS on Tesla T4 improves inference performance for small batch sizes, reduces launch latency, improves QoS, and enables the servicing of higher numbers of concurrent client requests.

The NVIDIA T4 GPU is a part of the NVIDIA AI Inference Platform that supports all AI frameworks and provides comprehensive tooling and integrations to drastically simplify the development and deployment of advanced AI.

#### Storage Design: Element Software

NetApp Element software powers the storage of the NetApp HCI systems. It delivers agile automation through scale-out flexibility and guaranteed application performance to accelerate new services.

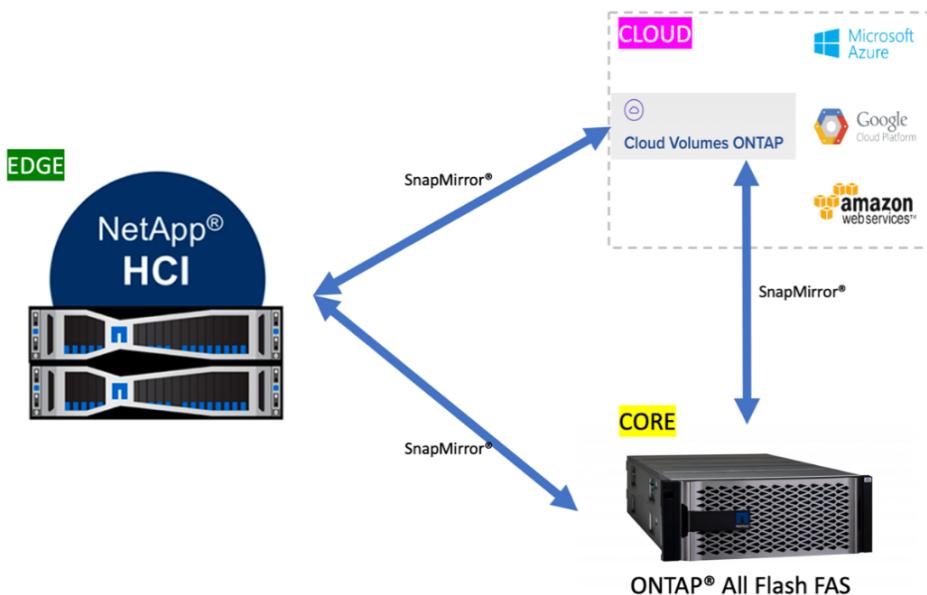
Storage nodes can be added to the system non-disruptively in increments of one, and the storage resources

are made available to the applications instantly. Every new node added to the system delivers a precise amount of additional performance and capacity to a usable pool. The data is automatically load balanced in the background across all nodes in the cluster, maintaining even utilization as the system grows.

Element software supports the NetApp HCI system to comfortably host multiple workloads by guaranteeing QoS to each workload. By providing fine-grained performance control with minimum, maximum, and burst settings for each workload, the software allows well-planned consolidations while protecting application performance. It decouples performance from capacity and allows each volume to be allocated with a specific amount of capacity and performance. These specifications can be modified dynamically without any interruption to data access.

As illustrated in the following figure, Element software integrates with NetApp ONTAP to enable data mobility between NetApp storage systems that are running different storage operating systems. Data can be moved from the Element software to ONTAP or vice versa by using NetApp SnapMirror technology. Element uses the same technology to provide cloud connectivity by integrating with NetApp Cloud Volumes ONTAP, which enables data mobility from the edge to the core and to multiple public cloud service providers.

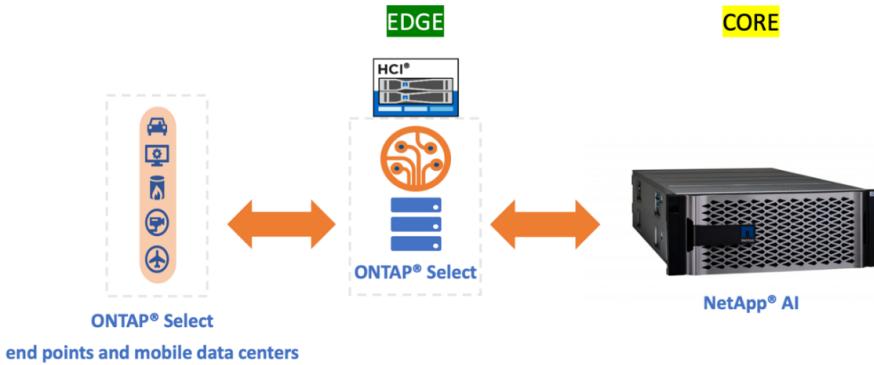
In this solution, the Element-backed storage provides the storage services that are required to run the workloads and applications on the NetApp HCI system.



### Storage Design: ONTAP Select

NetApp ONTAP Select introduces a software-defined data storage service model on top of NetApp HCI. It builds on NetApp HCI capabilities, adding a rich set of file and data services to the HCI platform while extending the data fabric.

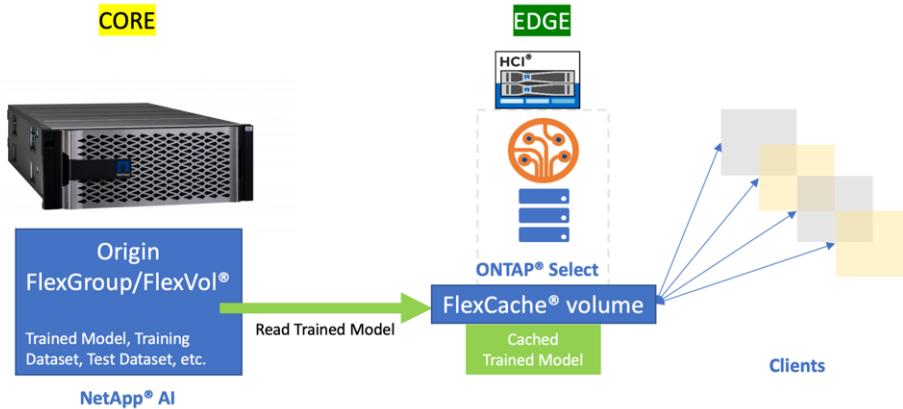
Although ONTAP Select is an optional component for implementing this solution, it does provide a host of benefits, including data gathering, protection, mobility, and so on, that are extremely useful in the context of the overall AI data lifecycle. It helps to simplify several day-to-day challenges for data handling, including ingestion, collection, training, deployment, and tiering.



ONTAP Select can run as a VM on VMware and still bring in most of the ONTAP capabilities that are available when it is running on a dedicated FAS platform, such as the following:

- Support for NFS and CIFS
- NetApp FlexClone technology
- NetApp FlexCache technology
- NetApp ONTAP FlexGroup volumes
- NetApp SnapMirror software

ONTAP Select can be used to leverage the FlexCache feature, which helps to reduce data-read latencies by caching frequently read data from a back-end origin volume, as is shown in the following figure. In the case of high-end inferencing applications with a lot of parallelization, multiple instances of the same model are deployed across the inferencing platform, leading to multiple reads of the same model. Newer versions of the trained model can be seamlessly introduced to the inferencing platform by verifying that the desired model is available in the origin or source volume.



13

© 2019 NetApp, Inc. All rights reserved. — NETAPP CONFIDENTIAL —



## NetApp Trident

NetApp Trident is an open-source dynamic storage orchestrator that allows you to manage storage resources across all major NetApp storage platforms. It integrates with Kubernetes natively so that persistent volumes (PVs) can be provisioned on demand with native Kubernetes interfaces and constructs. Trident enables microservices and containerized applications to use enterprise-class storage services such as QoS, storage efficiencies, and cloning to meet the persistent storage demands of applications.

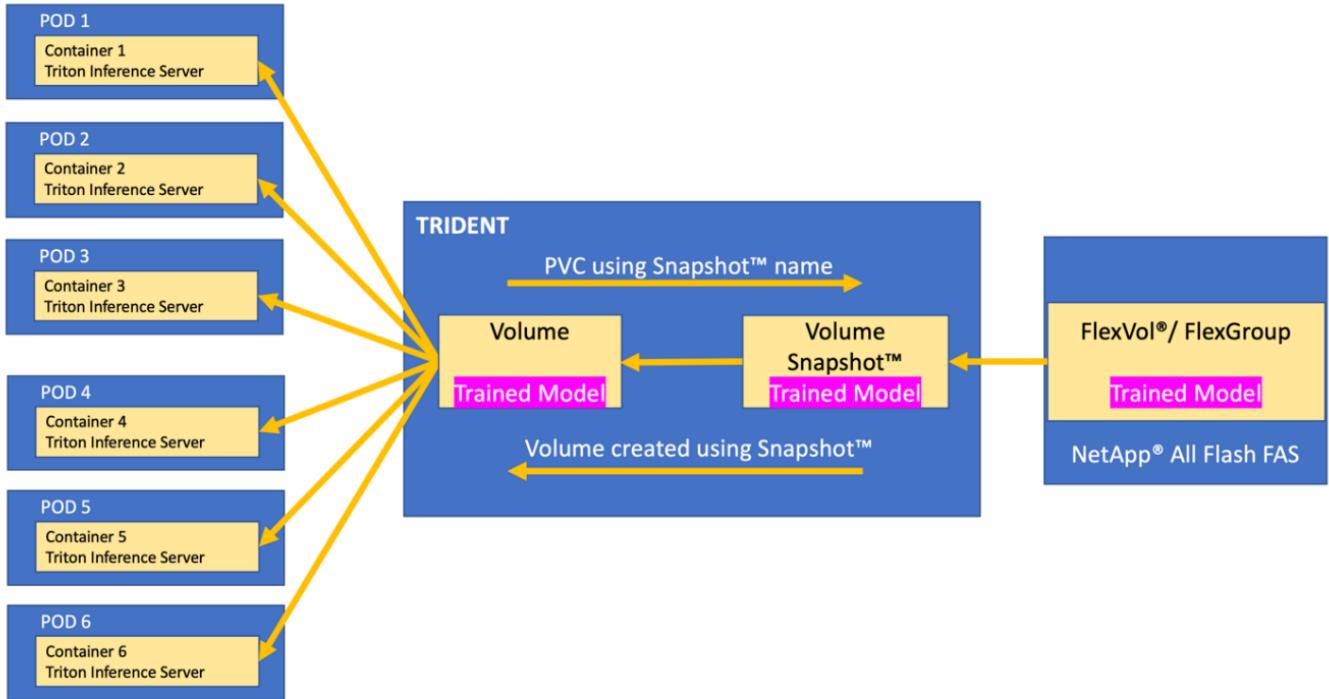
Containers are among the most popular methods of packaging and deploying applications, and Kubernetes is one of the most popular platforms for hosting containerized applications. In this solution, the inferencing platform is built on top of a Kubernetes infrastructure.

Trident currently supports storage orchestration across the following platforms:

- ONTAP: NetApp AFF, FAS, and Select
- Element software: NetApp HCI and NetApp SolidFire all-flash storage
- NetApp SANtricity software: E-Series and EF-series
- Cloud Volumes ONTAP
- Azure NetApp Files
- NetApp Cloud Volumes Service: AWS and Google Cloud

Trident is a simple but powerful tool to enable storage orchestration not just across multiple storage platforms, but also across the entire spectrum of the AI data lifecycle, ranging from the edge to the core to the cloud.

Trident can be used to provision a PV from a NetApp Snapshot copy that makes up the trained model. The following figure illustrates the Trident workflow in which a persistent volume claim (PVC) is created by referring to an existing Snapshot copy. Following this, Trident creates a volume by using the Snapshot copy.



This method of introducing trained models from a Snapshot copy supports robust model versioning. It simplifies the process of introducing newer versions of models to applications and switching inferencing between different versions of the model.

### NVIDIA DeepOps

NVIDIA DeepOps is a modular collection of Ansible scripts that can be used to automate the deployment of a Kubernetes infrastructure. There are multiple deployment tools available that can automate the deployment of a Kubernetes cluster. In this solution, DeepOps is the preferred choice because it does not just deploy a Kubernetes infrastructure, it also installs the necessary GPU drivers, NVIDIA Container Runtime for Docker (nvidia-docker2), and various other dependencies for GPU-accelerated work. It encapsulates the best practices for NVIDIA GPUs and can be customized or run as individual components as needed.

DeepOps internally uses Kubespray to deploy Kubernetes, and it is included as a submodule in DeepOps. Therefore, common Kubernetes cluster management operations such as adding nodes, removing nodes, and cluster upgrades should be performed using Kubespray.

A software based L2 LoadBalancer using MetalLB and an Ingress Controller based on NGINX are also deployed as part of this solution by using the scripts that are available with DeepOps.

In this solution, three Kubernetes master nodes are deployed as VMs, and the two H615c compute nodes with NVIDIA Tesla T4 GPUs are set up as Kubernetes worker nodes.

### NVIDIA GPU Operator

The GPU operator deploys the NVIDIA k8s-device-plugin for GPU support and runs the NVIDIA drivers as containers. It is based on the Kubernetes operator framework, which helps to automate the management of all NVIDIA software components that are needed to provision GPUs. The components include NVIDIA drivers, Kubernetes device plug-in for GPUs, the NVIDIA container runtime, and automatic node labeling, which is used in tandem with Kubernetes Node Feature Discovery.

The GPU operator is an important component of the [NVIDIA EGX](#) software-defined platform that is designed to make large-scale hybrid-cloud and edge operations possible and efficient. It is specifically useful when the Kubernetes cluster needs to scale quickly—for example, when provisioning additional GPU-based worker nodes and managing the lifecycle of the underlying software components. Because the GPU operator runs everything as containers, including NVIDIA drivers, administrators can easily swap various components by simply starting or stopping containers.

### NVIDIA Triton Inference Server

NVIDIA Triton Inference Server (Triton Server) simplifies the deployment of AI inferencing solutions in production data centers. This microservice is specifically designed for inferencing in production data centers. It maximizes GPU utilization and integrates seamlessly into DevOps deployments with Docker and Kubernetes.

Triton Server provides a common solution for AI inferencing. Therefore, researchers can focus on creating high-quality trained models, DevOps engineers can focus on deployment, and developers can focus on applications without the need to redesign the platform for each AI-powered application.

Here are some of the key features of Triton Server:

- **Support for multiple frameworks.** Triton Server can handle a mix of models, and the number of models is limited only by system disk and memory resources. It can support the TensorRT, TensorFlow GraphDef, TensorFlow SavedModel, ONNX, PyTorch, and Caffe2 NetDef model formats.
- \*Concurrent model execution. \*Multiple models or multiple instances of the same model can be run simultaneously on a GPU.
- **Multi-GPU support.** Triton Server can maximize GPU utilization by enabling inference for multiple models on one or more GPUs.
- **Support for batching.** Triton Server can accept requests for a batch of inputs and respond with the corresponding batch of outputs. The inference server supports multiple scheduling and batching algorithms that combine individual inference requests together to improve inference throughput. Batching algorithms are available for both stateless and stateful applications and need to be used appropriately. These scheduling and batching decisions are transparent to the client that is requesting inference.
- **Ensemble support.** An ensemble is a pipeline with multiple models with connections of input and output tensors between those models. An inference request can be made to an ensemble, which results in the execution of the complete pipeline.
- **Metrics.** Metrics are details about GPU utilization, server throughput, server latency, and health for auto scaling and load balancing.

NetApp HCI is a hybrid multi-cloud infrastructure that can host multiple workloads and applications, and the Triton Inference Server is well equipped to support the inferencing requirements of multiple applications.

In this solution, Triton Server is deployed on the Kubernetes cluster using a deployment file. With this method, the default configuration of Triton Server can be overridden and customized as required. Triton Server also provides an inference service using an HTTP or GRPC endpoint, allowing remote clients to request inferencing for any model that is being managed by the server.

A Persistent Volume is presented via NetApp Trident to the container that runs the Triton Inference Server and this persistent volume is configured as the model repository for the Inference server.

The Triton Inference Server is deployed with varying sets of resources using Kubernetes deployment files, and each server instance is presented with a LoadBalancer front end for seamless scalability. This approach also illustrates the flexibility and simplicity with which resources can be allocated to the inferencing workloads.

[Next: Deploying NetApp HCI – AI Inferencing at the Edge](#)

## Overview

This section describes the steps required to deploy the AI inferencing platform using NetApp HCI. The following list provides the high-level tasks involved in the setup:

1. [Configure network switches](#)
2. [Deploy the VMware virtual infrastructure on NetApp HCI using NDE](#)
3. [Configure the H615c compute nodes to be used as K8 worker nodes](#)
4. [Set up the deployment jump VM and K8 master VMs](#)
5. [Deploy a Kubernetes cluster with NVIDIA DeepOps](#)
6. [Deploy ONTAP Select within the virtual infrastructure](#)
7. [Deploy NetApp Trident](#)
8. [Deploy NVIDIA Triton inference Server](#)
9. [Deploy the client for the Triton inference server](#)
10. [Collect inference metrics from the Triton inference server](#)

### Configure Network Switches (Automated Deployment)

#### Prepare Required VLAN IDs

The following table lists the necessary VLANs for deployment, as outlined in this solution validation. You should configure these VLANs on the network switches prior to executing NDE.

Network Segment	Details	VLAN ID
Out-of-band management network	Network for HCI terminal user interface (TUI)	16
In-band management network	Network for accessing management interfaces of nodes, hosts, and guests	3488
VMware vMotion	Network for live migration of VMs	3489
iSCSI SAN storage	Network for iSCSI storage traffic	3490
Application	Network for Application traffic	3487
NFS	Network for NFS storage traffic	3491
IPL*	Interpeer link between Mellanox switches	4000
Native	Native VLAN	2

\*Only for Mellanox switches

#### Switch Configuration

This solution uses Mellanox SN2010 switches running Onyx. The Mellanox switches are configured using an Ansible playbook. Prior to running the Ansible playbook, you should perform the initial configuration of the switches manually:

1. Install and cable the switches to the uplink switch, compute, and storage nodes.
2. Power on the switches and configure them with the following details:
  - a. Host name
  - b. Management IP and gateway
  - c. NTP
3. Log into the Mellanox switches and run the following commands:

```
configuration write to pre-ansible  
configuration write to post-ansible
```

The `pre-ansible` configuration file created can be used to restore the switch's configuration to the state before the Ansible playbook execution.

The switch configuration for this solution is stored in the `post-ansible` configuration file.

4. The configuration playbook for Mellanox switches that follows best practices and requirements for NetApp HCI can be downloaded from the [NetApp HCI Toolkit](#).



The HCI Toolkit also provides a playbook to setup Cisco Nexus switches with similar best practices and requirements for NetApp HCI.



Additional guidance on populating the variables and executing the playbook is available in the respective switch README.md file.

5. Fill out the credentials to access the switches and variables needed for the environment. The following text is a sample of the variable file for this solution.

```
# vars file for nar_hci_mellanox_deploy  
#These set of variables will setup the Mellanox switches for NetApp HCI  
that uses a 2-cable compute connectivity option.  
#Ansible connection variables for mellanox  
ansible_connection: network_cli  
ansible_network_os: onyx  
#-----  
# Primary Variables  
#-----  
#Necessary VLANs for Standard NetApp HCI Deployment [native, Management,  
iSCSI_Storage, vMotion, VM_Network, IPL]  
#Any additional VLANs can be added to this in the prescribed format  
below  
netapp_hci_vlans:  
- {vlan_id: 2 , vlan_name: "Native" }  
- {vlan_id: 3488 , vlan_name: "IB-Management" }  
- {vlan_id: 3490 , vlan_name: "iSCSI_Storage" }  
- {vlan_id: 3489 , vlan_name: "vMotion" }
```

```

- {vlan_id: 3491 , vlan_name: "NFS " }
- {vlan_id: 3487 , vlan_name: "App_Network" }
- {vlan_id: 4000 , vlan_name: "IPL"}#Modify the VLAN IDs to suit your
environment
#Spanning-tree protocol type for uplink connections.
#The valid options are 'network' and 'normal'; selection depends on the
uplink switch model.
uplink_stp_type: network
-----
# IPL variables
-----
#Inter-Peer Link Portchannel
#ipl_portchannel to be defined in the format - Po100
ipl_portchannel: Po100
#Inter-Peer Link Addresses
#The IPL IP address should not be part of the management network. This
is typically a private network
ipl_ipaddr_a: 10.0.0.1
ipl_ipaddr_b: 10.0.0.2
#Define the subnet mask in CIDR number format. Eg: For subnet /22, use
ipl_ip_subnet: 22
ipl_ip_subnet: 24
#Inter-Peer Link Interfaces
#members to be defined with Eth in the format. Eg: Eth1/1
peer_link_interfaces:
  members: ['Eth1/20', 'Eth1/22']
  description: "peer link interfaces"
#MLAG VIP IP address should be in the same subnet as that of the
switches' mgmt0 interface subnet
#mlag_vip_ip to be defined in the format - <vip_ip>/<subnet_mask>. Eg:
x.x.x.x/y
mlag_vip_ip: <>
#MLAG VIP Domain Name
#The mlag domain must be unique name for each mlag domain.
#In case you have more than one pair of MLAG switches on the same
network, each domain (consist of two switches) should be configured with
different name.
mlag_domain_name: MLAG-VIP-DOM
-----
# Interface Details
-----
#Storage Bond10G Interface details
#members to be defined with Eth in the format. Eg: Eth1/1
#Only numerical digits between 100 to 1000 allowed for mlag_id
#Operational link speed [variable 'speed' below] to be defined in terms
of bytes.

```

```

#For 10 Gigabyte operational speed, define 10G. [Possible values - 10G and 25G]
#Interface descriptions append storage node data port numbers assuming all Storage Nodes' Port C -> Mellanox Switch A and all Storage Nodes' Port D -> Mellanox Switch B
#List the storage Bond10G interfaces, their description, speed and MLAG IDs in list of dictionaries format
storage_interfaces:
- {members: "Eth1/1", description: "HCI_Storage_Node_01", mlag_id: 101, speed: 25G}
- {members: "Eth1/2", description: "HCI_Storage_Node_02", mlag_id: 102, speed: 25G}
#In case of additional storage nodes, add them here
#Storage Bond1G Interface
#Mention whether or not these Mellanox switches will also be used for Storage Node Mgmt connections
#Possible inputs for storage_mgmt are 'yes' and 'no'
storage_mgmt: <>yes or no>>
#Storage Bond1G (Mgmt) interface details. Only if 'storage_mgmt' is set to 'yes'
#Members to be defined with Eth in the format. Eg: Eth1/1
#Interface descriptions append storage node management port numbers assuming all Storage Nodes' Port A -> Mellanox Switch A and all Storage Nodes' Port B -> Mellanox Switch B
#List the storage Bond1G interfaces and their description in list of dictionaries format
storage_mgmt_interfaces:
- {members: "Ethx/y", description: "HCI_Storage_Node_01"}
- {members: "Ethx/y", description: "HCI_Storage_Node_02"}
#In case of additional storage nodes, add them here
#LACP load balancing algorithm for IP hash method
#Possible options are: 'destination-mac', 'destination-ip', 'destination-port', 'source-mac', 'source-ip', 'source-port', 'source-destination-mac', 'source-destination-ip', 'source-destination-port'
#This variable takes multiple options in a single go
#For eg: if you want to configure load to be distributed in the port-channel based on the traffic source and destination IP address and port number, use 'source-destination-ip source-destination-port'
#By default, Mellanox sets it to source-destination-mac. Enter the values below only if you intend to configure any other load balancing algorithm
#Make sure the load balancing algorithm that is set here is also replicated on the host side
#Recommended algorithm is source-destination-ip source-destination-port
#Fill the lacp_load_balance variable only if you are using configuring interfaces on compute nodes in bond or LAG with LACP

```

```

lacp_load_balance: "source-destination-ip source-destination-port"
#Compute Interface details
#Members to be defined with Eth in the format. Eg: Eth1/1
#Fill the mlag_id field only if you intend to configure interfaces of
compute nodes into bond or LAG with LACP
#In case you do not intend to configure LACP on interfaces of compute
nodes, either leave the mlag_id field unfilled or comment it or enter NA
in the mlag_id field
#In case you have a mixed architecture where some compute nodes require
LACP and some don't,
#1. Fill the mlag_id field with appropriate MLAG ID for interfaces that
connect to compute nodes requiring LACP
#2. Either fill NA or leave the mlag_id field blank or comment it for
interfaces connecting to compute nodes that do not require LACP
#Only numerical digits between 100 to 1000 allowed for mlag_id.
#Operational link speed [variable 'speed' below] to be defined in terms
of bytes.
#For 10 Gigabyte operational speed, define 10G. [Possible values - 10G
and 25G]
#Interface descriptions append compute node port numbers assuming all
Compute Nodes' Port D -> Mellanox Switch A and all Compute Nodes' Port E
-> Mellanox Switch B
#List the compute interfaces, their speed, MLAG IDs and their
description in list of dictionaries format
compute_interfaces:
- members: "Eth1/7"#Compute Node for ESXi, setup by NDE
  description: "HCI_Compute_Node_01"
  mlag_id: #Fill the mlag_id only if you wish to use LACP on interfaces
  towards compute nodes
  speed: 25G
- members: "Eth1/8"#Compute Node for ESXi, setup by NDE
  description: "HCI_Compute_Node_02"
  mlag_id: #Fill the mlag_id only if you wish to use LACP on interfaces
  towards compute nodes
  speed: 25G
#In case of additional compute nodes, add them here in the same format
as above- members: "Eth1/9"#Compute Node for Kubernetes Worker node
  description: "HCI_Compute_Node_01"
  mlag_id: 109 #Fill the mlag_id only if you wish to use LACP on
  interfaces towards compute nodes
  speed: 10G
- members: "Eth1/10"#Compute Node for Kubernetes Worker node
  description: "HCI_Compute_Node_02"
  mlag_id: 110 #Fill the mlag_id only if you wish to use LACP on
  interfaces towards compute nodes
  speed: 10G

```

```

#Uplink Switch LACP support
#Possible options are 'yes' and 'no' - Set to 'yes' only if your uplink
switch supports LACP
uplink_switch_lacp: <>yes or no>>
#Uplink Interface details
#Members to be defined with Eth in the format. Eg: Eth1/1
#Only numerical digits between 100 to 1000 allowed for mlag_id.
#Operational link speed [variable 'speed' below] to be defined in terms
of bytes.
#For 10 Gigabyte operational speed, define 10G. [Possible values in
Mellanox are 1G, 10G and 25G]
#List the uplink interfaces, their description, MLAG IDs and their speed
in list of dictionaries format
uplink_interfaces:
- members: "Eth1/18"
  description_switch_a: "SwitchA:Ethx/y -> Uplink_Switch:Ethx/y"
  description_switch_b: "SwitchB:Ethx/y -> Uplink_Switch:Ethx/y"
  mlag_id: 118 #Fill the mlag_id only if 'uplink_switch_lacp' is set to
'yes'
  speed: 10G
  mtu: 1500

```



The fingerprint for the switch's key must match with that present in the host machine from where the playbook is being executed. To ensure this, add the key to `/root/.ssh/known_host` or any other appropriate location.

## Rollback the Switch Configuration

1. In case of any timeout failures or partial configuration, run the following command to roll back the switch to the initial state.

```
configuration switch-to pre-ansible
```



This operation requires a reboot of the switch.

2. Switch the configuration to the state before running the Ansible playbook.

```
configuration delete post-ansible
```

3. Delete the post-ansible file that had the configuration from the Ansible playbook.

```
configuration write to post-ansible
```

4. Create a new file with the same name post-ansible, write the pre-ansible configuration to it, and switch to the new configuration to restart configuration.

## IP Address Requirements

The deployment of the NetApp HCI inferencing platform with VMware and Kubernetes requires multiple IP addresses to be allocated. The following table lists the number of IP addresses required. Unless otherwise indicated, addresses are assigned automatically by NDE.

IP Address Quantity	Details	VLAN ID	IP Address
One per storage and compute node*	HCI terminal user interface (TUI) addresses	16	
One per vCenter Server (VM)	vCenter Server management address	3488	
One per management node (VM)	Management node IP address		
One per ESXi host	ESXi compute management addresses		
One per storage/witness node	NetApp HCI storage node management addresses		
One per storage cluster	Storage cluster management address		
One per ESXi host	VMware vMotion address	3489	
Two per ESXi host	ESXi host initiator address for iSCSI storage traffic	3490	
Two per storage node	Storage node target address for iSCSI storage traffic		
Two per storage cluster	Storage cluster target address for iSCSI storage traffic		
Two for mNode	mNode iSCSI storage access		

The following IPs are assigned manually when the respective components are configured.

IP Address Quantity	Details	VLAN ID	IP Address
One for Deployment Jump Management network	Deployment Jump VM to execute Ansible playbooks and configure other parts of the system – management connectivity	3488	
One per Kubernetes master node – management network	Kubernetes master node VMs (three nodes)	3488	

IP Address Quantity	Details	VLAN ID	IP Address
One per Kubernetes worker node – management network	Kubernetes worker nodes (two nodes)	3488	
One per Kubernetes worker node – NFS network	Kubernetes worker nodes (two nodes)	3491	
One per Kubernetes worker node – application network	Kubernetes worker nodes (two nodes)	3487	
Three for ONTAP Select – management network	ONTAP Select VM	3488	
One for ONTAP Select – NFS network	ONTAP Select VM – NFS data traffic	3491	
At least two for Triton Inference Server Load Balancer – application network	Load balancer IP range for Kubernetes load balancer service	3487	

\*This validation requires the initial setup of the first storage node TUI address. NDE automatically assigns the TUI address for subsequent nodes.

## DNS and Timekeeping Requirement

Depending on your deployment, you might need to prepare DNS records for your NetApp HCI system. NetApp HCI requires a valid NTP server for timekeeping; you can use a publicly available time server if you do not have one in your environment.

This validation involves deploying NetApp HCI with a new VMware vCenter Server instance using a fully qualified domain name (FQDN). Before deployment, you must have one Pointer (PTR) record and one Address (A) record created on the DNS server.

[Next: Virtual Infrastructure with Automated Deployment](#)

## Deploy VMware Virtual Infrastructure on NetApp HCI with NDE (Automated Deployment)

### NDE Deployment Prerequisites

Consult the [NetApp HCI Prerequisites Checklist](#) to see the requirements and recommendations for NetApp HCI before you begin deployment.

1. Network and switch requirements and configuration
2. Prepare required VLAN IDs
3. Switch configuration
4. IP Address Requirements for NetApp HCI and VMware
5. DNS and time-keeping requirements
6. Final preparations

## NDE Execution

Before you execute the NDE, you must complete the rack and stack of all components, configuration of the network switches, and verification of all prerequisites. You can execute NDE by connecting to the management address of a single storage node if you plan to allow NDE to automatically configure all addresses.

NDE performs the following tasks to bring an HCI system online:

1. Installs the storage node (NetApp Element software) on a minimum of two storage nodes.
2. Installs the VMware hypervisor on a minimum of two compute nodes.
3. Installs VMware vCenter to manage the entire NetApp HCI stack.
4. Installs and configures the NetApp storage management node (mNode) and NetApp Monitoring Agent.



This validation uses NDE to automatically configure all addresses. You can also set up DHCP in your environment or manually assign IP addresses for each storage node and compute node. These steps are not covered in this guide.

As mentioned previously, this validation uses a two-cable configuration for compute nodes.

Detailed steps for the NDE are not covered in this document.

For step-by-step guidance on completing the deployment of the base NetApp HCI platform, see the [Deployment guide](#).

5. After NDE has finished, login to the vCenter and create a Distributed Port Group [NetApp HCI VDS 01-NFS\\_Network](#) for the NFS network to be used by ONTAP Select and the application.

[Next: Configure NetApp H615c \(Manual Deployment\)](#)

### Configure NetApp H615c (Manual Deployment)

In this solution, the NetApp H615c compute nodes are configured as Kubernetes worker nodes. The Inferencing workload is hosted on these nodes.

Deploying the compute nodes involves the following tasks:

- Install Ubuntu 18.04.4 LTS.
- Configure networking for data and management access.
- Prepare the Ubuntu instances for Kubernetes deployment.

### Install Ubuntu 18.04.4 LTS

The following high-level steps are required to install the operating system on the H615c compute nodes:

1. Download Ubuntu 18.04.4 LTS from [Ubuntu releases](#).
2. Using a browser, connect to the IPMI of the H615c node and launch Remote Control.
3. Map the Ubuntu ISO using the Virtual Media Wizard and start the installation.
4. Select one of the two physical interfaces as the [Primary network interface](#) when prompted.

An IP from a DHCP source is allocated when available, or you can switch to a manual IP configuration

later. The network configuration is modified to a bond-based setup after the OS has been installed.

5. Provide a hostname followed by a domain name.
6. Create a user and provide a password.
7. Partition the disks according to your requirements.
8. Under Software Selection, select `OpenSSH server` and click Continue.
9. Reboot the node.

## Configure Networking for Data and Management Access

The two physical network interfaces of the Kubernetes worker nodes are set up as a bond and VLAN interfaces for management and application, and NFS data traffic is created on top of it.



The inferencing applications and associated containers use the application network for connectivity.

1. Connect to the console of the Ubuntu instance as a user with root privileges and launch a terminal session.
2. Navigate to `/etc/netplan` and open the `01-netcfg.yaml` file.
3. Update the netplan file based on the network details for the management, application, and NFS traffic in your environment.

The following template of the netplan file was used in this solution:

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    enp59s0f0: #Physical Interface 1
      match:
        macaddress: <<mac_address Physical Interface 1>>
      set-name: enp59s0f0
      mtu: 9000
    enp59s0f1: # Physical Interface 2
      match:
        macaddress: <<mac_address Physical Interface 2>>
      set-name: enp59s0f1
      mtu: 9000
  bonds:
    bond0:
      mtu: 9000
      dhcp4: false
      dhcp6: false
      interfaces: [ enp59s0f0, enp59s0f1 ]
      parameters:
```

```
        mode: 802.3ad
        mii-monitor-interval: 100

vlans:
    vlan.3488: #Management VLAN
        id: 3488
        xref:{relative_path}bond0
        dhcp4: false
        addresses: [ipv4_address/subnet]
        routes:
            - to: 0.0.0.0/0
                via: 172.21.232.111
                metric: 100
                table: 3488
            - to: x.x.x.x/x # Additional routes if any
                via: y.y.y.y
                metric: <<metric>>
                table: <<table #>>
        routing-policy:
            - from: 0.0.0.0/0
                priority: 32768#Higher Priority than table 3487
                table: 3488
    nameservers:
        addresses: [nameserver_ip]
        search: [ search_domain ]
    mtu: 1500
vlan.3487:
    id: 3487
    xref:{relative_path}bond0
    dhcp4: false
    addresses: [ipv4_address/subnet]
    routes:
        - to: 0.0.0.0/0
            via: 172.21.231.111
            metric: 101
            table: 3487
        - to: x.x.x.x/x
            via: y.y.y.y
            metric: <<metric>>
            table: <<table #>>
    routing-policy:
        - from: 0.0.0.0/0
            priority: 32769#Lower Priority
            table: 3487
    nameservers:
        addresses: [nameserver_ip]
        search: [ search_domain ]
```

```
mtu: 1500      wlan.3491:  
id: 3491  
xref:{relative_path}bond0  
dhcp4: false  
addresses: [ipv4_address/subnet]  
mtu: 9000
```

4. Confirm that the priorities for the routing policies are lower than the priorities for the main and default tables.
5. Apply the netplan.

```
sudo netplan --debug apply
```

6. Make sure that there are no errors.
7. If Network Manager is running, stop and disable it.

```
systemctl stop NetworkManager  
systemctl disable NetworkManager
```

8. Add a host record for the server in DNS.
9. Open a VI editor to [/etc/iproute2/rt\\_tables](#) and add the two entries.

```
#  
# reserved values  
#  
255      local  
254      main  
253      default  
0        unspec  
#  
# local  
#  
#1      inr.ruhel  
101     3488  
102     3487
```

10. Match the table number to what you used in the netplan.
11. Open a VI editor to [/etc/sysctl.conf](#) and set the value of the following parameters.

```
net.ipv4.conf.default.rp_filter=0  
net.ipv4.conf.all.rp_filter=0net.ipv4.ip_forward=1
```

12. Update the system.

```
sudo apt-get update && sudo apt-get upgrade
```

13. Reboot the system

14. Repeat steps 1 through 13 for the other Ubuntu instance.

[Next: Set Up the Deployment Jump and the Kubernetes Master Node VMs \(Manual Deployment\)](#)

#### **Set Up the Deployment Jump VM and the Kubernetes Master Node VMs (Manual Deployment)**

A Deployment Jump VM running a Linux distribution is used for the following purposes:

- Deploying ONTAP Select using an Ansible playbook
- Deploying the Kubernetes infrastructure with NVIDIA DeepOps and GPU Operator
- Installing and configuring NetApp Trident

Three more VMs running Linux are set up; these VMs are configured as Kubernetes Master Nodes in this solution.

Ubuntu 18.04.4 LTS was used in this solution deployment.

1. Deploy the Ubuntu 18.04.4 LTS VM with VMware tools

You can refer to the high-level steps described in section [Install Ubuntu 18.04.4 LTS](#).

2. Configure the in-band management network for the VM. See the following sample netplan template:

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).

network:
    version: 2
    renderer: networkd
    ethernets:
        ens160:
            dhcp4: false
            addresses: [ipv4_address/subnet]
            routes:
                - to: 0.0.0.0/0
                  via: 172.21.232.111
                  metric: 100
                  table: 3488
            routing-policy:
                - from: 0.0.0.0/0
                  priority: 32768
                  table: 3488
            nameservers:
                addresses: [nameserver_ip]
                search: [ search_domain ]
            mtu: 1500
```

This template is not the only way to setup the network. You can use any other approach that you prefer.

### 3. Apply the netplan.

```
sudo netplan --debug apply
```

### 4. Stop and disable Network Manager if it is running.

```
systemctl stop NetworkManager
systemctl disable NetworkManager
```

### 5. Open a VI editor to [/etc/iproute2/rt\\_tables](#) and add a table entry.

```
#  
# reserved values  
#  
255      local  
254      main  
253      default  
0        unspec  
#  
# local  
#  
#1      inr.ruhep  
101     3488
```

6. Add a host record for the VM in DNS.
7. Verify outbound internet access.
8. Update the system.

```
sudo apt-get update && sudo apt-get upgrade
```

9. Reboot the system.
10. Repeat steps 1 through 9 to set up the other three VMs.

[Next: Deploy a Kubernetes Cluster with NVIDIA DeepOps \(Automated Deployment\)](#)

#### Deploy a Kubernetes Cluster with NVIDIA DeepOps Automated Deployment

To deploy and configure the Kubernetes Cluster with NVIDIA DeepOps, complete the following steps:

1. Make sure that the same user account is present on all the Kubernetes master and worker nodes.
2. Clone the DeepOps repository.

```
git clone https://github.com/NVIDIA/deepops.git
```

3. Check out a recent release tag.

```
cd deepops  
git checkout tags/20.08
```

If this step is skipped, the latest development code is used, not an official release.

4. Prepare the Deployment Jump by installing the necessary prerequisites.

```
./scripts/setup.sh
```

5. Create and edit the Ansible inventory by opening a VI editor to [deepops/config/inventory](#).
  - a. List all the master and worker nodes under [all].
  - b. List all the master nodes under [kube-master]
  - c. List all the master nodes under [etcd]
  - d. List all the worker nodes under [kube-node]

```
#####
# ALL NODES
# NOTE: Use existing hostnames here, DeepOps will config
#####
[all]
hci-ai-k8-master-01      ansible_host=172.21.232.114
hci-ai-k8-master-02      ansible_host=172.21.232.115
hci-ai-k8-master-03      ansible_host=172.21.232.116
hci-ai-k8-worker-01      ansible_host=172.21.232.109
hci-ai-k8-worker-02      ansible_host=172.21.232.110

#####
# KUBERNETES
#####
[kube-master]
hci-ai-k8-master-01
hci-ai-k8-master-02
hci-ai-k8-master-03

# Odd number of nodes required
[etcd]
hci-ai-k8-master-01
hci-ai-k8-master-02
hci-ai-k8-master-03

# Also add mgmt/master nodes here if they will run non
[kube-node]
hci-ai-k8-worker-01
hci-ai-k8-worker-02

[k8s-cluster:children]
kube-master
kube-node
```

6. Enable GPUOperator by opening a VI editor to [deepops/config/group\\_vars/k8s-cluster.yml](#).

```
# Provide option to use GPU Operator instead of setting up NVIDIA driver and
# Docker configuration.
deepops_gpu_operator_enabled: true
```

7. Set the value of `deepops_gpu_operator_enabled` to true.

8. Verify the permissions and network configuration.

```
ansible all -m raw -a "hostname" -k -K
```

- If SSH to the remote hosts requires a password, use -k.
- If sudo on the remote hosts requires a password, use -K.

9. If the previous step passed without any issues, proceed with the setup of Kubernetes.

```
ansible-playbook --limit k8s-cluster playbooks/k8s-cluster.yml -k -K
```

10. To verify the status of the Kubernetes nodes and the pods, run the following commands:

```
kubectl get nodes
```

```
rarvind@deployment-jump:~/deepops$ kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
hci-ai-k8-master-01  Ready  master  2d19h  v1.17.6
hci-ai-k8-master-02  Ready  master  2d19h  v1.17.6
hci-ai-k8-master-03  Ready  master  2d19h  v1.17.6
hci-ai-k8-worker-01  Ready  <none>  2d19h  v1.17.6
hci-ai-k8-worker-02  Ready  <none>  2d19h  v1.17.6
```

```
kubectl get pods -A
```

It can take a few minutes for all the pods to run.

NAMESPACE	NAME	READY	STATUS
default	gpu-operator-74c97448d9-ppdlc	1/1	Running
default	nvidia-gpu-operator-node-feature-discovery-master-ffccb57dx9wtl	1/1	Running
default	nvidia-gpu-operator-node-feature-discovery-worker-2lr9t	1/1	Running
default	nvidia-gpu-operator-node-feature-discovery-worker-616x7	1/1	Running
default	nvidia-gpu-operator-node-feature-discovery-worker-jf696	1/1	Running
default	nvidia-gpu-operator-node-feature-discovery-worker-tmtwv	1/1	Running
default	nvidia-gpu-operator-node-feature-discovery-worker-z4nlh	1/1	Running
gpu-operator-resources	nvidia-container-toolkit-daemonset-7jbl4	1/1	Running
gpu-operator-resources	nvidia-container-toolkit-daemonset-x5ktb	1/1	Running
gpu-operator-resources	nvidia-dcgm-exporter-5x94p	1/1	Running
gpu-operator-resources	nvidia-dcgm-exporter-7cb1	1/1	Running
gpu-operator-resources	nvidia-device-plugin-daemonset-n8vrk	1/1	Running
gpu-operator-resources	nvidia-device-plugin-daemonset-z7j6s	1/1	Running
gpu-operator-resources	nvidia-device-plugin-validation	0/1	Completed
gpu-operator-resources	nvidia-driver-daemonset-7h752	1/1	Running
gpu-operator-resources	nvidia-driver-daemonset-v4rbj	1/1	Running
gpu-operator-resources	nvidia-driver-validation	0/1	Completed
kube-system	calico-kube-controllers-777478f4ff-jknxg	1/1	Running
kube-system	calico-node-2j9mr	1/1	Running
kube-system	calico-node-czk76	1/1	Running
kube-system	calico-node-jpdxn	1/1	Running
kube-system	calico-node-nwnvn	1/1	Running
kube-system	calico-node-ssjrx	1/1	Running
kube-system	coredns-76798d84dd-5pvgf	1/1	Running
kube-system	coredns-76798d84dd-w7l2j	1/1	Running
kube-system	dns-autoscaler-85f898cd5c-qqrpb	1/1	Running
kube-system	kube-apiserver-hci-ai-k8-master-01	1/1	Running
kube-system	kube-apiserver-hci-ai-k8-master-02	1/1	Running
kube-system	kube-apiserver-hci-ai-k8-master-03	1/1	Running
kube-system	kube-controller-manager-hci-ai-k8-master-01	1/1	Running
kube-system	kube-controller-manager-hci-ai-k8-master-02	1/1	Running
kube-system	kube-controller-manager-hci-ai-k8-master-03	1/1	Running
kube-system	kube-proxy-5znxk	1/1	Running
kube-system	kube-proxy-fk6h6	1/1	Running
kube-system	kube-proxy-hphfb	1/1	Running
kube-system	kube-proxy-qzxhr	1/1	Running
kube-system	kube-proxy-rkjds	1/1	Running
kube-system	kube-scheduler-hci-ai-k8-master-01	1/1	Running
kube-system	kube-scheduler-hci-ai-k8-master-02	1/1	Running
kube-system	kube-scheduler-hci-ai-k8-master-03	1/1	Running
kube-system	kubernetes-dashboard-5fcff756f-dmswt	1/1	Running
kube-system	kubernetes-metrics-scrapers-747b4fd5cd-4q4p2	1/1	Running
kube-system	nginx-proxy-hci-ai-k8-worker-01	1/1	Running
kube-system	nginx-proxy-hci-ai-k8-worker-02	1/1	Running
kube-system	nodelocaldns-2dmjr	1/1	Running
kube-system	nodelocaldns-b7xrw	1/1	Running
kube-system	nodelocaldns-jrhs2	1/1	Running
kube-system	nodelocaldns-jztzs	1/1	Running
kube-system	nodelocaldns-wgx84	1/1	Running

11. Verify that the Kubernetes setup can access and use the GPUs.

```
./scripts/k8s_verify_gpu.sh
```

Expected sample output:

```
rarvind@deployment-jump:~/deepops$ ./scripts/k8s_verify_gpu.sh
job_name=cluster-gpu-tests
Node found with 3 GPUs
Node found with 3 GPUs
total_gpus=6
Creating/Deleting sandbox Namespace
updating test yaml
downloading containers ...
```

```
job.batch/cluster-gpu-tests condition met
executing ...
Mon Aug 17 16:02:45 2020
+-----+
-----+
| NVIDIA-SMI 440.64.00      Driver Version: 440.64.00      CUDA Version:
10.2      |
|-----+-----+
+-----+
| GPU  Name      Persistence-M| Bus-Id      Disp.A | Volatile
Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|           Memory-Usage | GPU-Util
Compute M. |
|=====+=====+=====+=====+=====+=====+=====+
=====|
|     0  Tesla T4            On      | 00000000:18:00.0 Off  |
0 |
| N/A   38C    P8    10W /  70W |      0MiB / 15109MiB |      0%
Default |
+-----+-----+
+-----+
-----+
| Processes:                               GPU
Memory |
| GPU        PID  Type  Process name          Usage
|
|=====+=====+=====+=====+=====+
=====|
| No running processes found
|
+-----+
-----+
Mon Aug 17 16:02:45 2020
+-----+
-----+
| NVIDIA-SMI 440.64.00      Driver Version: 440.64.00      CUDA Version:
10.2      |
|-----+-----+
+-----+
| GPU  Name      Persistence-M| Bus-Id      Disp.A | Volatile
Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|           Memory-Usage | GPU-Util
Compute M. |
|=====+=====+=====+=====+=====+=====+=====+
=====|
```

```
| 0 Tesla T4          On | 00000000:18:00.0 Off |
0 |
| N/A 38C   P8    10W / 70W |      0MiB / 15109MiB |      0%
Default |
+-----+
+-----+
+-----+
| Processes:                                     GPU
Memory |
| GPU      PID  Type  Process name             Usage
|
| =====
===== |
| No running processes found
|
+-----+
+-----+
Mon Aug 17 16:02:45 2020
+-----+
+-----+
| NVIDIA-SMI 440.64.00     Driver Version: 440.64.00     CUDA Version:
10.2      |
|-----+-----+
+-----+
| GPU  Name      Persistence-M| Bus-Id      Disp.A | Volatile
Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|           Memory-Usage | GPU-Util
Compute M. |
|-----+-----+-----+-----+
===== |
| 0 Tesla T4          On | 00000000:18:00.0 Off |
0 |
| N/A 38C   P8    10W / 70W |      0MiB / 15109MiB |      0%
Default |
+-----+
+-----+
+-----+
| Processes:                                     GPU
Memory |
| GPU      PID  Type  Process name             Usage
|
| =====
===== |
| No running processes found
```

```
|  
+-----  
----+  
Mon Aug 17 16:02:45 2020  
+-----  
----+  
| NVIDIA-SMI 440.64.00     Driver Version: 440.64.00     CUDA Version:  
10.2      |  
|-----+-----+-----+  
+-----+  
| GPU Name          Persistence-M| Bus-Id          Disp.A | Volatile  
Uncorr. ECC |  
| Fan Temp Perf Pwr:Usage/Cap|     Memory-Usage | GPU-Util  
Compute M. |  
|=====+=====+=====+=====+=====+=====+  
=====|  
| 0 Tesla T4          On    | 00000000:18:00.0 Off |  
0 |  
| N/A 38C   P8    10W / 70W | 0MiB / 15109MiB | 0%  
Default |  
+-----+-----+  
+-----+  
+-----+  
----+  
| Processes:                      GPU  
Memory |  
| GPU     PID  Type  Process name           Usage  
|  
|=====+=====+=====+=====+  
=====|  
| No running processes found  
|  
+-----+  
----+  
Mon Aug 17 16:02:45 2020  
+-----  
----+  
| NVIDIA-SMI 440.64.00     Driver Version: 440.64.00     CUDA Version:  
10.2      |  
|-----+-----+-----+  
+-----+  
| GPU Name          Persistence-M| Bus-Id          Disp.A | Volatile  
Uncorr. ECC |  
| Fan Temp Perf Pwr:Usage/Cap|     Memory-Usage | GPU-Util  
Compute M. |  
|=====+=====+=====+=====+=====+=====+
```

```
=====|  
| 0 Tesla T4          On | 00000000:18:00.0 Off |  
0 |  
| N/A 38C   P8    10W / 70W |      0MiB / 15109MiB |      0%  
Default |  
+-----+  
+-----+  
+-----+  
| Processes:                                     GPU  
Memory |  
| GPU     PID  Type  Process name             Usage  
|  
|=====|  
=====|  
| No running processes found  
|  
+-----+  
+-----+  
Mon Aug 17 16:02:45 2020  
+-----+  
+-----+  
| NVIDIA-SMI 440.64.00     Driver Version: 440.64.00     CUDA Version:  
10.2      |  
|-----+  
+-----+  
| GPU  Name          Persistence-M| Bus-Id          Disp.A | Volatile  
Uncorr. ECC |  
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  
Compute M. |  
|=====+=====+=====+=====+  
=====|  
| 0 Tesla T4          On | 00000000:18:00.0 Off |  
0 |  
| N/A 38C   P8    10W / 70W |      0MiB / 15109MiB |      0%  
Default |  
+-----+  
+-----+  
+-----+  
| Processes:                                     GPU  
Memory |  
| GPU     PID  Type  Process name             Usage  
|  
|=====|  
=====|
```

```
| No running processes found
|
+-----+
-----+
Number of Nodes: 2
Number of GPUs: 6
6 / 6 GPU Jobs COMPLETED
job.batch "cluster-gpu-tests" deleted
namespace "cluster-gpu-verify" deleted
```

## 12. Install Helm on the Deployment Jump.

```
./scripts/install_helm.sh
```

## 13. Remove the taints on the master nodes.

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```

This step is required to run the LoadBalancer pods.

## 14. Deploy LoadBalancer.

## 15. Edit the `config/helm/metallb.yml` file and provide a range of IP addresses in the [Application Network](#) to be used as LoadBalancer.

```
---
# Default address range matches private network for the virtual cluster
# defined in virtual/.
# You should set this address range based on your site's infrastructure.
configInline:
  address-pools:
    - name: default
      protocol: layer2
      addresses:
        - 172.21.231.130-172.21.231.140#Application Network
controller:
  nodeSelector:
    node-role.kubernetes.io/master: ""
```

## 16. Run a script to deploy LoadBalancer.

```
./scripts/k8s_deploy_loadbalancer.sh
```

17. Deploy an Ingress Controller.

```
./scripts/k8s_deploy_ingress.sh
```

Next: Deploy and Configure ONTAP Select in the VMware Virtual Infrastructure (Automated Deployment)

**Deploy and Configure ONTAP Select in the VMware Virtual Infrastructure (Automated Deployment)**

To deploy and configure an ONTAP Select instance within the VMware Virtual Infrastructure, complete the following steps:

1. From the Deployment Jump VM, login to the [NetApp Support Site](#) and download the ONTAP Select OVA for ESXi.
2. Create a directory OTS and obtain the Ansible roles for deploying ONTAP Select.

```
mkdir OTS
cd OTS
git clone https://github.com/NetApp/ansible.git
cd ansible
```

3. Install the prerequisite libraries.

```
pip install requests
pip install pyvmomi
Open a VI Editor and create a playbook ``ots_setup.yaml`` with the below
content to deploy the ONTAP Select OVA and initialize the ONTAP cluster.
---
- name: Create ONTAP Select Deploy VM from OVA (ESXi)
  hosts: localhost
  gather_facts: false
  connection: 'local'
  vars_files:
    - ots_deploy_vars.yaml
  roles:
    - na_ots_deploy
- name: Wait for 1 minute before starting cluster setup
  hosts: localhost
  gather_facts: false
  tasks:
    - pause:
        minutes: 1
- name: Create ONTAP Select cluster (ESXi)
  hosts: localhost
  gather_facts: false
  vars_files:
    - ots_cluster_vars.yaml
  roles:
    - na_ots_cluster
```

4. Open a VI editor, create a variable file `ots_deploy_vars.yaml`, and fill in hte following parameters:

```
target_vcenter_or_esxi_host: "10.xxx.xx.xx"# vCenter IP
host_login: "yourlogin@yourlab.local" # vCenter Username
ovf_path: "/run/deploy/ovapath/ONTAPdeploy.ova"# Path to OVA on
Deployment Jump VM
datacenter_name: "your-Lab"# Datacenter name in vCenter
esx_cluster_name: "your Cluster"# Cluster name in vCenter
datastore_name: "your-select-dt"# Datastore name in vCenter
mgt_network: "your-mgmt-network"# Management Network to be used by OVA
deploy_name: "test-deploy-vm"# Name of the ONTAP Select VM
deploy_ipAddress: "10.xxx.xx.xx"# Management IP Address of ONTAP Select
VM
deploy_gateway: "10.xxx.xx.1"# Default Gateway
deploy_proxy_url: ""# Proxy URL (Optional and if used)
deploy_netMask: "255.255.255.0"# Netmask
deploy_product_company: "NetApp"# Name of Organization
deploy_primaryDNS: "10.xxx.xx.xx"# Primary DNS IP
deploy_secondaryDNS: ""# Secondary DNS (Optional)
deploy_searchDomains: "your.search.domain.com"# Search Domain Name
```

Update the variables to match your environment.

5. Open a VI editor, create a variable file `ots_cluster_vars.yaml`, and fill it out with the following parameters:

```

node_count: 1#Number of nodes in the ONTAP Cluster
monitor_job: truemonitor_deploy_job: true
deploy_api_url: #Use the IP of the ONTAP Select VM
deploy_login: "admin"
vcenter_login: "administrator@vsphere.local"
vcenter_name: "172.21.232.100"
esxi_hosts:
  - host_name: 172.21.232.102
  - host_name: 172.21.232.103
cluster_name: "hci-ai-ots"# Name of ONTAP Cluster
cluster_ip: "172.21.232.118"# Cluster Management IP
cluster_netmask: "255.255.255.0"
cluster_gateway: "172.21.232.1"
cluster_ontap_image: "9.7"
cluster_ntp:
  - "10.61.186.231"
cluster_dns_ips:
  - "10.61.186.231"
cluster_dns_domains:
  - "sddc.netapp.com"
mgt_network: "NetApp HCI VDS 01-Management_Network"# Name of VM Port
Group for Mgmt Network
data_network: "NetApp HCI VDS 01-NFS_Network"# Name of VM Port Group for
NFS Network
internal_network: ""# Not needed for Single Node Cluster
instance_type: "small"
cluster_nodes:
  - node_name: "{{ cluster_name }}-01"
    ipAddress: 172.21.232.119# Node Management IP
    storage_pool: NetApp-HCI-Datastore-02 # Name of Datastore in vCenter
    to use
    capacityTB: 1# Usable capacity will be ~700GB
    host_name: 172.21.232.102# IP Address of an ESXi host to deploy node

```

Update the variables to match your environment.

## 6. Start ONTAP Select setup.

```

ansible-playbook ots_setup.yaml --extra-vars deploy_pwd=$'"P@ssw0rd"''
--extra-vars vcenter_password=$'"P@ssw0rd"' --extra-vars
ontap_pwd=$'"P@ssw0rd"' --extra-vars host_esx_password=$'"P@ssw0rd"''
--extra-vars host_password=$'"P@ssw0rd"' --extra-vars
deploy_password=$'"P@ssw0rd"'

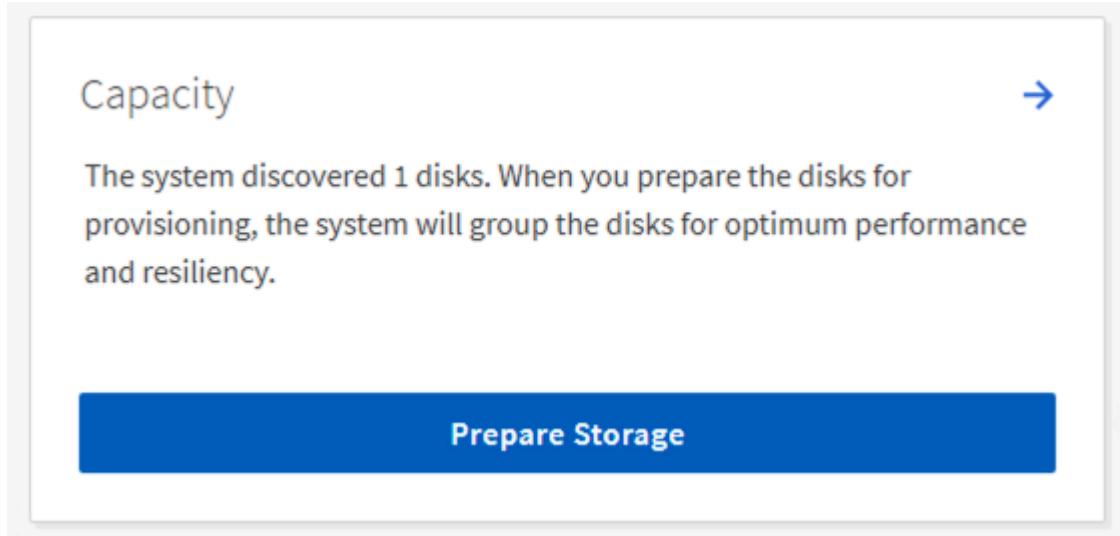
```

7. Update the command with `deploy_pwd` (ONTAP Select VM instance), `\vcenter_password`(vCenter), `ontap_pwd` (ONTAP login password), `host_esx_password` (VMware ESXi), `host_password` (vCenter), and `deploy_password` (ONTAP Select VM instance).

## Configure the ONTAP Select Cluster – Manual Deployment

To configure the ONTAP Select cluster, complete the following steps:

1. Open a browser and log into the ONTAP cluster's System Manager using its cluster management IP.
2. On the DASHBOARD page, click Prepare Storage under Capacity.



3. Select the radio button to continue without onboard key manager, and click Prepare Storage.
4. On the NETWORK page, click the + sign in the Broadcast Domains window.

Broadcast Domains		
Cluster	9000 MTU	IPspace: Cluster
Default	1500 MTU	IPspace: Default hci-ai-ots-01 e0b e0c
Mgmt	1500 MTU	IPspace: Default hci-ai-ots-01 e0a

5. Enter the Name as `NFS`, set the MTU to `9000`, and select the port `e0b`. Click Save.

# Add Broadcast Domain

Specify the following details to add a new broadcast domain.

NAME

NFS

MTU

9000

ASSIGN PORTS [?](#)

Port Name	hci-ai-ots-01
e0b	<input checked="" type="checkbox"/>
e0c	<input type="checkbox"/>

**Save**

[Cancel](#)

6. On the DASHBOARD page, click [Configure Protocols](#) under Network.

## Network

No protocols are enabled. To begin serving data to clients, enable the required protocols and assign the protocol addresses.

[Configure Protocols](#)

7. Enter a name for the SVM, select Enable NFS, provide an IP and subnet mask for the NFS LIF, set the Broadcast Domain to NFS, and click Save.

## Configure Protocols

X

ONTAP exposes protocol services through storage VMs. [More details](#)

STORAGE VM NAME

infra-NFS-hci-ai

### Access Protocol

SMB/CIFS and NFS

iSCSI

Enable SMB/CIFS

Enable NFS

DEFAULT LANGUAGE [?](#)

c.utf\_8

### NETWORK INTERFACE

One network interface per node is recommended.

hci-ai-ots-01

IP ADDRESS

172.21.235.119

SUBNET MASK

255.255.255.0

GATEWAY

Add optional gateway

BROADCAST DOMAIN

NFS

**Save**

[Cancel](#)

8. Click STORAGE in the left pane, and from the dropdown select Storage VMs

- a. Edit the SVM.

## Storage VMs

+ Add

Name	State
infra-NFS-hci-ai	running

⋮

[Edit](#)

[Delete](#)

[Stop](#)

- b. Select the checkbox under Resource Allocation, make sure that the local tier is listed, and click Save.

## Edit Storage VM

X

STORAGE VM NAME

infra-NFS-hci-ai

DEFAULT LANGUAGE

c.utf\_8



## Resource Allocation

- Limit volume creation to preferred local tiers

LOCAL TIERS

hci\_ai\_ots\_01\_SSD\_1 X

Cancel

Save

9. Click the SVM name, and on the right panel scroll down to Policies.
10. Click the arrow within the Export Policies tile, and click the default policy.
11. If there is a rule already defined, you can edit it; if no rule exists, then create a new one.
  - a. Select NFS Network Clients as the Client Specification.
  - b. Select the Read-Only and Read/Write checkboxes.
  - c. Select the checkbox to Allow Superuser Access.

## New Rule

**CLIENT SPECIFICATION**

172.21.235.0/24

**ACCESS PROTOCOLS**

SMB/CIFS  
 FlexCache  
 NFS    NFSv3    NFSv4

**ACCESS DETAILS**

Type	<input checked="" type="checkbox"/> Read-Only	<input checked="" type="checkbox"/> Read/Write
UNIX	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kerberos 5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kerberos 5i	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kerberos 5p	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NTLM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Allow Superuser Access

[Cancel](#)
**Save**

[Next: Deploy NetApp Trident \(Automated Deployment\)](#)

### Deploy NetApp Trident (Automated Deployment)

NetApp Trident is deployed by using an Ansible playbook that is available with NVIDIA DeepOps. Follow these steps to set up NetApp Trident:

1. From the Deployment Jump VM, navigate to the DeepOps directory and open a VI editor to `config/group_vars/netapp-trident.yml`. The file from DeepOps lists two backends and two storage classes. In this solution only one backend and storage class are used.

Use the following template to update the file and its parameters (highlighted in yellow) to match your environment.

```
--  
# vars file for netapp-trident playbook  
# URL of the Trident installer package that you wish to download and use  
trident_version: "20.07.0"># Version of Trident desired  
trident_installer_url:  
"https://github.com/NetApp/trident/releases/download/v{{ trident_version }}/trident-installer-{{ trident_version }}.tar.gz"  
# Kubernetes version  
# Note: Do not include patch version, e.g. provide value of 1.16, not  
1.16.7.  
# Note: Versions 1.14 and above are supported when deploying Trident  
with DeepOps.  
# If you are using an earlier version, you must deploy Trident  
manually.  
k8s_version: 1.17.9# Version of Kubernetes running  
# Denotes whether or not to create new backends after deploying trident  
# For more info, refer to: https://netapp-  
trident.readthedocs.io/en/stable-v20.04/kubernetes/operator-  
install.html#creating-a-trident-backend  
create_backends: true  
# List of backends to create  
# For more info on parameter values, refer to: https://netapp-  
trident.readthedocs.io/en/stable-  
v20.04/kubernetes/operations/tasks/backends/ontap.html  
# Note: Parameters other than those listed below are not available when  
creating a backend via DeepOps  
# If you wish to use other parameter values, you must create your  
backend manually.  
backends_to_create:  
  - backendName: ontap-flexvol  
    storageDriverName: ontap-nas # only 'ontap-nas' and 'ontap-nas-  
flexgroup' are supported when creating a backend via DeepOps  
    managementLIF: 172.21.232.118# Cluster Management IP or SVM Mgmt LIF  
IP  
    dataLIF: 172.21.235.119# NFS LIF IP  
    svm: infra-NFS-hci-ai# Name of SVM  
    username: admin# Username to connect to the ONTAP cluster  
    password: P@ssw0rd# Password to login  
    storagePrefix: trident  
    limitAggregateUsage: ""  
    limitVolumeSize: ""  
    nfsMountOptions: ""  
    defaults:  
      spaceReserve: none  
      snapshotPolicy: none  
      snapshotReserve: 0
```

```

splitOnClone: false
encryption: false
unixPermissions: 777
snapshotDir: false
exportPolicy: default
securityStyle: unix
tieringPolicy: none
# Add additional backends as needed
# Denotes whether or not to create new StorageClasses for your NetApp
storage
# For more info, refer to: https://netapp-
trident.readthedocs.io/en/stable-v20.04/kubernetes/operator-
install.html#creating-a-storage-class
create_StorageClasses: true
# List of StorageClasses to create
# Note: Each item in the list should be an actual K8s StorageClass
definition in yaml format
# For more info on StorageClass definitions, refer to https://netapp-
trident.readthedocs.io/en/stable-
v20.04/kubernetes/concepts/objects.html#kubernetes-storageclass-objects.
storageClasses_to_create:
- apiVersion: storage.k8s.io/v1
  kind: StorageClass
  metadata:
    name: ontap-flexvol
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
  provisioner: csi.trident.netapp.io
  parameters:
    backendType: "ontap-nas"
# Add additional StorageClasses as needed
# Denotes whether or not to copy tridentctl binary to localhost
copy_tridentctl_to_localhost: true
# Directory that tridentctl will be copied to on localhost
tridentctl_copy_to_directory: ../ # will be copied to 'deepops/'
directory

```

## 2. Setup NetApp Trident by using the Ansible playbook.

```
ansible-playbook -l k8s-cluster playbooks/netapp-trident.yml
```

## 3. Verify that Trident is running.

```
./tridentctl -n trident version
```

The expected output is as follows:

```
rarvind@deployment-jump:~/deepops$ ./tridentctl -n trident version
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 20.07.0         | 20.07.0          |
+-----+-----+
```

[Next: Deploy NVIDIA Triton Inference Server \(Automated Deployment\)](#)

#### Deploy NVIDIA Triton Inference Server (Automated Deployment)

To set up automated deployment for the Triton Inference Server, complete the following steps:

1. Open a VI editor and create a PVC yaml file `vi pvc-triton-model-repo.yaml`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: triton-pvc  namespace: triton
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: ontap-flexvol
```

2. Create the PVC.

```
kubectl create -f pvc-triton-model-repo.yaml
```

3. Open a VI editor, create a deployment for the Triton Inference Server, and call the file `triton_deployment.yaml`.

```
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: triton-3gpu
    name: triton-3gpu
    namespace: triton
```

```
spec:
  ports:
    - name: grpc-trtis-serving
      port: 8001
      targetPort: 8001
    - name: http-trtis-serving
      port: 8000
      targetPort: 8000
    - name: prometheus-metrics
      port: 8002
      targetPort: 8002
  selector:
    app: triton-3gpu
  type: LoadBalancer
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: triton-1gpu
  name: triton-1gpu
  namespace: triton
spec:
  ports:
    - name: grpc-trtis-serving
      port: 8001
      targetPort: 8001
    - name: http-trtis-serving
      port: 8000
      targetPort: 8000
    - name: prometheus-metrics
      port: 8002
      targetPort: 8002
  selector:
    app: triton-1gpu
  type: LoadBalancer
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: triton-3gpu
  name: triton-3gpu
  namespace: triton
spec:
  replicas: 1
```

```
selector:
  matchLabels:
    app: triton-3gpu      version: v1
template:
  metadata:
    labels:
      app: triton-3gpu
      version: v1
spec:
  containers:
    - image: nvcr.io/nvidia/tritonserver:20.07-v1-py3
      command: ["/bin/sh", "-c"]
      args: ["trtserver --model-store=/mnt/model-repo"]
      imagePullPolicy: IfNotPresent
      name: triton-3gpu
      ports:
        - containerPort: 8000
        - containerPort: 8001
        - containerPort: 8002
      resources:
        limits:
          cpu: "2"
          memory: 4Gi
          nvidia.com/gpu: 3
        requests:
          cpu: "2"
          memory: 4Gi
          nvidia.com/gpu: 3
      volumeMounts:
        - name: triton-model-repo
          mountPath: /mnt/model-repo      nodeSelector:
            gpu-count: "3"
      volumes:
        - name: triton-model-repo
          persistentVolumeClaim:
            claimName: triton-pvc---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: triton-1gpu
  name: triton-1gpu
  namespace: triton
spec:
  replicas: 3
  selector:
```

```

matchLabels:
  app: triton-1gpu
  version: v1
template:
  metadata:
    labels:
      app: triton-1gpu
      version: v1
spec:
  containers:
    - image: nvcr.io/nvidia/tritonserver:20.07-v1-py3
      command: ["/bin/sh", "-c", "sleep 1000"]
      args: ["trtserver --model-store=/mnt/model-repo"]
      imagePullPolicy: IfNotPresent
      name: triton-1gpu
      ports:
        - containerPort: 8000
        - containerPort: 8001
        - containerPort: 8002
      resources:
        limits:
          cpu: "2"
          memory: 4Gi
          nvidia.com/gpu: 1
        requests:
          cpu: "2"
          memory: 4Gi
          nvidia.com/gpu: 1
      volumeMounts:
        - name: triton-model-repo
          mountPath: /mnt/model-repo
          nodeSelector:
            gpu-count: "1"
      volumes:
        - name: triton-model-repo
          persistentVolumeClaim:
            claimName: triton-pvc

```

Two deployments are created here as an example. The first deployment spins up a pod that uses three GPUs and has replicas set to 1. The other deployment spins up three pods each using one GPU while the replica is set to 3. Depending on your requirements, you can change the GPU allocation and replica counts.

Both of the deployments use the PVC created earlier and this persistent storage is provided to the Triton inference servers as the model repository.

For each deployment, a service of type LoadBalancer is created. The Triton Inference Server can be accessed by using the LoadBalancer IP which is in the application network.

A nodeSelector is used to ensure that both deployments get the required number of GPUs without any issues.

4. Label the K8 worker nodes.

```
kubectl label nodes hci-ai-k8-worker-01 gpu-count=3  
kubectl label nodes hci-ai-k8-worker-02 gpu-count=1
```

5. Create the deployment.

```
kubectl apply -f triton_deployment.yaml
```

6. Make a note of the LoadBalancer service external LPS.

```
kubectl get services -n triton
```

The expected sample output is as follows:

```
rarvind@deployment-jump:~/triton-inference-server$ kubectl get services -n triton  
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP      PORT(S)          AGE  
triton-1gpu-v20-07-v1   LoadBalancer   10.233.21.185  172.21.231.133  8001:31238/TCP,8000:30171/TCP,8002:32348/TCP  10h  
triton-3gpu-v20-07-v1   LoadBalancer   10.233.13.17   172.21.231.132  8001:31549/TCP,8000:30220/TCP,8002:31517/TCP  10h
```

7. Connect to any one of the pods that were created from the deployment.

```
kubectl exec -n triton --stdin --tty triton-1gpu-86c4c8dd64-545lx --  
/bin/bash
```

8. Set up the model repository by using the example model repository.

```
git clone  
cd triton-inference-server  
git checkout r20.07
```

9. Fetch any missing model definition files.

```
cd docs/examples  
.fetch_models.sh
```

10. Copy all the models to the model repository location or just a specific model that you wish to use.

```
cp -r model_repository/resnet50_netdef/ /mnt/model-repo/
```

In this solution, only the resnet50\_netdef model is copied over to the model repository as an example.

## 11. Check the status of the Triton Inference Server.

```
curl -v <<LoadBalancer_IP_recorded earlier>>:8000/api/status
```

The expected sample output is as follows:

```
curl -v 172.21.231.132:8000/api/status
*   Trying 172.21.231.132...
* TCP_NODELAY set
* Connected to 172.21.231.132 (172.21.231.132) port 8000 (#0)
> GET /api/status HTTP/1.1
> Host: 172.21.231.132:8000
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 200 OK
< NV-Status: code: SUCCESS server_id: "inference:0" request_id: 9
< Content-Length: 1124
< Content-Type: text/plain
<
id: "inference:0"
version: "1.15.0"
uptime_ns: 377890294368
model_status {
    key: "resnet50_netdef"
    value {
        config {
            name: "resnet50_netdef"
            platform: "caffe2_netdef"
            version_policy {
                latest {
                    num_versions: 1
                }
            }
            max_batch_size: 128
            input {
                name: "gpu_0/data"
                data_type: TYPE_FP32
                format: FORMAT_NCHW
                dims: 3
                dims: 224
                dims: 224
            }
        }
    }
}
```

```

output {
    name: "gpu_0/softmax"
    data_type: TYPE_FP32
    dims: 1000
    label_filename: "resnet50_labels.txt"
}
instance_group {
    name: "resnet50_netdef"
    count: 1
    gpus: 0
    gpus: 1
    gpus: 2
    kind: KIND_GPU
}
default_model_filename: "model.netdef"
optimization {
    input_pinned_memory {
        enable: true
    }
    output_pinned_memory {
        enable: true
    }
}
version_status {
    key: 1
    value {
        ready_state: MODEL_READY
        ready_state_reason {
        }
    }
}
ready_state: SERVER_READY
* Connection #0 to host 172.21.231.132 left intact

```

[Next: Deploy the Client for Triton Inference Server \(Automated Deployment\)](#)

#### **Deploy the Client for Triton Inference Server (Automated Deployment)**

To deploy the client for the Triton Inference Server, complete the following steps:

1. Open a VI editor, create a deployment for the Triton client, and call the file `triton_client.yaml`.

```
---  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  labels:  
    app: triton-client  
    name: triton-client  
    namespace: triton  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: triton-client  
      version: v1  
  template:  
    metadata:  
      labels:  
        app: triton-client  
        version: v1  
    spec:  
      containers:  
        - image: nvcr.io/nvidia/tritonserver:20.07- v1- py3-clientsdk  
          imagePullPolicy: IfNotPresent  
          name: triton-client  
          resources:  
            limits:  
              cpu: "2"  
              memory: 4Gi  
            requests:  
              cpu: "2"  
              memory: 4Gi
```

## 2. Deploy the client.

```
kubectl apply -f triton_client.yaml
```

Next: [Collect Inference Metrics from Triton Inference Server](#)

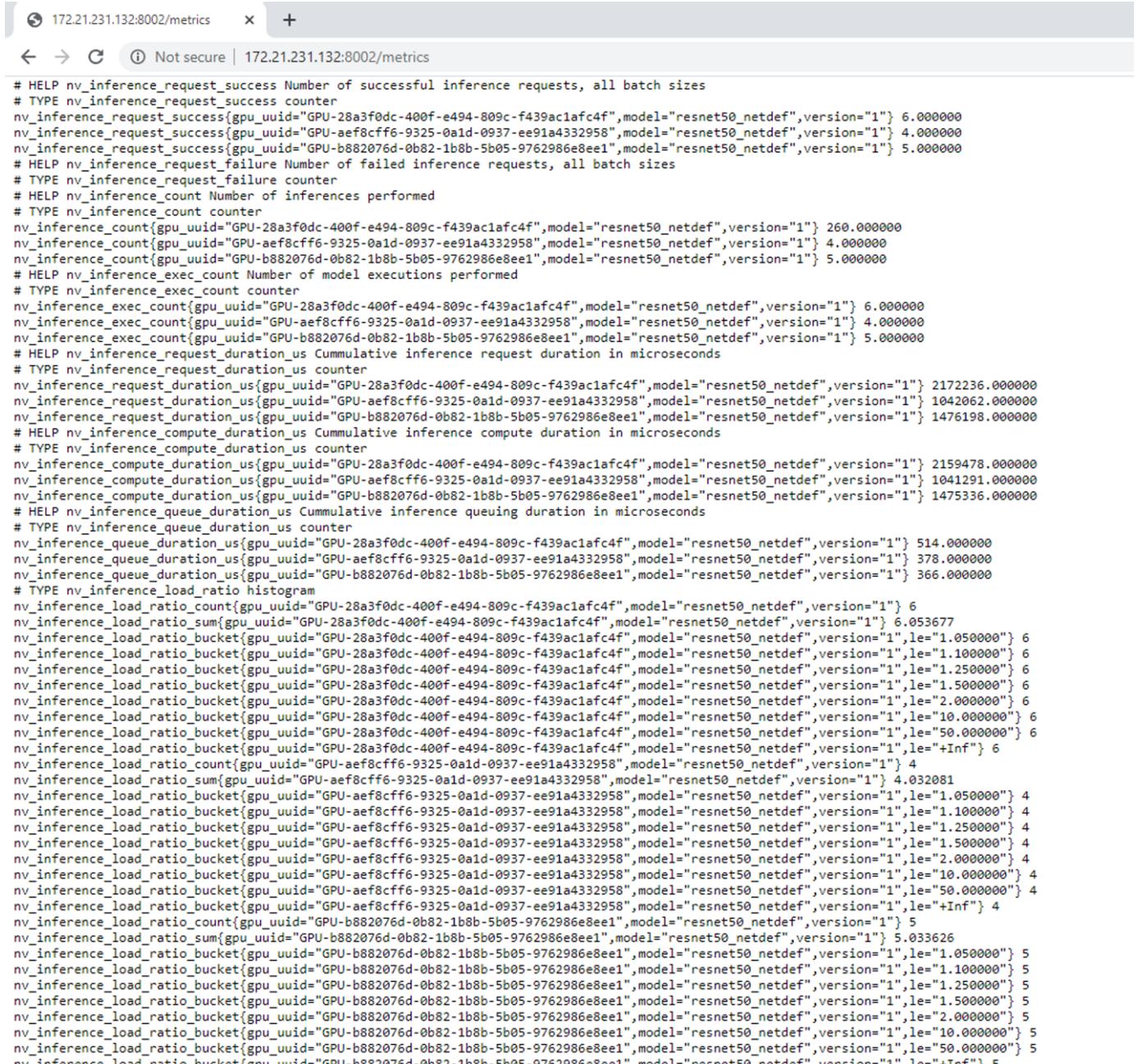
### Collect Inference Metrics from Triton Inference Server

The Triton Inference Server provides Prometheus metrics indicating GPU and request statistics.

By default, these metrics are available at "http://<triton\_inference\_server\_IP>:8002/metrics".

The Triton Inference Server IP is the LoadBalancer IP that was recorded earlier.

The metrics are only available by accessing the endpoint and are not pushed or published to any remote server.



```
# HELP nv_inference_request_success Number of successful inference requests, all batch sizes
# TYPE nv_inference_request_success counter
nv_inference_request_success{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 6.000000
nv_inference_request_success{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 4.000000
nv_inference_request_success{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 5.000000
# HELP nv_inference_request_failure Number of failed inference requests, all batch sizes
# TYPE nv_inference_request_failure counter
# HELP nv_inference_count Number of inferences performed
# TYPE nv_inference_count counter
nv_inference_count{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 260.000000
nv_inference_count{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 4.000000
nv_inference_count{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 5.000000
# HELP nv_inference_exec_count Number of model executions performed
# TYPE nv_inference_exec_count counter
nv_inference_exec_count{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 6.000000
nv_inference_exec_count{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 4.000000
nv_inference_exec_count{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 5.000000
# HELP nv_inference_request_duration_us Cumulative inference request duration in microseconds
# TYPE nv_inference_request_duration_us counter
nv_inference_request_duration_us{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 2172236.000000
nv_inference_request_duration_us{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 1042062.000000
nv_inference_request_duration_us{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 1476198.000000
# HELP nv_inference_compute_duration_us Cumulative inference compute duration in microseconds
# TYPE nv_inference_compute_duration_us counter
nv_inference_compute_duration_us{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 2159478.000000
nv_inference_compute_duration_us{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 1041291.000000
nv_inference_compute_duration_us{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 1475336.000000
# HELP nv_inference_queue_duration_us Cumulative inference queuing duration in microseconds
# TYPE nv_inference_queue_duration_us counter
nv_inference_queue_duration_us{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 514.000000
nv_inference_queue_duration_us{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 378.000000
nv_inference_queue_duration_us{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 366.000000
# TYPE nv_inference_load_ratio histogram
nv_inference_load_ratio_count{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 6
nv_inference_load_ratio_sum{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1"} 6.053677
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="1.050000"} 6
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="1.100000"} 6
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="1.250000"} 6
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="1.500000"} 6
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="2.000000"} 6
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="10.000000"} 6
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="50.000000"} 6
nv_inference_load_ratio_bucket{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f",model="resnet50_netdef",version="1",le="+Inf"} 6
nv_inference_load_ratio_count{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 4
nv_inference_load_ratio_sum{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1"} 4.032081
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="1.050000"} 4
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="1.100000"} 4
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="1.250000"} 4
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="1.500000"} 4
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="2.000000"} 4
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="10.000000"} 4
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="50.000000"} 4
nv_inference_load_ratio_bucket{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958",model="resnet50_netdef",version="1",le="+Inf"} 4
nv_inference_load_ratio_count{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 5
nv_inference_load_ratio_sum{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1"} 5.033626
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="1.050000"} 5
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="1.100000"} 5
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="1.250000"} 5
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="1.500000"} 5
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="2.000000"} 5
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="10.000000"} 5
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="50.000000"} 5
nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="+Inf"} 5
```

```

nv_inference_load_ratio_bucket{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1",model="resnet50_netdef",version="1",le="+Inf"} 5
# HELP nv_gpu_utilization GPU utilization rate [0.0 - 1.0)
# TYPE nv_gpu_utilization gauge
nv_gpu_utilization{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1"} 0.000000
nv_gpu_utilization{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f"} 0.000000
nv_gpu_utilization{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958"} 0.000000
# HELP nv_gpu_memory_total_bytes GPU total memory, in bytes
# TYPE nv_gpu_memory_total_bytes gauge
nv_gpu_memory_total_bytes{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1"} 15843721216.000000
nv_gpu_memory_total_bytes{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f"} 15843721216.000000
nv_gpu_memory_total_bytes{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958"} 15843721216.000000
# HELP nv_gpu_memory_used_bytes GPU used memory, in bytes
# TYPE nv_gpu_memory_used_bytes gauge
nv_gpu_memory_used_bytes{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1"} 1466236928.000000
nv_gpu_memory_used_bytes{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f"} 13004767232.000000
nv_gpu_memory_used_bytes{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958"} 1466236928.000000
# HELP nv_gpu_power_usage GPU power usage in watts
# TYPE nv_gpu_power_usage gauge
nv_gpu_power_usage{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1"} 27.999000
nv_gpu_power_usage{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f"} 28.428000
nv_gpu_power_usage{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958"} 27.632000
# HELP nv_gpu_power_limit GPU power management limit in watts
# TYPE nv_gpu_power_limit gauge
nv_gpu_power_limit{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1"} 70.000000
nv_gpu_power_limit{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f"} 70.000000
nv_gpu_power_limit{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958"} 70.000000
# HELP nv_energy_consumption GPU energy consumption in joules since the Triton Server started
# TYPE nv_energy_consumption counter
nv_energy_consumption{gpu_uuid="GPU-b882076d-0b82-1b8b-5b05-9762986e8ee1"} 9796.449000
nv_energy_consumption{gpu_uuid="GPU-28a3f0dc-400f-e494-809c-f439ac1afc4f"} 9997.538000
nv_energy_consumption{gpu_uuid="GPU-aef8cff6-9325-0a1d-0937-ee91a4332958"} 9669.536000

```

---

## Next: Validation Results

### Validation Results

To run a sample inference request, complete the following steps:

1. Get a shell to the client container/pod.

```
kubectl exec --stdin --tty <<client_pod_name>> -- /bin/bash
```

2. Run a sample inference request.

```
image_client -m resnet50_netdef -s INCEPTION -u
<<LoadBalancer_IP_recorded_earlier>>:8000 -c 3 images/mug.jpg
```

```
root@triton-client-v20-07-v1-5566895bc-zqz6w:/workspace# image_client -m resnet50_netdef -s INCEPTION -u 172.21.231.133:8000 -c 3 images/mug.jpg
Request 0, batch size 1
Image 'images/mug.jpg':
  504 (COFFEE MUG) = 0.723991
  968 (CUP) = 0.270953
  967 (ESPRESSO) = 0.00115996
```

This inferencing request calls the `resnet50_netdef` model that is used for image recognition. Other clients can also send inferencing requests concurrently by following a similar approach and calling out the appropriate model.

## Next: Where to Find Additional Information

### Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp HCI Theory of Operations  
<https://www.netapp.com/us/media/wp-7261.pdf>
- NetApp Product Documentation  
[docs.netapp.com](http://docs.netapp.com)
- NetApp HCI Solution Catalog Documentation  
<https://docs.netapp.com/us-en/hci/solutions/index.html>
- HCI Resources page  
<https://mysupport.netapp.com/info/web/ECMLP2831412.html>
- ONTAP Select  
<https://www.netapp.com/us/products/data-management-software/ontap-select-sds.aspx>
- NetApp Trident  
<https://netapp-trident.readthedocs.io/en/stable-v20.01/>
- NVIDIA DeepOps  
<https://github.com/NVIDIA/deepops>
- NVIDIA Triton Inference Server  
<https://docs.nvidia.com/deeplearning/sdk/triton-inference-server-master-branch-guide/docs/index.html>

## WP-7328: NetApp Conversational AI Using NVIDIA Jarvis

Rick Huang, Sung-Han Lin, NetApp  
Davide Onofrio, NVIDIA

The NVIDIA DGX family of systems is made up of the world's first integrated artificial intelligence (AI)-based systems that are purpose-built for enterprise AI. NetApp AFF storage systems deliver extreme performance and industry-leading hybrid cloud data-management capabilities. NetApp and NVIDIA have partnered to create the NetApp ONTAP AI reference architecture, a turnkey solution for AI and machine learning (ML) workloads that provides enterprise-class performance, reliability, and support.

This white paper gives directional guidance to customers building conversational AI systems in support of different use cases in various industry verticals. It includes information about the deployment of the system using NVIDIA Jarvis. The tests were performed using an NVIDIA DGX Station and a NetApp AFF A220 storage system.

The target audience for the solution includes the following groups:

- Enterprise architects who design solutions for the development of AI models and software for conversational AI use cases such as a virtual retail assistant
- Data scientists looking for efficient ways to achieve language modeling development goals
- Data engineers in charge of maintaining and processing text data such as customer questions and

dialogue transcripts

- Executive and IT decision makers and business leaders interested in transforming the conversational AI experience and achieving the fastest time to market from AI initiatives

[Next: Solution Overview](#)

## Solution Overview

### NetApp ONTAP AI and Cloud Sync

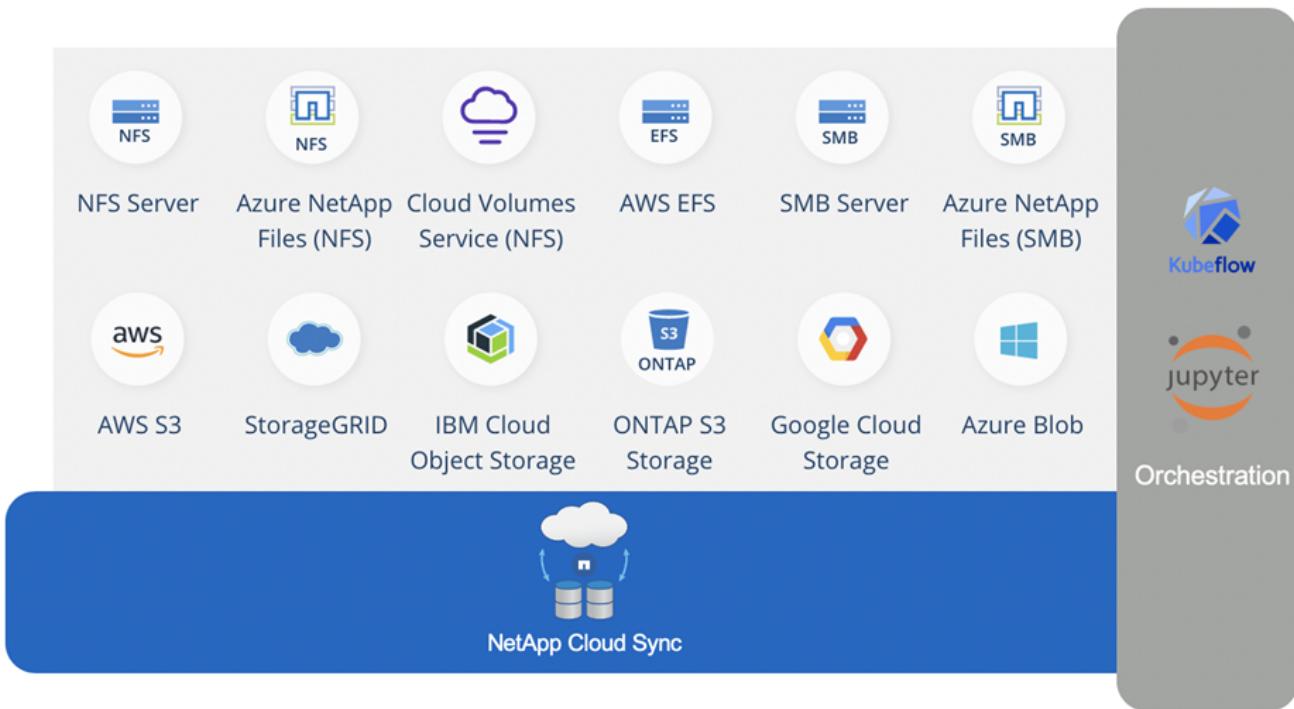
The NetApp ONTAP AI architecture, powered by NVIDIA DGX systems and NetApp cloud-connected storage systems, was developed and verified by NetApp and NVIDIA. This reference architecture gives IT organizations the following advantages:

- Eliminates design complexities
- Enables independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost pointsNetApp ONTAP AI tightly integrates DGX systems and NetApp AFF A220 storage systems with state-of-the-art networking. NetApp ONTAP AI and DGX systems simplify AI deployments by eliminating design complexity and guesswork. Customers can start small and grow their systems in an uninterrupted manner while intelligently managing data from the edge to the core to the cloud and back.

NetApp Cloud Sync enables you to move data easily over various protocols, whether it's between two NFS shares, two CIFS shares, or one file share and Amazon S3, Amazon Elastic File System (EFS), or Azure Blob storage. Active-active operation means that you can continue to work with both source and target at the same time, incrementally synchronizing data changes when required. By enabling you to move and incrementally synchronize data between any source and destination system, whether on-premises or cloud-based, Cloud Sync opens up a wide variety of new ways in which you can use data. Migrating data between on-premises systems, cloud on-boarding and cloud migration, or collaboration and data analytics all become easily achievable. The figure below shows available sources and destinations.

In conversational AI systems, developers can leverage Cloud Sync to archive conversation history from the cloud to data centers to enable offline training of natural language processing (NLP) models. By training models to recognize more intents, the conversational AI system will be better equipped to manage more complex questions from end-users.

### NVIDIA Jarvis Multimodal Framework



[NVIDIA Jarvis](#) is an end-to-end framework for building conversational AI services. It includes the following GPU-optimized services:

- Automatic speech recognition (ASR)
- Natural language understanding (NLU)
- Integration with domain-specific fulfillment services
- Text-to-speech (TTS)
- Computer vision (CV) Jarvis-based services use state-of-the-art deep learning models to address the complex and challenging task of real-time conversational AI. To enable real-time, natural interaction with an end user, the models need to complete computation in under 300 milliseconds. Natural interactions are challenging, requiring multimodal sensory integration. Model pipelines are also complex and require coordination across the above services.

Jarvis is a fully accelerated, application framework for building multimodal conversational AI services that use an end-to-end deep learning pipeline. The Jarvis framework includes pretrained conversational AI models, tools, and optimized end-to-end services for speech, vision, and NLU tasks. In addition to AI services, Jarvis enables you to fuse vision, audio, and other sensor inputs simultaneously to deliver capabilities such as multi-user, multi-context conversations in applications such as virtual assistants, multi-user diarization, and call center assistants.

### NVIDIA NeMo

[NVIDIA NeMo](#) is an open-source Python toolkit for building, training, and fine-tuning GPU-accelerated state-of-the-art conversational AI models using easy-to-use application programming interfaces (APIs). NeMo runs mixed precision compute using Tensor Cores in NVIDIA GPUs and can scale up to multiple GPUs easily to deliver the highest training performance possible. NeMo is used to build models for real-time ASR, NLP, and TTS applications such as video call transcriptions, intelligent video assistants, and automated call center support across different industry verticals, including healthcare, finance, retail, and telecommunications.

We used NeMo to train models that recognize complex intents from user questions in archived conversation history. This training extends the capabilities of the retail virtual assistant beyond what Jarvis supports as

delivered.

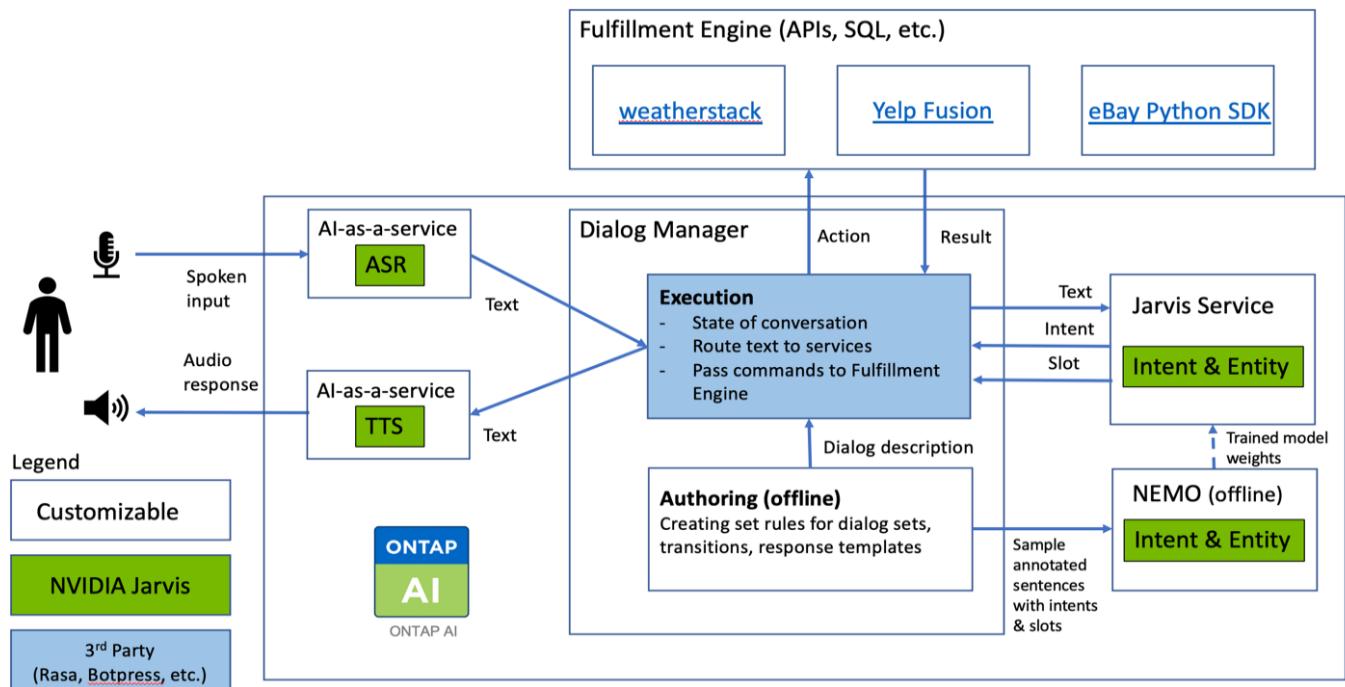
## Retail Use Case Summary

Using NVIDIA Jarvis, we built a virtual retail assistant that accepts speech or text input and answers questions regarding weather, points-of-interest, and inventory pricing. The conversational AI system is able to remember conversation flow, for example, ask a follow-up question if the user does not specify location for weather or points-of-interest. The system also recognizes complex entities such as “Thai food” or “laptop memory.” It understands natural language questions like “will it rain next week in Los Angeles?” A demonstration of the retail virtual assistant can be found in [Customize States and Flows for Retail Use Case](#).

Next: Solution Technology

## Solution Technology

The following figure illustrates the proposed conversational AI system architecture. You can interact with the system with either speech signal or text input. If spoken input is detected, Jarvis Al-as-service (AlaaS) performs ASR to produce text for Dialog Manager. Dialog Manager remembers states of conversation, routes text to corresponding services, and passes commands to Fulfillment Engine. Jarvis NLP Service takes in text, recognizes intents and entities, and outputs those intents and entity slots back to Dialog Manager, which then sends Action to Fulfillment Engine. Fulfillment Engine consists of third-party APIs or SQL databases that answer user queries. After receiving Result from Fulfillment Engine, Dialog Manager routes text to Jarvis TTS AlaaS to produce an audio response for the end-user. We can archive conversation history, annotate sentences with intents and slots for NeMo training such that NLP Service improves as more users interact with the system.



## Hardware Requirements

This solution was validated using one DGX Station and one AFF A220 storage system. Jarvis requires either a T4 or V100 GPU to perform deep neural network computations.

The following table lists the hardware components that are required to implement the solution as tested.

Hardware	Quantity
T4 or V100 GPU	1
NVIDIA DGX Station	1

## Software Requirements

The following table lists the software components that are required to implement the solution as tested.

Software	Version or Other Information
NetApp ONTAP data management software	9.6
Cisco NX-OS switch firmware	7.0(3)I6(1)
NVIDIA DGX OS	4.0.4 - Ubuntu 18.04 LTS
NVIDIA Jarvis Framework	EA v0.2
NVIDIA NeMo	nvcr.io/nvidia/nemo:v0.10
Docker container platform	18.06.1-ce [e68fc7a]

[Next: Build a Virtual Assistant Using Jarvis, Cloud Sync, and NeMo Overview](#)

## Overview

This section provides detail on the implementation of the virtual retail assistant.

[Next: Jarvis Deployment](#)

### Jarvis Deployment

You can sign up for [Jarvis Early Access program](#) to gain access to Jarvis containers on NVIDIA GPU Cloud (NGC). After receiving credentials from NVIDIA, you can deploy Jarvis using the following steps:

1. Sign-on to NGC.
2. Set your organization on NGC: `ea-2-jarvis`.
3. Locate Jarvis EA v0.2 assets: Jarvis containers are in [Private Registry > Organization Containers](#).
4. Select Jarvis: navigate to [Model Scripts](#) and click [Jarvis Quick Start](#)
5. Verify that all assets are working properly.
6. Find the documentation to build your own applications: PDFs can be found in [Model Scripts > Jarvis Documentation > File Browser](#).

[Next: Customize States and Flows for Retail Use Case](#)

### Customize States and Flows for Retail Use Case

You can customize States and Flows of Dialog Manager for your specific use cases. In our retail example, we have the following four yaml files to direct the conversation

according to different intents.

See the following list of file names and description of each file:

- `main_flow.yml`: Defines the main conversation flows and states and directs the flow to the other three yaml files when necessary.
- `retail_flow.yml`: Contains states related to retail or points-of-interest questions. The system either provides the information of the nearest store, or the price of a given item.
- `weather_flow.yml`: Contains states related to weather questions. If the location cannot be determined, the system asks a follow up question to clarify.
- `error_flow.yml`: Handles cases where user intents do not fall into the above three yaml files. After displaying an error message, the system re-routes back to accepting user questions. The following sections contain the detailed definitions for these yaml files.

### `main_flow.yml`

```
name: JarvisRetail
intent_transitions:
    jarvis_error: error
    price_check: retail_price_check
    inventory_check: retail_inventory_check
    store_location: retail_store_location
    weather.weather: weather
    weather.temperature: temperature
    weather.sunny: sunny
    weather.cloudy: cloudy
    weather.snow: snow
    weather.rainfall: rain
    weather.snow_yes_no: snowfall
    weather.rainfall_yes_no: rainfall
    weather.temperature_yes_no: tempyesno
    weather.humidity: humidity
    weather.humidity_yes_no: humidity
    navigation.startnavigationpoi: retail # Transitions should be context
and slot based. Redirecting for now.
    navigation.geteta: retail
    navigation.showdirection: retail
    navigation.showmappoi: idk_what_you_talkin_about
    nomatch.none: idk_what_you_talkin_about
states:
    init:
        type: message_text
        properties:
            text: "Hi, welcome to NARA retail and weather service. How can I
help you?"
        input_intent:
```

```

type: input_context
properties:
  nlp_type: jarvis
  entities:
    intent: dontcare
# This state is executed if the intent was not understood
dont_get_the_intent:
  type: message_text_random
  properties:
    responses:
      - "Sorry I didn't get that! Please come again."
      - "I beg your pardon! Say that again?"
      - "Are we talking about weather? What would you like to know?"
      - "Sorry I know only about the weather"
      - "You can ask me about the weather, the rainfall, the
temperature, I don't know much more"
  delay: 0
  transitions:
    next_state: input_intent
idk_what_you_talkin_about:
  type: message_text_random
  properties:
    responses:
      - "Sorry I didn't get that! Please come again."
      - "I beg your pardon! Say that again?"
      - "Are we talking about retail or weather? What would you like to
know?"
      - "Sorry I know only about retail and the weather"
      - "You can ask me about retail information or the weather, the
rainfall, the temperature. I don't know much more."
  delay: 0
  transitions:
    next_state: input_intent
error:
  type: change_context
  properties:
    update_keys:
      intent: 'error'
  transitions:
    flow: error_flow
retail_inventory_check:
  type: change_context
  properties:
    update_keys:
      intent: 'retail_inventory_check'
  transitions:

```

```
        flow: retail_flow
retail_price_check:
    type: change_context
properties:
    update_keys:
        intent: 'check_item_price'
transitions:
    flow: retail_flow
retail_store_location:
    type: change_context
properties:
    update_keys:
        intent: 'find_the_store'
transitions:
    flow: retail_flow
weather:
    type: change_context
properties:
    update_keys:
        intent: 'weather'
transitions:
    flow: weather_flow
temperature:
    type: change_context
properties:
    update_keys:
        intent: 'temperature'
transitions:
    flow: weather_flow
rainfall:
    type: change_context
properties:
    update_keys:
        intent: 'rainfall'
transitions:
    flow: weather_flow
sunny:
    type: change_context
properties:
    update_keys:
        intent: 'sunny'
transitions:
    flow: weather_flow
cloudy:
    type: change_context
properties:
```

```
    update_keys:
        intent: 'cloudy'
transitions:
    flow: weather_flow
snow:
    type: change_context
properties:
    update_keys:
        intent: 'snow'
transitions:
    flow: weather_flow
rain:
    type: change_context
properties:
    update_keys:
        intent: 'rain'
transitions:
    flow: weather_flow
snowfall:
    type: change_context
properties:
    update_keys:
        intent: 'snowfall'
transitions:
    flow: weather_flow
tempyesno:
    type: change_context
properties:
    update_keys:
        intent: 'tempyesno'
transitions:
    flow: weather_flow
humidity:
    type: change_context
properties:
    update_keys:
        intent: 'humidity'
transitions:
    flow: weather_flow
end_state:
    type: reset
transitions:
    next_state: init
```

## retail\_flow.yml

```
name: retail_flow
states:
  store_location:
    type: conditional_exists
    properties:
      key: '{{location}}'
    transitions:
      exists: retail_state
      notexists: ask_retail_location
  retail_state:
    type: Retail
    properties:
    transitions:
      next_state: output_retail
  output_retail:
    type: message_text
    properties:
      text: '{{retail_status}}'
    transitions:
      next_state: input_intent
  ask_retail_location:
    type: message_text
    properties:
      text: "For which location? I can find the closest store near you."
    transitions:
      next_state: input_retail_location
  input_retail_location:
    type: input_user
    properties:
      nlp_type: jarvis
      entities:
        slot: location
        require_match: true
    transitions:
      match: retail_state
      notmatch: check_retail_jarvis_error
  output_retail_acknowledge:
    type: message_text_random
    properties:
      responses:
        - 'ok in {{location}}'
        - 'the store in {{location}}'
        - 'I always wanted to shop in {{location}}'
    delay: 0
```

```

transitions:
  next_state: retail_state
output_retail_notlocation:
  type: message_text
  properties:
    text: "I did not understand the location. Can you please repeat?"
transitions:
  next_state: input_intent
check_rerail_jarvis_error:
  type: conditional_exists
  properties:
    key: '{{jarvis_error}}'
transitions:
  exists: show_retail_jarvis_api_error
  notexists: output_retail_notlocation
show_retail_jarvis_api_error:
  type: message_text
  properties:
    text: "I am having trouble understanding right now. Come again on
that?"
transitions:
  next_state: input_intent

```

## **weather\_flow.yml**

```

name: weather_flow
states:
  check_weather_location:
    type: conditional_exists
    properties:
      key: '{{location}}'
    transitions:
      exists: weather_state
      notexists: ask_weather_location
  weather_state:
    type: Weather
    properties:
    transitions:
      next_state: output_weather
  output_weather:
    type: message_text
    properties:
      text: '{{weather_status}}'
    transitions:
      next_state: input_intent

```

```
ask_weather_location:
    type: message_text
    properties:
        text: "For which location?"
transitions:
    next_state: input_weather_location
input_weather_location:
    type: input_user
    properties:
        nlp_type: jarvis
        entities:
            slot: location
            require_match: true
transitions:
    match: weather_state
    notmatch: check_jarvis_error
output_weather_acknowledge:
    type: message_text_random
    properties:
        responses:
            - 'ok in {{location}}'
            - 'the weather in {{location}}'
            - 'I always wanted to go in {{location}}'
    delay: 0
transitions:
    next_state: weather_state
output_weather_notlocation:
    type: message_text
    properties:
        text: "I did not understand the location, can you please repeat?"
transitions:
    next_state: input_intent
check_jarvis_error:
    type: conditional_exists
    properties:
        key: '{{jarvis_error}}'
transitions:
    exists: show_jarvis_api_error
    notexists: output_weather_notlocation
show_jarvis_api_error:
    type: message_text
    properties:
        text: "I am having troubled understanding right now. Come again on
that, else check jarvis services?"
transitions:
    next_state: input_intent
```

## error\_flow.yml

```
name: error_flow
states:
  error_state:
    type: message_text_random
    properties:
      responses:
        - "Sorry I didn't get that!"
        - "Are we talking about retail or weather? What would you like to know?"
        - "Sorry I know only about retail information or the weather"
        - "You can ask me about retail information or the weather, the rainfall, the temperature. I don't know much more"
        - "Let's talk about retail or the weather!"
    delay: 0
    transitions:
      next_state: input_intent
```

[Next: Connect to Third-Party APIs as Fulfillment Engine](#)

### Connect to Third-Party APIs as Fulfillment Engine

We connected the following third-party APIs as a Fulfillment Engine to answer questions:

- [WeatherStack API](#): returns weather, temperature, rainfall, and snow in a given location.
- [Yelp Fusion API](#): returns the nearest store information in a given location.
- [eBay Python SDK](#): returns the price of a given item.

[Next: NetApp Retail Assistant Demonstration](#)

### NetApp Retail Assistant Demonstration

We recorded a demonstration video of NetApp Retail Assistant (NARA). Click [this link](#) to open the following figure and play the video demonstration.

# NetApp NARA



Hi, welcome to NARA retail and weather service. How can I help you?

Write your message...

Submit

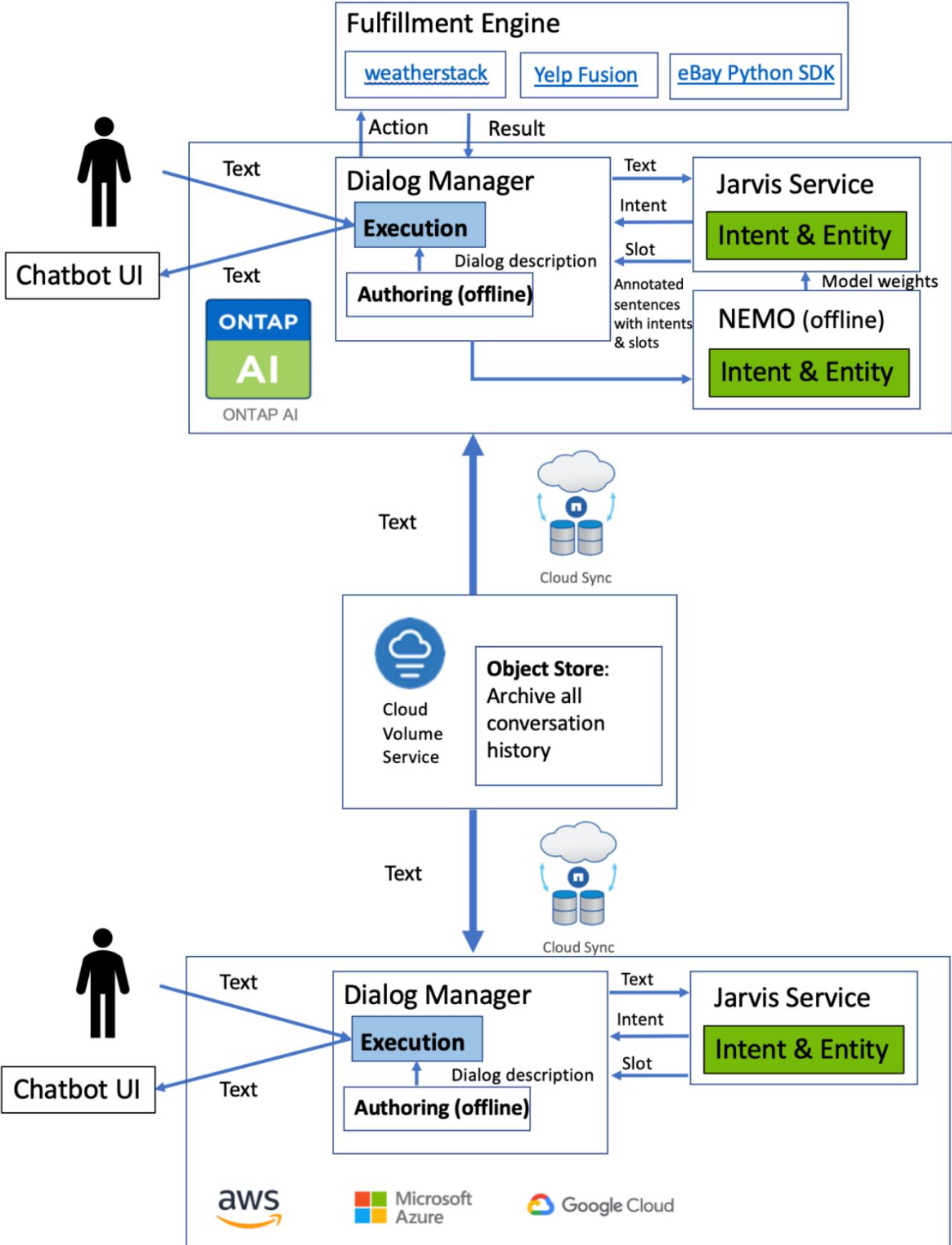
System replied. Waiting for user input.

Unmute System Speech

Next: Use NetApp Cloud Sync to Archive Conversation History

## Use NetApp Cloud Sync to Archive Conversation History

By dumping conversation history into a CSV file once a day, we can then leverage Cloud Sync to download the log files into local storage. The following figure shows the architecture of having Jarvis deployed on-premises and in public clouds, while using Cloud Sync to send conversation history for NeMo training. Details of NeMo training can be found in the section [Expand Intent Models Using NeMo Training](#).



Next: Expand Intent Models Using NeMo Training

## Expand Intent Models Using NeMo Training

NVIDIA NeMo is a toolkit built by NVIDIA for creating conversational AI applications. This toolkit includes collections of pre-trained modules for ASR, NLP, and TTS, enabling researchers and data scientists to easily compose complex neural network architectures and put more focus on designing their own applications.

As shown in the previous example, NARA can only handle a limited type of question. This is because the pre-trained NLP model only trains on these types of questions. If we want to enable NARA to handle a broader range of questions, we need to retrain it with our own datasets. Thus, here, we demonstrate how we can use NeMo to extend the NLP model to satisfy the requirements. We start by converting the log collected from NARA into the format for NeMo, and then train with the dataset to enhance the NLP model.

### Model

Our goal is to enable NARA to sort the items based on user preferences. For instance, we might ask NARA to suggest the highest-rated sushi restaurant or might want NARA to look up the jeans with the lowest price. To this end, we use the intent detection and slot filling model provided in NeMo as our training model. This model allows NARA to understand the intent of searching preference.

### Data Preparation

To train the model, we collect the dataset for this type of question, and convert it to the NeMo format. Here, we listed the files we use to train the model.

#### dict.intents.csv

This file lists all the intents we want the NeMo to understand. Here, we have two primary intents and one intent only used to categorize the questions that do not fit into any of the primary intents.

```
price_check  
find_the_store  
unknown
```

#### dict.slots.csv

This file lists all the slots we can label on our training questions.

```
B-store.type  
B-store.name  
B-store.status  
B-store.hour.start  
B-store.hour.end  
B-store.hour.day  
B-item.type  
B-item.name  
B-item.color  
B-item.size  
B-item.quantity  
B-location  
B-cost.high
```

```
B-cost.average
B-cost.low
B-time.period_of_time
B-rating.high
B-rating.average
B-rating.low
B-interrogative.location
B-interrogative.manner
B-interrogative.time
B-interrogative.personal
B-interrogative
B-verb
B-article
I-store.type
I-store.name
I-store.status
I-store.hour.start
I-store.hour.end
I-store.hour.day
I-item.type
I-item.name
I-item.color
I-item.size
I-item.quantity
I-location
I-cost.high
I-cost.average
I-cost.low
I-time.period_of_time
I-rating.high
I-rating.average
I-rating.low
I-interrogative.location
I-interrogative.manner
I-interrogative.time
I-interrogative.personal
I-interrogative
I-verb
I-article
O
```

### train.tsv

This is the main training dataset. Each line starts with the question following the intent category listing in the file dict.intent.csv. The label is enumerated starting from zero.

## train\_slots.tsv

```
20 46 24 25 6 32 6  
52 52 24 6  
23 52 14 40 52 25 6 32 6  
...
```

## Train the Model

```
docker pull nvcr.io/nvidia/nemo:v0.10
```

We then use the following command to launch the container. In this command, we limit the container to use a single GPU (GPU ID = 1) since this is a lightweight training exercise. We also map our local workspace /workspace/nemo/ to the folder inside container /nemo.

```
NV_GPU='1' docker run --runtime=nvidia -it --shm-size=16g \  
--network=host --ulimit memlock=-1 --ulimit  
stack=67108864 \  
-v /workspace/nemo:/nemo\  
--rm nvcr.io/nvidia/nemo:v0.10
```

Inside the container, if we want to start from the original pre-trained BERT model, we can use the following command to start the training procedure. `data_dir` is the argument to set up the path of the training data. `work_dir` allows you to configure where you want to store the checkpoint files.

```
cd examples/nlp/intent_detection_slot_tagging/  
python joint_intent_slot_with_bert.py \  
--data_dir /nemo/training_data\  
--work_dir /nemo/log
```

If we have new training datasets and want to improve the previous model, we can use the following command to continue from the point we stopped. `checkpoint_dir` takes the path to the previous checkpoints folder.

```
cd examples/nlp/intent_detection_slot_tagging/  
python joint_intent_slot_infer.py \  
--data_dir /nemo/training_data \  
--checkpoint_dir /nemo/log/2020-05-04_18-34-20/checkpoints/ \  
--eval_file_prefix test
```

## Inference the Model

We need to validate the performance of the trained model after a certain number of epochs. The following command allows us to test the query one-by-one. For instance, in this command, we want to check if our

model can properly identify the intention of the query `where can I get the best pasta`.

```
cd examples/nlp/intent_detection_slot_tagging/
python joint_intent_slot_infer_b1.py \
--checkpoint_dir /nemo/log/2020-05-29_23-50-58/checkpoints/ \
--query "where can i get the best pasta" \
--data_dir /nemo/training_data/ \
--num_epochs=50
```

Then, the following is the output from the inference. In the output, we can see that our trained model can properly predict the intention `find_the_store`, and return the keywords we are interested in. With these keywords, we enable the NARA to search for what users want and do a more precise search.

```
[NeMo I 2020-05-30 00:06:54 actions:728] Evaluating batch 0 out of 1
[NeMo I 2020-05-30 00:06:55 inference_utils:34] Query: where can i get the
best pasta
[NeMo I 2020-05-30 00:06:55 inference_utils:36] Predicted intent: 1
find_the_store
[NeMo I 2020-05-30 00:06:55 inference_utils:50] where B-
interrogative.location
[NeMo I 2020-05-30 00:06:55 inference_utils:50] can O
[NeMo I 2020-05-30 00:06:55 inference_utils:50] i O
[NeMo I 2020-05-30 00:06:55 inference_utils:50] get B-verb
[NeMo I 2020-05-30 00:06:55 inference_utils:50] the B-article
[NeMo I 2020-05-30 00:06:55 inference_utils:50] best B-rating.high
[NeMo I 2020-05-30 00:06:55 inference_utils:50] pasta B-item.type
```

[Next: Conclusion](#)

## Conclusion

A true conversational AI system engages in human-like dialogue, understands context, and provides intelligent responses. Such AI models are often huge and highly complex. With NVIDIA GPUs and NetApp storage, massive, state-of-the-art language models can be trained and optimized to run inference rapidly. This is a major stride towards ending the trade-off between an AI model that is fast versus one that is large and complex. GPU-optimized language understanding models can be integrated into AI applications for industries such as healthcare, retail, and financial services, powering advanced digital voice assistants in smart speakers and customer service lines. These high-quality conversational AI systems allow businesses across verticals to provide previously unattainable personalized services when engaging with customers.

Jarvis enables the deployment of use cases such as virtual assistants, digital avatars, multimodal sensor fusion (CV fused with ASR/NLP/TTS), or any ASR/NLP/TTS/CV stand-alone use case, such as transcription. We built a virtual retail assistant that can answer questions regarding weather, points-of-interest, and inventory pricing. We also demonstrated how to improve the natural language understanding capabilities of the conversational AI system by archiving conversation history using Cloud Sync and training NeMo models on new data.

[Next: Acknowledgments](#)

## Acknowledgments

The authors gratefully acknowledge the contributions that were made to this white paper by our esteemed colleagues from NVIDIA: Davide Onofrio, Alex Qi, Sicong Ji, Marty Jain, and Robert Sohigian. The authors would also like to acknowledge the contributions of key NetApp team members: Santosh Rao, David Arnette, Michael Oglesby, Brent Davis, Andy Sayare, Erik Mulder, and Mike McNamara.

Our sincere appreciation and thanks go to all these individuals, who provided insight and expertise that greatly assisted in the creation of this paper.

## Next: Where to Find Additional Information

### Where to Find Additional Information

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

- NVIDIA DGX Station, V100 GPU, GPU Cloud
  - NVIDIA DGX Station  
<https://www.nvidia.com/en-us/data-center/dgx-station/>
  - NVIDIA V100 Tensor Core GPU  
<https://www.nvidia.com/en-us/data-center/tesla-v100/>
  - NVIDIA NGC  
<https://www.nvidia.com/en-us/gpu-cloud/>
- NVIDIA Jarvis Multimodal Framework
  - NVIDIA Jarvis  
<https://developer.nvidia.com/nvidia-jarvis>
  - NVIDIA Jarvis Early Access  
<https://developer.nvidia.com/nvidia-jarvis-early-access>
- NVIDIA NeMo
  - NVIDIA NeMo  
<https://developer.nvidia.com/nvidia-nemo>
  - Developer Guide  
<https://nvidia.github.io/NeMo/>
- NetApp AFF systems
  - NetApp AFF A-Series Datasheet  
<https://www.netapp.com/us/media/ds-3582.pdf>
  - NetApp Flash Advantage for All Flash FAS  
<https://www.netapp.com/us/media/ds-3733.pdf>
  - ONTAP 9 Information Library  
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>
  - NetApp ONTAP FlexGroup Volumes technical report  
<https://www.netapp.com/us/media/tr-4557.pdf>
- NetApp ONTAP AI

- ONTAP AI with DGX-1 and Cisco Networking Design Guide  
<https://www.netapp.com/us/media/nva-1121-design.pdf>
- ONTAP AI with DGX-1 and Cisco Networking Deployment Guide  
<https://www.netapp.com/us/media/nva-1121-deploy.pdf>
- ONTAP AI with DGX-1 and Mellanox Networking Design Guide  
<http://www.netapp.com/us/media/nva-1138-design.pdf>
- ONTAP AI with DGX-2 Design Guide  
<https://www.netapp.com/us/media/nva-1135-design.pdf>

## TR-4858: NetApp Orchestration Solution with Run:AI

Rick Huang, David Arnette, Sung-Han Lin, NetApp  
Yaron Goldberg, Run:AI

NetApp AFF storage systems deliver extreme performance and industry-leading hybrid cloud data-management capabilities. NetApp and Run:AI have partnered to demonstrate the unique capabilities of the NetApp ONTAP AI solution for artificial intelligence (AI) and machine learning (ML) workloads that provides enterprise-class performance, reliability, and support. Run:AI orchestration of AI workloads adds a Kubernetes-based scheduling and resource utilization platform to help researchers manage and optimize GPU utilization. Together with the NVIDIA DGX systems, the combined solution from NetApp, NVIDIA, and Run:AI provide an infrastructure stack that is purpose-built for enterprise AI workloads. This technical report gives directional guidance to customers building conversational AI systems in support of various use cases and industry verticals. It includes information about the deployment of Run:AI and a NetApp AFF A800 storage system and serves as a reference architecture for the simplest way to achieve fast, successful deployment of AI initiatives.

The target audience for the solution includes the following groups:

- Enterprise architects who design solutions for the development of AI models and software for Kubernetes-based use cases such as containerized microservices
- Data scientists looking for efficient ways to achieve efficient model development goals in a cluster environment with multiple teams and projects
- Data engineers in charge of maintaining and running production models
- Executive and IT decision makers and business leaders who would like to create the optimal Kubernetes cluster resource utilization experience and achieve the fastest time to market from AI initiatives

[Next: Solution Overview](#)

### Solution Overview

#### NetApp ONTAP AI and AI Control Plane

The NetApp ONTAP AI architecture, developed and verified by NetApp and NVIDIA, is powered by NVIDIA DGX systems and NetApp cloud-connected storage systems. This reference architecture gives IT organizations the following advantages:

- Eliminates design complexities
- Enables independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost points

NetApp ONTAP AI tightly integrates DGX systems and NetApp AFF A800 storage systems with state-of-the-art networking. NetApp ONTAP AI and DGX systems simplify AI deployments by eliminating design complexity and guesswork. Customers can start small and grow their systems in an uninterrupted manner while intelligently managing data from the edge to the core to the cloud and back.

NetApp AI Control Plane is a full stack AI, ML, and deep learning (DL) data and experiment management solution for data scientists and data engineers. As organizations increase their use of AI, they face many challenges, including workload scalability and data availability. NetApp AI Control Plane addresses these challenges through functionalities, such as rapidly cloning a data namespace just as you would a Git repo, and defining and implementing AI training workflows that incorporate the near-instant creation of data and model baselines for traceability and versioning. With NetApp AI Control Plane, you can seamlessly replicate data across sites and regions and swiftly provision Jupyter Notebook workspaces with access to massive datasets.

### Run:AI Platform for AI Workload Orchestration

Run:AI has built the world's first orchestration and virtualization platform for AI infrastructure. By abstracting workloads from the underlying hardware, Run:AI creates a shared pool of GPU resources that can be dynamically provisioned, enabling efficient orchestration of AI workloads and optimized use of GPUs. Data scientists can seamlessly consume massive amounts of GPU power to improve and accelerate their research while IT teams retain centralized, cross-site control and real-time visibility over resource provisioning, queuing, and utilization. The Run:AI platform is built on top of Kubernetes, enabling simple integration with existing IT and data science workflows.

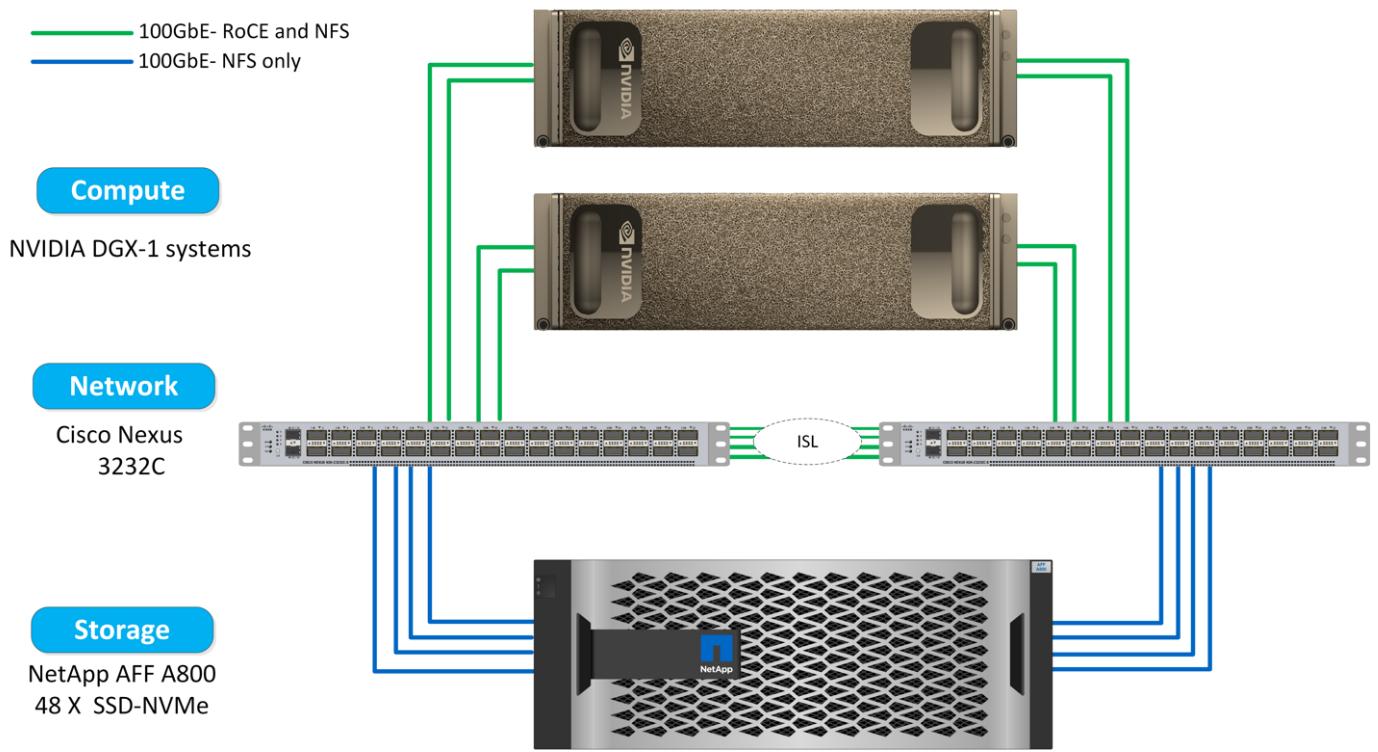
The Run:AI platform provides the following benefits:

- **Faster time to innovation.** By using Run:AI resource pooling, queueing, and prioritization mechanisms together with a NetApp storage system, researchers are removed from infrastructure management hassles and can focus exclusively on data science. Run:AI and NetApp customers increase productivity by running as many workloads as they need without compute or data pipeline bottlenecks.
- **Increased team productivity.** Run:AI fairness algorithms guarantee that all users and teams get their fair share of resources. Policies around priority projects can be preset, and the platform enables dynamic allocation of resources from one user or team to another, helping users to get timely access to coveted GPU resources.
- **Improved GPU utilization.** The Run:AI Scheduler enables users to easily make use of fractional GPUs, integer GPUs, and multiple nodes of GPUs for distributed training on Kubernetes. In this way, AI workloads run based on your needs, not capacity. Data science teams are able to run more AI experiments on the same infrastructure.

[Next: Solution Technology](#)

### Solution Technology

This solution was implemented with one NetApp AFF A800 system, two DGX-1 servers, and two Cisco Nexus 3232C 100GbE-switches. Each DGX-1 server is connected to the Nexus switches with four 100GbE connections that are used for inter-GPU communications by using remote direct memory access (RDMA) over Converged Ethernet (RoCE). Traditional IP communications for NFS storage access also occur on these links. Each storage controller is connected to the network switches by using four 100GbE-links. The following figure shows the ONTAP AI solution architecture used in this technical report for all testing scenarios.



#### Hardware Used in This Solution

This solution was validated using the ONTAP AI reference architecture two DGX-1 nodes and one AFF A800 storage system. See [NVA-1121](#) for more details about the infrastructure used in this validation.

The following table lists the hardware components that are required to implement the solution as tested.

Hardware	Quantity
DGX-1 systems	2
AFF A800	1
Nexus 3232C switches	2

#### Software Requirements

This solution was validated using a basic Kubernetes deployment with the Run:AI operator installed. Kubernetes was deployed using the [NVIDIA DeepOps](#) deployment engine, which deploys all required components for a production-ready environment. DeepOps automatically deployed [NetApp Trident](#) for persistent storage integration with the k8s environment, and default storage classes were created so containers leverage storage from the AFF A800 storage system. For more information on Trident with Kubernetes on ONTAP AI, see [TR-4798](#).

The following table lists the software components that are required to implement the solution as tested.

Software	Version or Other Information
NetApp ONTAP data management software	9.6p4
Cisco NX-OS switch firmware	7.0(3)I6(1)
NVIDIA DGX OS	4.0.4 - Ubuntu 18.04 LTS

Software	Version or Other Information
Kubernetes version	1.17
Trident version	20.04.0
Run:AI CLI	v2.1.13
Run:AI Orchestration Kubernetes Operator version	1.0.39
Docker container platform	18.06.1-ce [e68fc7a]

Additional software requirements for Run:AI can be found at [Run:AI GPU cluster prerequisites](#).

## Next: Optimal Cluster and GPU Utilization with Run AI

### Optimal Cluster and GPU Utilization with Run:AI

The following sections provide details on the Run:AI installation, test scenarios, and results performed in this validation.

We validated the operation and performance of this system by using industry standard benchmark tools, including TensorFlow benchmarks. The ImageNet dataset was used to train ResNet-50, which is a famous Convolutional Neural Network (CNN) DL model for image classification. ResNet-50 delivers an accurate training result with a faster processing time, which enabled us to drive a sufficient demand on the storage.

## Next: Run AI Installation.

### Run:AI Installation

To install Run:AI, complete the following steps:

1. Install the Kubernetes cluster using DeepOps and configure the NetApp default storage class.
2. Prepare GPU nodes:
  - a. Verify that NVIDIA drivers are installed on GPU nodes.
  - b. Verify that `nvidia-docker` is installed and configured as the default docker runtime.
3. Install Run:AI:
  - a. Log into the [Run:AI Admin UI](#) to create the cluster.
  - b. Download the created `runai-operator-<clustername>.yaml` file.
  - c. Apply the operator configuration to the Kubernetes cluster.

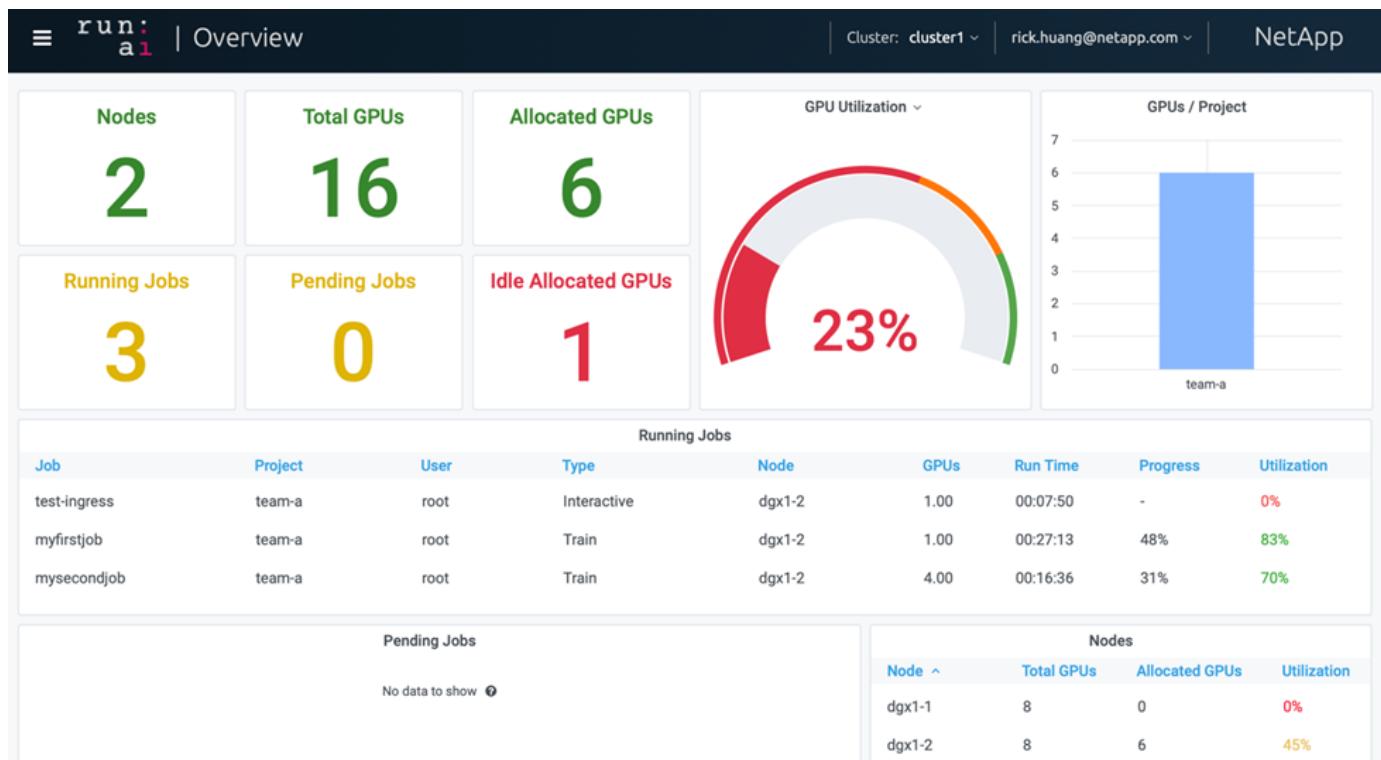
```
kubectl apply -f runai-operator-<clustername>.yaml
```

4. Verify the installation:
  - a. Go to <https://app.run.ai/>.
  - b. Go to the Overview dashboard.
  - c. Verify that the number of GPUs on the top right reflects the expected number of GPUs and the GPU nodes are all in the list of servers. For more information about Run:AI deployment, see [Installing Run:AI on an on-premise Kubernetes cluster](#) and [Installing the Run:AI CLI](#).

Next: Run AI Dashboards and Views

## Run:AI Dashboards and Views

After installing Run:AI on your Kubernetes cluster and configuring the containers correctly, you see the following dashboards and views on <https://app.run.ai> in your browser, as shown in the following figure.



There are 16 total GPUs in the cluster provided by two DGX-1 nodes. You can see the number of nodes, the total available GPUs, the allocated GPUs that are assigned with workloads, the total number of running jobs, pending jobs, and idle allocated GPUs. On the right side, the bar diagram shows GPUs per Project, which summarizes how different teams are using the cluster resource. In the middle is the list of currently running jobs with job details, including job name, project, user, job type, the node each job is running on, the number of GPU(s) allocated for that job, the current run time of the job, job progress in percentage, and the GPU utilization for that job. Note that the cluster is under-utilized (GPU utilization at 23%) because there are only three running jobs submitted by a single team (`team-a`).

In the following section, we show how to create multiple teams in the Projects tab and allocate GPUs for each team to maximize cluster usage and manage resources when there are many users per cluster. The test scenarios mimic enterprise environments in which memory and GPU resources are shared among training, inferencing, and interactive workloads.

Next: Creating Projects for Data Science Teams and Allocating GPUs

## Creating Projects for Data Science Teams and Allocating GPUs

Researchers can submit workloads through the Run:AI CLI, Kubeflow, or similar processes. To streamline resource allocation and create prioritization, Run:AI introduces the concept of Projects. Projects are quota entities that associate a project name with GPU allocation and preferences. It is a simple and convenient way to manage multiple data science teams.

A researcher submitting a workload must associate a project with a workload request. The Run:AI scheduler compares the request against the current allocations and the project and determines whether the workload can

be allocated resources or whether it should remain in a pending state.

As a system administrator, you can set the following parameters in the Run:AI Projects tab:

- **Model projects.** Set a project per user, set a project per team of users, and set a project per a real organizational project.
- **Project quotas.** Each project is associated with a quota of GPUs that can be allocated for this project at the same time. This is a guaranteed quota in the sense that researchers using this project are guaranteed to get this number of GPUs no matter what the status in the cluster is. As a rule, the sum of the project allocation should be equal to the number of GPUs in the cluster. Beyond that, a user of this project can receive an over-quota. As long as GPUs are unused, a researcher using this project can get more GPUs. We demonstrate over-quota testing scenarios and fairness considerations in [Achieving High Cluster Utilization with Over-Quota GPU Allocation](#), [Basic Resource Allocation Fairness](#), and [Over-Quota Fairness](#).
- Create a new project, update an existing project, and delete an existing project.
- **Limit jobs to run on specific node groups.** You can assign specific projects to run only on specific nodes. This is useful when the project team needs specialized hardware, for example, with enough memory. Alternatively, a project team might be the owner of specific hardware that was acquired with a specialized budget, or when you might need to direct build or interactive workloads to work on weaker hardware and direct longer training or unattended workloads to faster nodes. For commands to group nodes and set affinity for a specific project, see the [Run:AI Documentation](#).
- **Limit the duration of interactive jobs.** Researchers frequently forget to close interactive jobs. This might lead to a waste of resources. Some organizations prefer to limit the duration of interactive jobs and close them automatically.

The following figure shows the Projects view with four teams created. Each team is assigned a different number of GPUs to account for different workloads, with the total number of GPUs equal to that of the total available GPUs in a cluster consisting of two DGX-1s.

Project Name	Assigned GPUs	Created	Training Node Affinity	Interactive Node Affinity
team-a	2	07/27/20, 9:28AM	none	none
team-b	4	07/28/20, 7:50AM	none	none
team-c	2	07/28/20, 7:50AM	none	none
team-d	8	07/28/20, 7:51AM	none	none

[Next: Submitting Jobs in Run AI CLI](#)

### Submitting Jobs in Run:AI CLI

This section provides the detail on basic Run:AI commands that you can use to run any Kubernetes job. It is divided into three parts according to workload type. AI/ML/DL workloads can be divided into two generic types:

- **Unattended training sessions.** With these types of workloads, the data scientist prepares a self-running workload and sends it for execution. During the execution, the customer can examine the results. This type of workload is often used in production or when model development is at a stage where no human intervention is required.

- **Interactive build sessions.** With these types of workloads, the data scientist opens an interactive session with Bash, Jupyter Notebook, remote PyCharm, or similar IDEs and accesses GPU resources directly. We include a third scenario for running interactive workloads with connected ports to reveal an internal port to the container user..

## Unattended Training Workloads

After setting up projects and allocating GPU(s), you can run any Kubernetes workload using the following command at the command line:

```
$ runai project set team-a runai submit hyper1 -i gcr.io/run-ai-demo/quickstart -g 1
```

This command starts an unattended training job for team-a with an allocation of a single GPU. The job is based on a sample docker image, [gcr.io/run-ai-demo/quickstart](https://gcr.io/run-ai-demo/quickstart). We named the job `hyper1`. You can then monitor the job's progress by running the following command:

```
$ runai list
```

The following figure shows the result of the `runai list` command. Typical statuses you might see include the following:

- `ContainerCreating`. The docker container is being downloaded from the cloud repository.
- `Pending`. The job is waiting to be scheduled.
- `Running`. The job is running.

```
You can run "runai get hyper1 -p team-a" to check the job status
~> runai list
Showing jobs for project team-a
NAME    STATUS   AGE     NODE          IMAGE                               TYPE      PROJECT  USER   GPUs
hyper1  Running  11s   gke-dev-yaron1-gpu-4-pool-154f511d-5nk5  gcr.io/run-ai-demo/quickstart  Train    team-a  yaron  1
```

To get an additional status on your job, run the following command:

```
$ runai get hyper1
```

To view the logs of the job, run the `runai logs <job-name>` command:

```
$ runai logs hyper1
```

In this example, you should see the log of a running DL session, including the current training epoch, ETA, loss function value, accuracy, and time elapsed for each step.

You can view the cluster status on the Run:AI UI at <https://app.run.ai/>. Under Dashboards > Overview, you can monitor GPU utilization.

To stop this workload, run the following command:

```
$ runai delte hyper1
```

This command stops the training workload. You can verify this action by running `runai list` again. For more detail, see [launching unattended training workloads](#).

## Interactive Build Workloads

After setting up projects and allocating GPU(s) you can run an interactive build workload using the following command at the command line:

```
$ runai submit build1 -i python -g 1 --interactive --command sleep --args infinity
```

The job is based on a sample docker image python. We named the job build1.



The `--interactive` flag means that the job does not have a start or end. It is the researcher's responsibility to close the job. The administrator can define a time limit for interactive jobs after which they are terminated by the system.

The `--g 1` flag allocates a single GPU to this job. The command and argument provided is `--command sleep--args infinity`. You must provide a command, or the container starts and then exits immediately.

The following commands work similarly to the commands described in [Unattended Training Workloads](#):

- `runai list`: Shows the name, status, age, node, image, project, user, and GPUs for jobs.
- `runai get build1`: Displays additional status on the job build1.
- `runai delete build1`: Stops the interactive workload build1. To get a bash shell to the container, the following command:

```
$ runai bash build1
```

This provides a direct shell into the computer. Data scientists can then develop or finetune their models within the container.

You can view the cluster status on the Run:AI UI at <https://app.run.ai>. For more detail, see [starting and using interactive build workloads](#).

## Interactive Workloads with Connected Ports

As an extension of interactive build workloads, you can reveal internal ports to the container user when starting a container with the Run:AI CLI. This is useful for cloud environments, working with Jupyter Notebooks, or connecting to other microservices. [Ingress](#) allows access to Kubernetes services from outside the Kubernetes cluster. You can configure access by creating a collection of rules that define which inbound connections reach which services.

For better management of external access to the services in a cluster, we suggest that cluster administrators install [Ingress](#) and configure LoadBalancer.

To use Ingress as a service type, run the following command to set the method type and the ports when submitting your workload:

```
$ runai submit test-ingress -i jupyter/base-notebook -g 1 \
--interactive --service-type=ingress --port 8888 \
--args="--NotebookApp.base_url=test-ingress" --command=start-notebook.sh
```

After the container starts successfully, execute `runai list` to see the `SERVICE URL(S)` with which to access the Jupyter Notebook. The URL is composed of the ingress endpoint, the job name, and the port. For example, see <https://10.255.174.13/test-ingress-8888>.

For more details, see [launching an interactive build workload with connected ports](#).

## Next: Achieving High Cluster Utilization

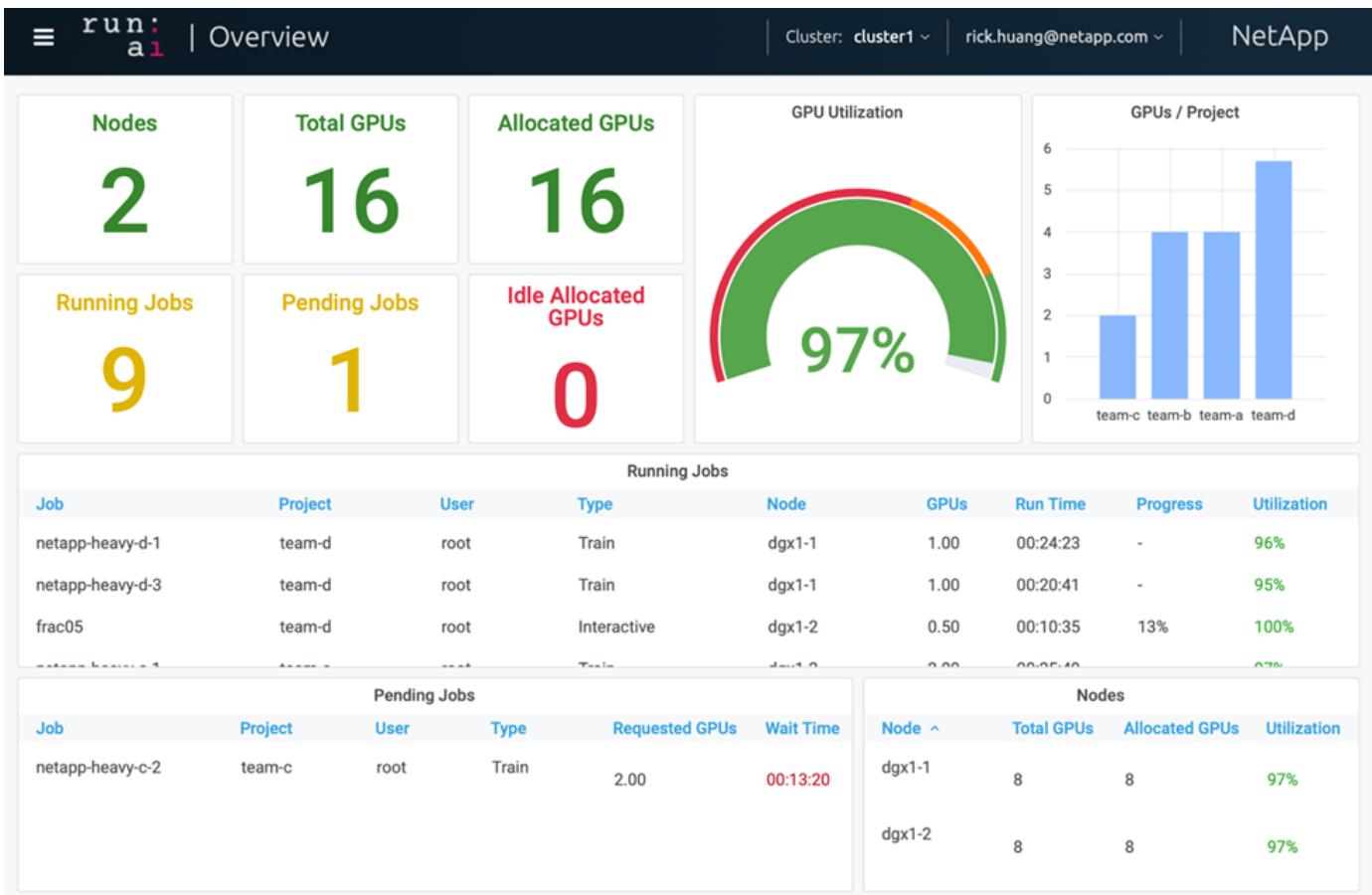
### Achieving High Cluster Utilization

In this section, we emulate a realistic scenario in which four data science teams each submit their own workloads to demonstrate the Run:AI orchestration solution that achieves high cluster utilization while maintaining prioritization and balancing GPU resources. We start by using the ResNet-50 benchmark described in the section [ResNet-50 with ImageNet Dataset Benchmark Summary](#):

```
$ runai submit netapp1 -i netapp/tensorflow-tf1-py3:20.01.0 --local-image
--large-shm -v /mnt:/mnt -v /tmp:/tmp --command python --args
"/netapp/scripts/run.py" --args "--
dataset_dir=/mnt/mount_0/dataset/imagenet/imagenet_original/" --args "--
num_mounts=2" --args "--dgx_version=dgx1" --args "--num_devices=1" -g 1
```

We ran the same ResNet-50 benchmark as in [NVA-1121](#). We used the flag `--local-image` for containers not residing in the public docker repository. We mounted the directories `/mnt` and `/tmp` on the host DGX-1 node to `/mnt` and `/tmp` to the container, respectively. The dataset is at NetApp AFFA800 with the `dataset_dir` argument pointing to the directory. Both `--num_devices=1` and `-g 1` mean that we allocate one GPU for this job. The former is an argument for the `run.py` script, while the latter is a flag for the `runai submit` command.

The following figure shows a system overview dashboard with 97% GPU utilization and all sixteen available GPUs allocated. You can easily see how many GPUs are allocated for each team in the GPUs/Project bar chart. The Running Jobs pane shows the current running job names, project, user, type, node, GPUs consumed, run time, progress, and utilization details. A list of workloads in queue with their wait time is shown in Pending Jobs. Finally, the Nodes box offers GPU numbers and utilization for individual DGX-1 nodes in the cluster.



[Next: Fractional GPU Allocation for Less Demanding or Interactive Workloads](#)

### Fractional GPU Allocation for Less Demanding or Interactive Workloads

When researchers and developers are working on their models, whether in the development, hyperparameter tuning, or debugging stages, such workloads usually require fewer computational resources. It is therefore more efficient to provision fractional GPU and memory such that the same GPU can simultaneously be allocated to other workloads. Run:AI's orchestration solution provides a fractional GPU sharing system for containerized workloads on Kubernetes. The system supports workloads running CUDA programs and is especially suited for lightweight AI tasks such as inference and model building. The fractional GPU system transparently gives data science and AI engineering teams the ability to run multiple workloads simultaneously on a single GPU. This enables companies to run more workloads, such as computer vision, voice recognition, and natural language processing on the same hardware, thus lowering costs.

Run:AI's fractional GPU system effectively creates virtualized logical GPUs with their own memory and computing space that containers can use and access as if they were self-contained processors. This enables several workloads to run in containers side-by-side on the same GPU without interfering with each other. The solution is transparent, simple, and portable and it requires no changes to the containers themselves.

A typical usecase could see two to eight jobs running on the same GPU, meaning that you could do eight times the work with the same hardware.

For the job `frac05` belonging to project `team-d` in the following figure, we can see that the number of GPUs allocated was 0.50. This is further verified by the `nvidia-smi` command, which shows that the GPU memory available to the container was 16,255MB: half of the 32GB per V100 GPU in the DGX-1 node.

```

root@run-deploy:~# runai bash frac05 -p team-d
root@frac05-0:/workload# nvidia-smi
Tue Jul 28 15:17:03 2020
+-----+
| NVIDIA-SMI 450.51.05    Driver Version: 450.51.05    CUDA Version: 11.0    |
|-----+-----+-----+
| GPU  Name      Persistence-MI Bus-Id      Disp.A  Volatile Uncorr. ECC  |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M.  |
|                   |          |           |          |          |          MIG M. |
|=====+=====+=====+=====+=====+=====+=====+=====+=====|
|  0  Tesla V100-SXM2... On   | 00000000:07:00.0 Off |          0 | | |
| N/A  57C    P0    240W / 300W | 15525MiB / 16255MiB | 100%     Default |
|                   |          |           |          |          N/A |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:
| GPU  GI  CI      PID  Type  Process name          GPU Memory  |
|       ID  ID
|=====+=====+=====+=====+=====+=====+=====+=====|
|  0  N/A  N/A      156   C    python3            15525MiB  |
+-----+

```

[Next: Achieving High Cluster Utilization with Over-Quota GPU Allocation](#)

#### Achieving High Cluster Utilization with Over-Quota GPU Allocation

In this section and in the sections [Basic Resource Allocation Fairness](#), and [Over-Quota Fairness](#), we have devised advanced testing scenarios to demonstrate the Run:AI orchestration capabilities for complex workload management, automatic preemptive scheduling, and over-quota GPU provisioning. We did this to achieve high cluster-resource usage and optimize enterprise-level data science team productivity in an ONTAP AI environment.

For these three sections, set the following projects and quotas:

Project	Quota
team-a	4
team-b	2
team-c	2
team-d	8

In addition, we use the following containers for these three sections:

- Jupyter Notebook: [jupyter/base-notebook](#)
- Run:AI quickstart: [gcr.io/run-ai-demo/quickstart](#)

We set the following goals for this test scenario:

- Show the simplicity of resource provisioning and how resources are abstracted from users
- Show how users can easily provision fractions of a GPU and integer number of GPUs
- Show how the system eliminates compute bottlenecks by allowing teams or users to go over their resource quota if there are free GPUs in the cluster
- Show how data pipeline bottlenecks are eliminated by using the NetApp solution when running compute-intensive jobs, such as the NetApp container
- Show how multiple types of containers are running using the system
  - Jupyter Notebook
  - Run:AI container
- Show high utilization when the cluster is full

For details on the actual command sequence executed during the testing, see [Testing Details for Section 4.8](#).

When all 13 workloads are submitted, you can see a list of container names and GPUs allocated, as shown in the following figure. We have seven training and six interactive jobs, simulating four data science teams, each with their own models running or in development. For interactive jobs, individual developers are using Jupyter Notebooks to write or debug their code. Thus, it is suitable to provision GPU fractions without using too many cluster resources.

NAME	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUS	CREATED BY CLI	SERVICE URL(S)
b-4-gg	Running	2m	dgx1-2	gcr.io/run-ai-demo/quickstart	Train	team-b	root	2	true	
c-5-g	Running	2m	dgx1-2	gcr.io/run-ai-demo/quickstart	Train	team-c	root	1	true	
c-4-gg	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-c	root	2	true	
b-3-g	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-b	root	1	true	
c-3-g02	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Interactive	team-c	root	0.2	true	
d-1-gggg	Running	2m	dgx1-2	gcr.io/run-ai-demo/quickstart	Train	team-d	root	4	true	
c-2-g03	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Interactive	team-c	root	0.3	true	
c-1-g05	Running	2m	dgx1-1	gcr.io/run-ai-demo/quickstart	Interactive	team-c	root	0.5	true	
a-2-gg	Running	3m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-a	root	2	true	
b-2-g04	Running	3m	dgx1-2	gcr.io/run-ai-demo/quickstart	Interactive	team-b	root	0.4	true	
a-1-g	Running	3m	dgx1-1	gcr.io/run-ai-demo/quickstart	Train	team-a	root	1	true	
b-1-g06	Running	3m	dgx1-2	gcr.io/run-ai-demo/quickstart	Interactive	team-b	root	0.6	true	
a-1-1-jupyter	Running	3m	dgx1-1	jupyter/base-notebook	Interactive	team-a	root	1	true	<a href="http://10.61.218.134/a-1-1-jupyter">http://10.61.218.134/a-1-1-jupyter</a> , <a href="https://10.61.218.134/a-1-1-jupyter">https://10.61.218.134/a-1-1-jupyter</a>

The results of this testing scenario show the following:

- The cluster should be full: 16/16 GPUs are used.
- High cluster utilization.
- More experiments than GPUs due to fractional allocation.
- `team-d` is not using all their quota; therefore, `team-b` and `team-c` can use additional GPUs for their experiments, leading to faster time to innovation.

[Next: Basic Resource Allocation Fairness](#)

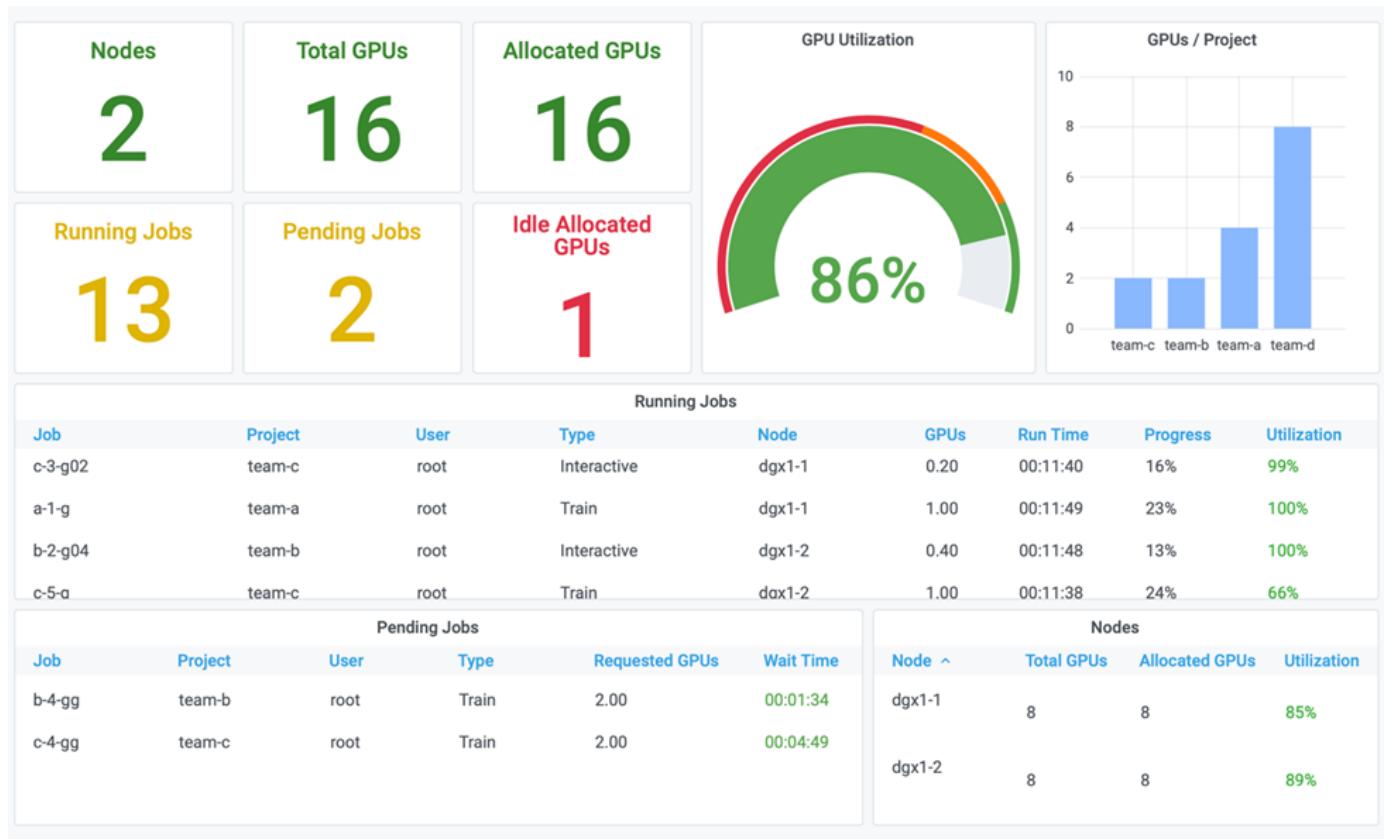
### Basic Resource Allocation Fairness

In this section, we show that, when `team-d` asks for more GPUs (they are under their quota), the system pauses the workloads of `team-b` and `team-c` and moves them into a pending state in a fair-share manner.

For details including job submissions, container images used, and command sequences executed, see the section [Testing Details for Section 4.9](#).

The following figure shows the resulting cluster utilization, GPUs allocated per team, and pending jobs due to automatic load balancing and preemptive scheduling. We can observe that when the total number of GPUs

requested by all team workloads exceeds the total available GPUs in the cluster, Run:AI's internal fairness algorithm pauses one job each for `team-b` and `team-c` because they have met their project quota. This provides overall high cluster utilization while data science teams still work under resource constraints set by an administrator.



The results of this testing scenario demonstrate the following:

- Automatic load balancing.** The system automatically balances the quota of the GPUs, such that each team is now using their quota. The workloads that were paused belong to teams that were over their quota.
- Fair share pause.** The system chooses to stop the workload of one team that was over their quota and then stop the workload of the other team. Run:AI has internal fairness algorithms.

Next: Over-Quota Fairness

### Over-Quota Fairness

In this section, we expand the scenario in which multiple teams submit workloads and exceed their quota. In this way, we demonstrate how Run:AI's fairness algorithm allocates cluster resources according to the ratio of preset quotas.

Goals for this test scenario:

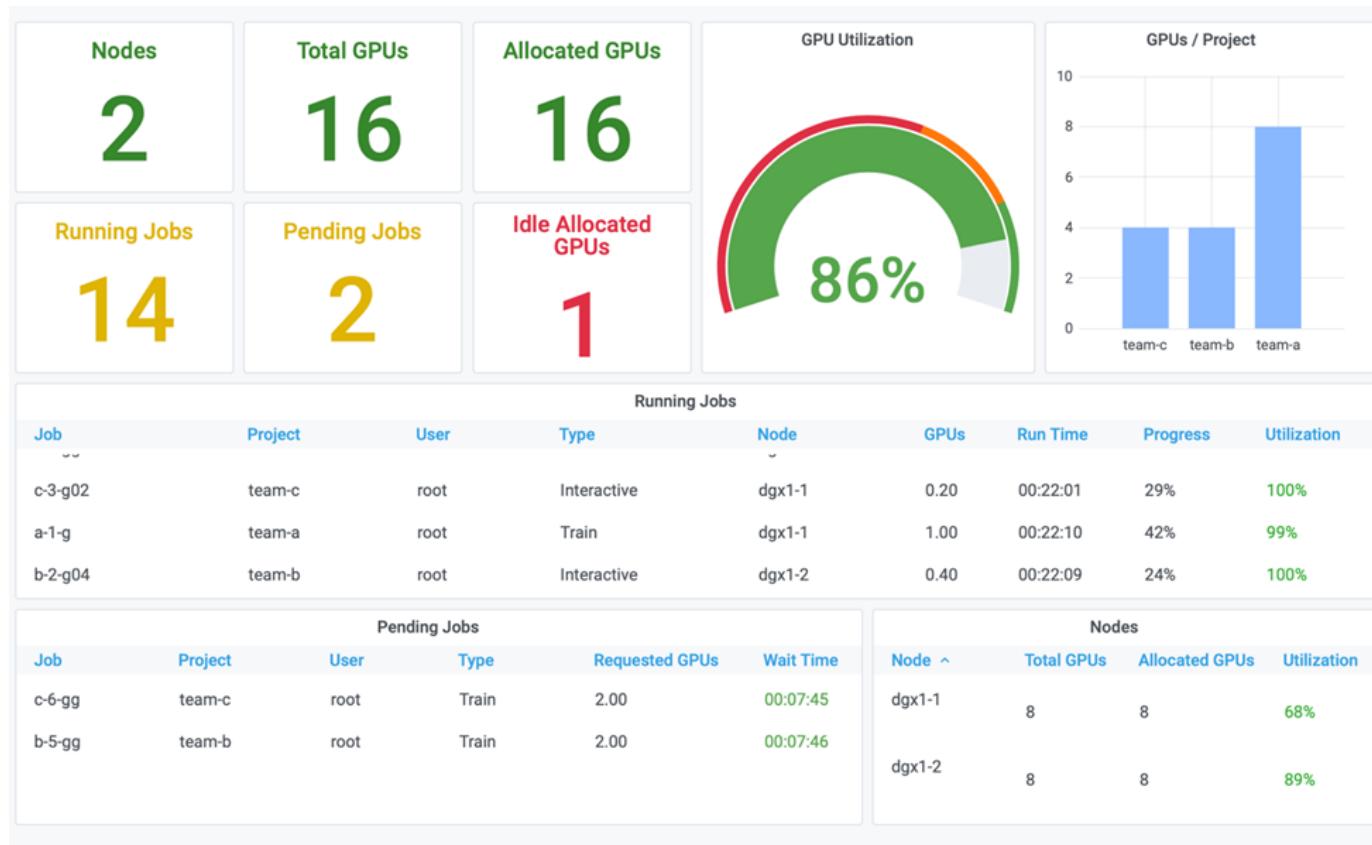
- Show queuing mechanism when multiple teams are requesting GPUs over their quota.
- Show how the system distributes a fair share of the cluster between multiple teams that are over their quota according to the ratio between their quotas, so that the team with the larger quota gets a larger share of the spare capacity.

At the end of [Basic Resource Allocation Fairness](#), there are two workloads queued: one for `team-b` and one

for `team-c`. In this section, we queue additional workloads.

For details including job submissions, container images used, and command sequences executed, see [Testing Details for section 4.10](#).

When all jobs are submitted according to the section [Testing Details for section 4.10](#), the system dashboard shows that `team-a`, `team-b`, and `team-c` all have more GPUs than their preset quota. `team-a` occupies four more GPUs than its preset soft quota (four), whereas `team-b` and `team-c` each occupy two more GPUs than their soft quota (two). The ratio of over-quota GPUs allocated is equal to that of their preset quota. This is because the system used the preset quota as a reference of priority and provisioned accordingly when multiple teams request more GPUs, exceeding their quota. Such automatic load balancing provides fairness and prioritization when enterprise data science teams are actively engaged in AI model development and production.



The results of this testing scenario show the following:

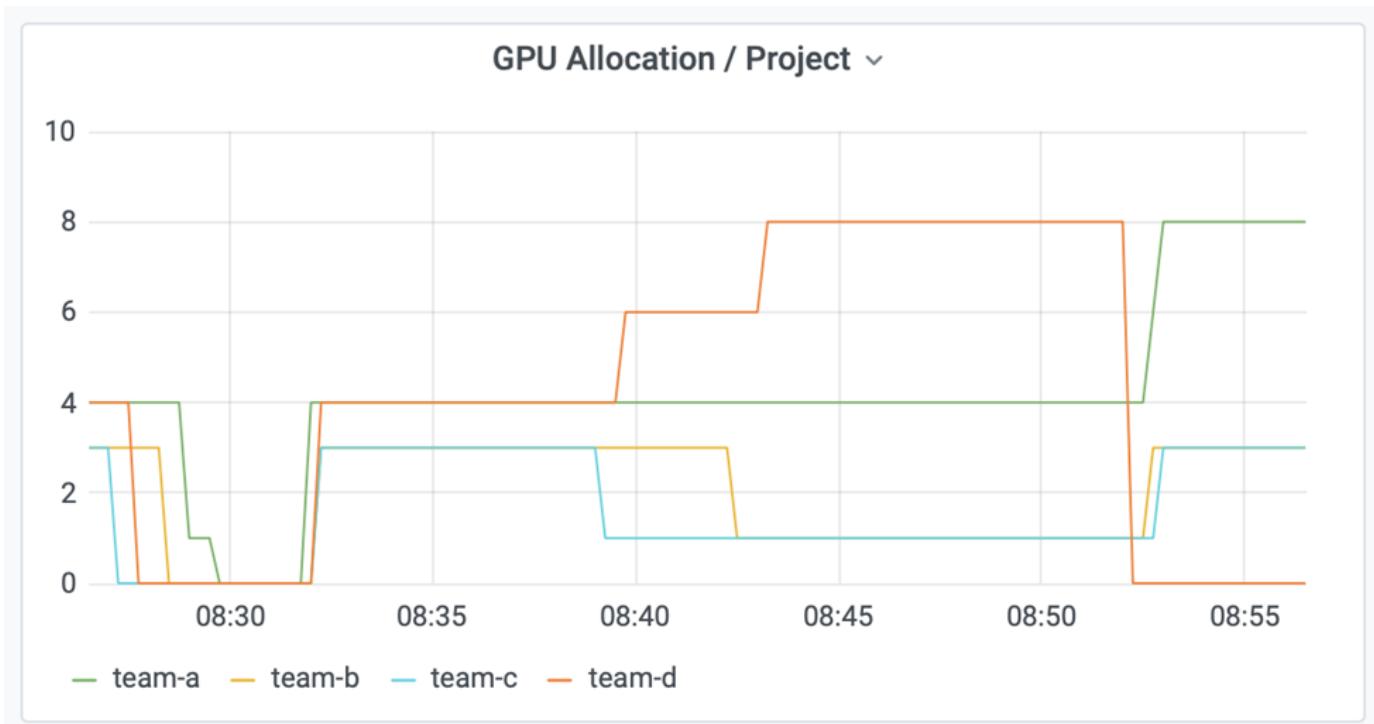
- The system starts to de-queue the workloads of other teams.
- The order of the dequeuing is decided according to fairness algorithms, such that `team-b` and `team-c` get the same amount of over-quota GPUs (since they have a similar quota), and `team-a` gets a double amount of GPUs since their quota is two times higher than the quota of `team-b` and `team-c`.
- All the allocation is done automatically.

Therefore, the system should stabilize on the following states:

Project	GPUs allocated	Comment
team-a	8/4	Four GPUs over the quota. Empty queue.

Project	GPUs allocated	Comment
team-b	4/2	Two GPUs over the quota. One workload queued.
team-c	4/2	Two GPUs over the quota. One workload queued.
team-d	0/8	Not using GPUs at all, no queued workloads.

The following figure shows the GPU allocation per project over time in the Run:AI Analytics dashboard for the sections [Achieving High Cluster Utilization with Over-Quota GPU Allocation](#), [Basic Resource Allocation Fairness](#), and [Over-Quota Fairness](#). Each line in the figure indicates the number of GPUs provisioned for a given data science team at any time. We can see that the system dynamically allocates GPUs according to workloads submitted. This allows teams to go over quota when there are available GPUs in the cluster, and then preempt jobs according to fairness, before finally reaching a stable state for all four teams.



Next: [Saving Data to a Trident-Provisioned PersistentVolume](#)

### Saving Data to a Trident-Provisioned PersistentVolume

NetApp Trident is a fully supported open source project designed to help you meet the sophisticated persistence demands of your containerized applications. You can read and write data to a Trident-provisioned Kubernetes PersistentVolume (PV) with the added benefit of data tiering, encryption, NetApp Snapshot technology, compliance, and high performance offered by NetApp ONTAP data management software.

### Reusing PVCs in an Existing Namespace

For larger AI projects, it might be more efficient for different containers to read and write data to the same Kubernetes PV. To reuse a Kubernetes Persistent Volume Claim (PVC), the user must have already created a PVC. See the [NetApp Trident documentation](#) for details on creating a PVC. Here is an example of reusing an existing PVC:

```
$ runai submit pvc-test -p team-a --pvc test:/tmp/pvc1mount -i gcr.io/run-ai-demo/quickstart -g 1
```

Run the following command to see the status of job `pvc-test` for project `team-a`:

```
$ runai get pvc-test -p team-a
```

You should see the PV `/tmp/pvc1mount` mounted to `team-a` job `pvc-test`. In this way, multiple containers can read from the same volume, which is useful when there are multiple competing models in development or in production. Data scientists can build an ensemble of models and then combine prediction results by majority voting or other techniques.

Use the following to access the container shell:

```
$ runai bash pvc-test -p team-a
```

You can then check the mounted volume and access your data within the container.

This capability of reusing PVCs works with NetApp FlexVol volumes and NetApp ONTAP FlexGroup volumes, enabling data engineers more flexible and robust data management options to leverage your data fabric powered by NetApp.

[Next: Conclusion](#)

## Conclusion

NetApp and Run:AI have partnered in this technical report to demonstrate the unique capabilities of the NetApp ONTAP AI solution together with the Run:AI Platform for simplifying orchestration of AI workloads. The preceding steps provide a reference architecture to streamline the process of data pipelines and workload orchestration for deep learning. Customers looking to implement these solutions are encouraged to reach out to NetApp and Run:AI for more information.

[Next: Testing Details for Section 4.8](#)

## Testing Details for Section 4.8

This section contains the testing details for the section [Achieving High Cluster Utilization with Over-Quota GPU Allocation](#).

Submit jobs in the following order:

Project	Image	# GPUs	Total	Comment
team-a	Jupyter	1	1/4	—
team-a	NetApp	1	2/4	—
team-a	Run:AI	2	4/4	Using all their quota
team-b	Run:AI	0.6	0.6/2	Fractional GPU

Project	Image	# GPUs	Total	Comment
team-b	Run:AI	0.4	1/2	Fractional GPU
team-b	NetApp	1	2/2	-
team-b	NetApp	2	4/2	Two over quota
team-c	Run:AI	0.5	0.5/2	Fractional GPU
team-c	Run:AI	0.3	0.8/2	Fractional GPU
team-c	Run:AI	0.2	1/2	Fractional GPU
team-c	NetApp	2	3/2	One over quota
team-c	NetApp	1	4/2	Two over quota
team-d	NetApp	4	4/8	Using half of their quota

Command structure:

```
$ runai submit <job-name> -p <project-name> -g <#GPUs> -i <image-name>
```

Actual command sequence used in testing:

```
$ runai submit a-1-1-jupyter -i jupyter/base-notebook -g 1 \
--interactive --service-type=ingress --port 8888 \
--args="--NotebookApp.base_url=team-a-test-ingress" --command=start
-notebook.sh -p team-a
$ runai submit a-1-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-a
$ runai submit a-2-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a
$ runai submit b-1-g06 -i gcr.io/run-ai-demo/quickstart -g 0.6
--interactive -p team-b
$ runai submit b-2-g04 -i gcr.io/run-ai-demo/quickstart -g 0.4
--interactive -p team-b
$ runai submit b-3-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-b
$ runai submit b-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-b
$ runai submit c-1-g05 -i gcr.io/run-ai-demo/quickstart -g 0.5
--interactive -p team-c
$ runai submit c-2-g03 -i gcr.io/run-ai-demo/quickstart -g 0.3
--interactive -p team-c
$ runai submit c-3-g02 -i gcr.io/run-ai-demo/quickstart -g 0.2
--interactive -p team-c
$ runai submit c-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-c
$ runai submit c-5-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-c
$ runai submit d-1-gggg -i gcr.io/run-ai-demo/quickstart -g 4 -p team-d
```

At this point, you should have the following states:

Project	GPUs Allocated	Workloads Queued
team-a	4/4 (soft quota/actual allocation)	None
team-b	4/2	None
team-c	4/2	None
team-d	4/8	None

See the section [Achieving High Cluster Utilization with Over-quota GPU Allocation](#) for discussions on the proceeding testing scenario.

[Next: Testing Details for Section 4.9](#)

### Testing Details for Section 4.9

This section contains testing details for the section [Basic Resource Allocation Fairness](#).

Submit jobs in the following order:

Project	# GPUs	Total	Comment
team-d	2	6/8	Team-b/c workload pauses and moves to <a href="#">pending</a> .
team-d	2	8/8	Other team (b/c) workloads pause and move to <a href="#">pending</a> .

See the following executed command sequence:

```
$ runai submit d-2-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-d
$ runai submit d-3-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-d
```

At this point, you should have the following states:

Project	GPUs Allocated	Workloads Queued
team-a	4/4	None
team-b	2/2	None
team-c	2/2	None
team-d	8/8	None

See the section [Basic Resource Allocation Fairness](#) for a discussion on the proceeding testing scenario.

[Next: Testing Details for Section 4.10](#)

### Testing Details for Section 4.10

This section contains testing details for the section [Over-Quota Fairness](#).

Submit jobs in the following order for `team-a`, `team-b`, and `team-c`:

Project	# GPUs	Total	Comment
team-a	2	4/4	1 workload queued
team-a	2	4/4	2 workloads queued
team-b	2	2/2	2 workloads queued
team-c	2	2/2	2 workloads queued

See the following executed command sequence:

```
$ runai submit a-3-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a$ runai submit a-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a$ runai submit b-5-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-b$ runai submit c-6-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-c
```

At this point, you should have the following states:

Project	GPUs Allocated	Workloads Queued
team-a	4/4	Two workloads asking for GPUs two each
team-b	2/2	Two workloads asking for two GPUs each
team-c	2/2	Two workloads asking for two GPUs each
team-d	8/8	None

Next, delete all the workloads for `team-d`:

```
$ runai delete -p team-d d-1-gggg d-2-gg d-3-gg
```

See the section [Over-Quota Fairness](#), for discussions on the proceeding testing scenario.

[Next: Where to Find Additional Information](#)

## Where to Find Additional Information

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

- NVIDIA DGX Systems
  - NVIDIA DGX-1 System  
<https://www.nvidia.com/en-us/data-center/dgx-1/>
  - NVIDIA V100 Tensor Core GPU  
<https://www.nvidia.com/en-us/data-center/tesla-v100/>

- NVIDIA NGC  
<https://www.nvidia.com/en-us/gpu-cloud/>
- Run:AI container orchestration solution
  - Run:AI product introduction  
<https://docs.run.ai/home/components/>
  - Run:AI installation documentation  
<https://docs.run.ai/Administrator/Cluster-Setup/Installing-Run-AI-on-an-on-premise-Kubernetes-Cluster/>  
<https://docs.run.ai/Administrator/Researcher-Setup/Installing-the-Run-AI-Command-Line-Interface/>
  - Submitting jobs in Run:AI CLI  
<https://docs.run.ai/Researcher/Walkthroughs/Walkthrough-Launch-Unattended-Training-Workloads-/>  
<https://docs.run.ai/Researcher/Walkthroughs/Walkthrough-Start-and-Use-Interactive-Build-Workloads-/>
  - Allocating GPU fractions in Run:AI CLI  
<https://docs.run.ai/Researcher/Walkthroughs/Walkthrough-Using-GPU-Fractions/>
- NetApp AI Control Plane
  - Technical report  
<https://www.netapp.com/us/media/tr-4798.pdf>
  - Short-form demo  
[https://youtu.be/gfr\\_sO27Rvo](https://youtu.be/gfr_sO27Rvo)
  - GitHub repository  
[https://github.com/NetApp/kubeflow\\_jupyter\\_pipeline](https://github.com/NetApp/kubeflow_jupyter_pipeline)
- NetApp AFF systems
  - NetApp AFF A-Series Datasheet  
<https://www.netapp.com/us/media/ds-3582.pdf>
  - NetApp Flash Advantage for All Flash FAS  
<https://www.netapp.com/us/media/ds-3733.pdf>
  - ONTAP 9 Information Library  
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>
  - NetApp ONTAP FlexGroup Volumes technical report  
<https://www.netapp.com/us/media/tr-4557.pdf>
- NetApp ONTAP AI
  - ONTAP AI with DGX-1 and Cisco Networking Design Guide  
<https://www.netapp.com/us/media/nva-1121-design.pdf>
  - ONTAP AI with DGX-1 and Cisco Networking Deployment Guide  
<https://www.netapp.com/us/media/nva-1121-deploy.pdf>
  - ONTAP AI with DGX-1 and Mellanox Networking Design Guide  
<http://www.netapp.com/us/media/nva-1138-design.pdf>
  - ONTAP AI with DGX-2 Design Guide  
<https://www.netapp.com/us/media/nva-1135-design.pdf>

# Modern Data Analytics

# **Hybrid Cloud / Virtualization**

**Get Started With NetApp & VMware**

VMware on NetApp: Your journey starts here!

► <https://d3cy9zhslanhfa.cloudfla...>

If you're ready to start transforming your VMware environment, browse the latest solution overview, review our latest technical solutions and product demonstrations. If you're ready for the next step, engage NetApp and VMware community of experts to help plan and execute your data center modernization, hybrid cloud or containerized application initiatives.

Not sure where to start? [Contact](#) a member of the VMware Experts at NetApp.

## Learn about NetApp and VMware Solutions

- [NetApp & VMware: Better Together](#)
- [ONTAP 9.8 Latest Features for VMware Overview](#)
- [Leveraging SnapCenter Plugin for VMware vSphere](#)
- [Redefining VMware Performance with NetApp and NVMe](#)
- [A Low-Cost Performant World for VMware Cloud on AWS](#)
- [Introducing VMware Tanzu with NetApp](#)

[ront.net/media/D30CEDFE-](#)

- [Virtual Desktop Infrastructure \(VDI\): Delivering Employee Workstations on Demand](#)
- [VMware on AWS: Architecture and Service Options](#)
- [Programming with NetApp Cloud Volumes Service APIs To Optimize AWS Experience](#)
- [Kubernetes: Running K8s on vSphere and Tanzu](#)

## Build Your Virtualized Data Fabric

### Review our latest NetApp Solutions for VMware

- [VMware vSphere with ONTAP : NetApp Solutions](#)
- [VMware vSphere Virtual Volumes with ONTAP](#)
- [SnapCenter Plug-in for VMware vSphere](#)
- [NetApp Modern NVMeoF VMware vSphere Workload Design & Validation](#)
- [NetApp Modern NVMeoF Cloud-Connected Flash Solution for VMware & SQL Server](#)
- [Accelerate Your Kubernetes Journey with VMware Tanzu & ONTAP](#)
- [Lower The Cost of Running VMware Cloud on AWS](#)

### Explore video demonstrations of the latest VMware solutions

- [Best Practices for VMware vSphere and NetApp ONTAP](#)
- [Your VMware Environment - Let's Run it on NVMe-oF with ONTAP](#)
- [vVols Disaster Recovery with ONTAP Tools and VMware SRM](#)
- [Provisioning and Managing FlexGroup Datastores with ONTAP Tools](#)
- [NetApp NFS VAAI Plugin Update](#)
- [Scale-Out Virtual Desktops with NetApp ONTAP FlexGroup](#)
- [VMware Backup and Recovery for the Data Fabric](#)
- [Easier Data Protection with SnapCenter Plug-in for VMware vSphere](#)

## Deploy flexible hybrid-cloud & modernized applications infrastructure for VMware

### Videos

- [Architecting VMware Datastores on NetApp All Flash FAS](#)
- [Let's Automate - Build Your VMware Cloud with ONTAP](#)
- [A Low-Cost Performant World for VMware Cloud on AWS](#)
- [Migrate Your VMware VMs to Google Cloud](#)



**Deploying Dynamic Persistent  
NetApp Storage for VMware Tanzu,  
part 1**

**Deploying Dynamic Persistent  
NetApp Storage for VMware Tanzu,  
part 2**

**Deploying Dynamic Persistent  
NetApp Storage for VMware Tanzu,  
part 3**

## Blogs

- [VMware Cloud on AWS: How Fujitsu Saves Millions using CVO](#)

## Engage NetApp & VMware Experts

- [Join The VMware Solutions Discussion Forum](#)
- [Contact The NetApp Global Services Team To Get Started](#)

# VMware Virtualization for ONTAP

## NetApp ONTAP for VMware vSphere Administrators

### Introduction to ONTAP for vSphere Administrators

#### Why ONTAP for vSphere?

NetApp ONTAP simplifies storage and data management operations and distinctly complements VMware environments, whether deploying on-premises or to the cloud. NetApp best-in-class data protection, storage efficiency innovations, and outstanding performance in both SAN- and NAS-based VMware architectures are among the reasons why tens of thousands of customers have selected ONTAP as their storage solution for vSphere deployments.

NetApp provides numerous VMware plug-ins, validations, and qualifications of various VMware products to support customers facing the unique challenges of administering a virtualization environment. NetApp does for storage and data management what VMware does for virtualization, allowing customers to focus on their core competencies rather than managing physical storage. This nearly 20-year partnership between VMware and NetApp continues to evolve and add customer value as new technologies, such as VMware Cloud Foundation and Tanzu, emerge, while continuing to support the foundation of vSphere.

Key factors customers value include:

- **Unified storage**
- **Storage efficiency**
- **Virtual volumes and storage policy-based management**
- **Hybrid cloud**

For more information regarding supported NetApp and VMware solutions, see the following resources:

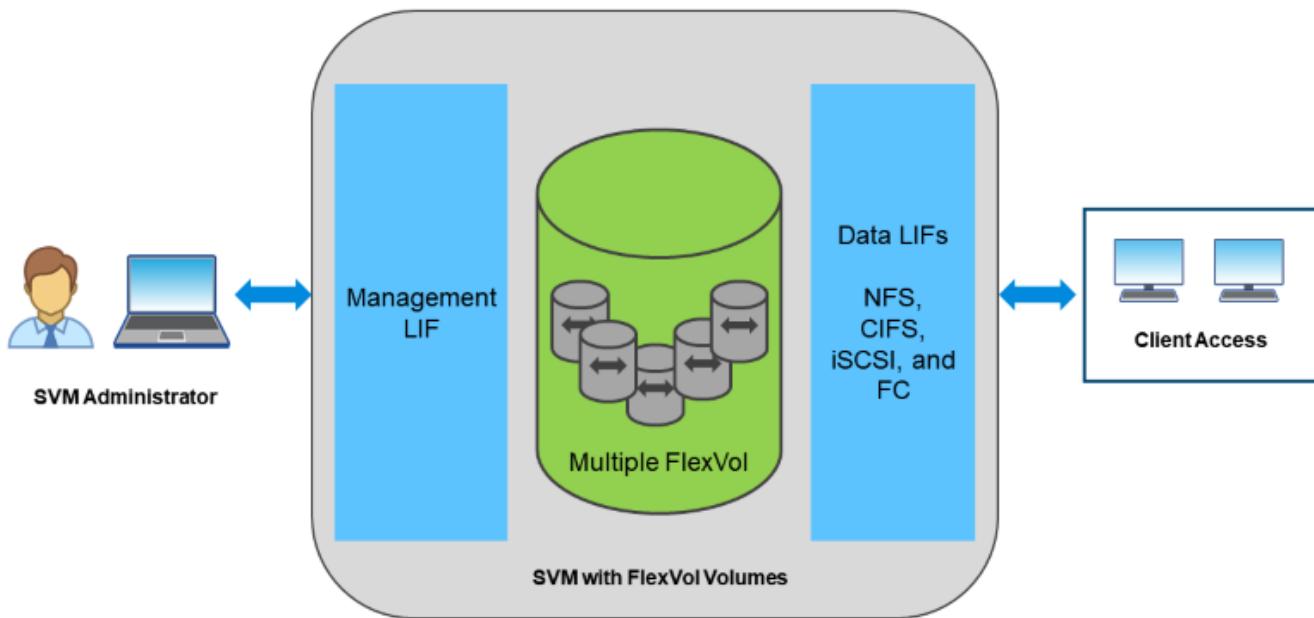
- [The NetApp Interoperability Matrix Tool \(IMT\)](#). The IMT defines the qualified components and versions you can use to build FC/FCoE, iSCSI, NFS and CIFS configurations.
- [The VMware Compatibility Guide](#). The VMware Compatibility guide lists System, I/O, Storage/SAN and Backup compatibility with VMware Infrastructure and software products
- [NetApp ONTAP Tools for VMware](#). ONTAP tools for VMware vSphere is a single vCenter Server plug-in that includes the VSC, VASA Provider, and Storage Replication Adapter (SRA) extensions.

## ONTAP Unified Storage

### About Unified Storage

Systems running ONTAP software are unified in several significant ways. Originally this approach referred to supporting both NAS and SAN protocols on one storage system, and ONTAP continues to be a leading platform for SAN along with its original strength in NAS.

A storage virtual machine (SVM) is a logical construct allowing client access to systems running ONTAP software. SVMs can serve data concurrently through multiple data access protocols via logical interfaces (LIFs). SVMs provide file-level data access through NAS protocols, such as CIFS and NFS, and block-level data access through SAN protocols, such as iSCSI, FC/FCoE, and NVMe. SVMs can serve data to SAN and NAS clients independently at the same time.



In the vSphere world, this approach could also mean a unified system for virtual desktop infrastructure (VDI) together with virtual server infrastructure (VSI). Systems running ONTAP software are typically less expensive for VSI than traditional enterprise arrays and yet have advanced storage efficiency capabilities to handle VDI in the same system. ONTAP also unifies a variety of storage media, from SSDs to SATA, and can extend that easily into the cloud. There's no need to buy one flash array for performance, a SATA array for archives, and separate systems for the cloud. ONTAP ties them all together.

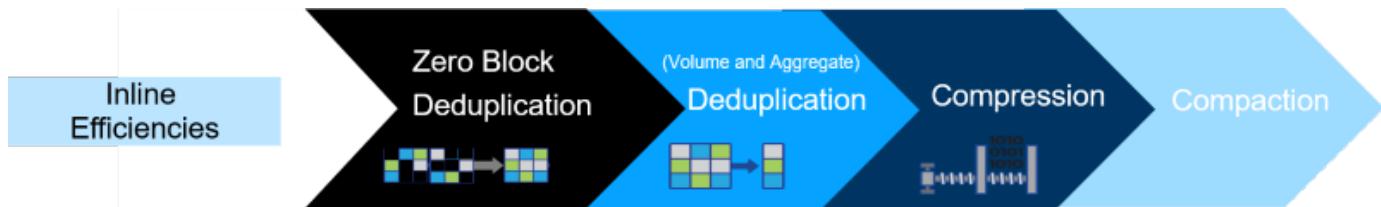


For more information on SVMs, unified storage and client access, see [Storage Virtualization](#) in the ONTAP 9 Documentation center.

### About storage efficiencies

Although NetApp was the first to deliver deduplication for production workloads, this innovation wasn't the first or last one in this area. It started with ONTAP Snapshot copies, a space-efficient data protection mechanism with no performance effect, along with FlexClone technology to instantly make read/write copies of VMs for production and backup use. NetApp went on to deliver inline capabilities, including deduplication, compression, and zero-block deduplication, to squeeze out the most storage from expensive SSDs. Most recently, ONTAP added compaction to strengthen our storage efficiencies.

- **Inline zero-block deduplication.** Eliminates space wasted by all-zero blocks.
- **Inline compression.** Compresses data blocks to reduce the amount of physical storage required.
- **Inline deduplication.** Eliminates incoming blocks with existing blocks on disk.
- **Inline data compaction.** Packs smaller I/O operations and files into each physical block.



You can run deduplication, data compression, and data compaction together or independently to achieve optimal space savings on a FlexVol volume. The combination of these capabilities has resulted in customers seeing savings of up to 5:1 for VSI and up to 30:1 for VDI.



For more information on ONTAP storage efficiencies, see [Using deduplication, data compression, and data compaction to increase storage efficiency](#) in the ONTAP 9 Documentation center.

## Virtual Volumes (vVols) and Storage Policy Based Management (SPBM)

### About vVols and SPBM

NetApp was an early design partner with VMware in the development of vSphere Virtual Volumes (vVols), providing architectural input and early support for vVols and VMware vSphere APIs for Storage Awareness (VASA). Not only did this approach bring VM granular storage management to VMFS, it also supported automation of storage provisioning through Storage Policy-Based Management (SPBM).

SPBM provides a framework that serves as an abstraction layer between the storage services available to your virtualization environment and the provisioned storage elements via policies. This approach allows storage architects to design storage pools with different capabilities that can be easily consumed by VM administrators. Administrators can then match virtual machine workload requirements against the provisioned storage pools, allowing for granular control of various settings on a per-VM or virtual disk level.

ONTAP leads the storage industry in vVols scale, supporting hundreds of thousands of vVols in a single cluster, whereas enterprise array and smaller flash array vendors support as few as several thousand vVols per array. NetApp is also driving the evolution of VM granular management with upcoming capabilities in support of vVols 3.0.



For more information on VMware vSphere Virtual Volumes, SPBM, and ONTAP, see [8585D031CD0682C2/B08AAC](#) [4400: VMware vSphere Virtual Volumes with ONTAP](#).

## Hybrid Cloud with ONTAP and vSphere

### About Hybrid Cloud

Whether used for an on-premises private cloud, public-cloud infrastructure, or a hybrid cloud that combines the best of both, ONTAP solutions help you build your data fabric to streamline and optimize data management. Start with high-performance, all-flash systems, then couple them with either disk or cloud storage systems for data protection and cloud compute.

Choose from Azure, AWS, IBM, or Google clouds to optimize costs and avoid lock-in. Leverage advanced support for OpenStack and container technologies as needed.

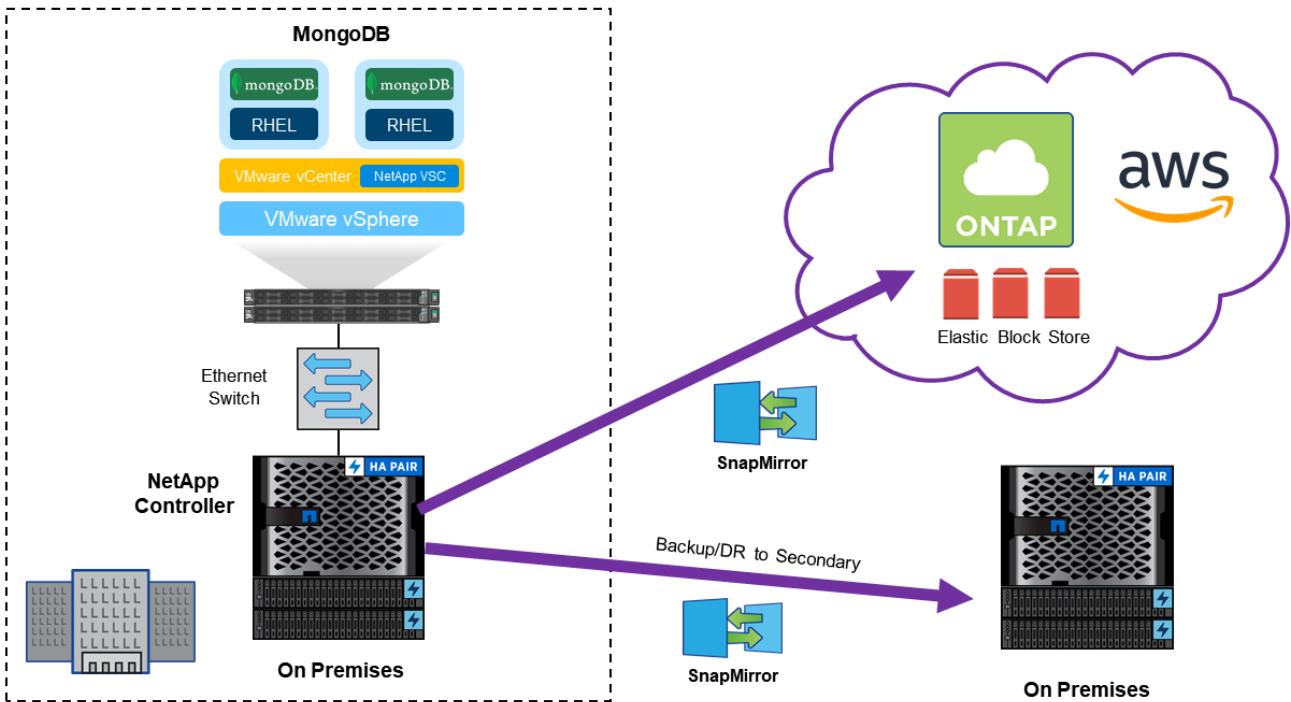
Data protection is often the first thing customers try when they begin their cloud journey. Protection can be as simple as asynchronous replication of key data or as complex as a complete hot-backup site. Data protection is based primarily on NetApp SnapMirror technology.

Some customers choose to move entire workloads to the cloud. This can be more complicated than just using the cloud for data protection, but ONTAP makes moving easier because you do not have to rewrite your applications to use cloud-based storage. ONTAP in the cloud works just like on-premises ONTAP does.

Your on-premises ONTAP system offers data efficiency features that enable you to store more data in less physical space and to tier rarely used data to lower cost storage. Whether you use a hybrid cloud configuration or move an entire workload to the cloud, ONTAP maximizes storage performance and efficiency.

NetApp also offers cloud-based backup (SnapMirror Cloud, Cloud Backup Service, and Cloud Sync) and storage tiering and archiving tools (FabricPool) for ONTAP to help reduce operating expenses and leverage the broad reach of the cloud.

The following figure provides a sample hybrid cloud use case.



For more information on ONTAP and hybrid clouds, see [ONTAP and the Cloud](#) in the ONTAP 9 Documentation Center.

## TR-4597: VMware vSphere for ONTAP

Karl Konnerth, NetApp

NetApp ONTAP software has been a leading storage solution for VMware vSphere environments for almost two decades and continues to add innovative capabilities to simplify management while reducing costs. This document introduces the ONTAP solution for vSphere, including the latest product information and best practices, to streamline deployment, reduce risk, and simplify management.

Best practices supplement other documents such as guides and compatibility lists. They are developed based on lab testing and extensive field experience by NetApp engineers and customers. They might not be the only supported practices that work in every environment, but they are generally the simplest solutions that meet the needs of most customers.

This document is focused on capabilities in recent releases of ONTAP (9.x) running on vSphere 6.0 or later. See the section [ONTAP and vSphere release-specific information](#) for details related to specific releases.

### Why ONTAP for vSphere?

There are many reasons why tens of thousands of customers have selected ONTAP as their storage solution for vSphere, such as a unified storage system supporting both SAN and NAS protocols, robust data protection capabilities using space-efficient NetApp Snapshot copies, and a wealth of tools to help you manage application data. Using a storage system separate from the hypervisor allows you to offload many functions and maximize your investment in vSphere host systems. This approach not only makes sure your host resources are focused on application workloads, but it also avoids random performance effects on applications from storage operations.

Using ONTAP together with vSphere is a great combination that lets you reduce host hardware and VMware software expenses. You can also protect your data at lower cost with consistent high performance. Because virtualized workloads are mobile, you can explore different approaches using Storage vMotion to move VMs across VMFS, NFS, or vVols datastores, all on the same storage system.

Here are key factors customers value today:

- **Unified storage.** Systems running ONTAP software are unified in several significant ways. Originally this approach referred to both NAS and SAN protocols, and ONTAP continues to be a leading platform for SAN along with its original strength in NAS. In the vSphere world, this approach could also mean a unified system for virtual desktop infrastructure (VDI) together with virtual server infrastructure (VSI). Systems running ONTAP software are typically less expensive for VSI than traditional enterprise arrays and yet have advanced storage efficiency capabilities to handle VDI in the same system. ONTAP also unifies a variety of storage media, from SSDs to SATA, and can extend that easily into the cloud. There's no need to buy one flash array for performance, a SATA array for archives, and separate systems for the cloud. ONTAP ties them all together.
- **Virtual volumes and storage policy-based management.** NetApp was an early design partner with VMware in the development of vSphere Virtual Volumes (vVols), providing architectural input and early support for vVols and VMware vSphere APIs for Storage Awareness (VASA). Not only did this approach bring granular VM storage management to VMFS, it also supported automation of storage provisioning through storage policy-based management. This approach allows storage architects to design storage pools with different capabilities that can be easily consumed by VM administrators. ONTAP leads the storage industry in vVol scale, supporting hundreds of thousands of vVols in a single cluster, whereas enterprise array and smaller flash array vendors support as few as several thousand vVols per array. NetApp is also driving the evolution of granular VM management with upcoming capabilities in support of vVols 3.0.
- **Storage efficiency.** Although NetApp was the first to deliver deduplication for production workloads, this innovation wasn't the first or last one in this area. It started with ONTAP Snapshot copies, a space-efficient data protection mechanism with no performance effect, along with FlexClone technology to instantly make read/write copies of VMs for production and backup use. NetApp went on to deliver inline capabilities, including deduplication, compression, and zero-block deduplication, to squeeze out the most storage from expensive SSDs. Most recently, ONTAP added the ability to pack smaller I/O operations and files into a disk block using compaction. The combination of these capabilities has resulted in customers seeing savings of up to 5:1 for VSI and up to 30:1 for VDI.
- **Hybrid cloud.** Whether used for on-premises private cloud, public cloud infrastructure, or a hybrid cloud that combines the best of both, ONTAP solutions help you build your data fabric to streamline and optimize data management. Start with high-performance all-flash systems, then couple them with either disk or cloud storage systems for data protection and cloud compute. Choose from Azure, AWS, IBM, or Google clouds to optimize costs and avoid lock-in. Leverage advanced support for OpenStack and container technologies as needed. NetApp also offers cloud-based backup (SnapMirror Cloud, Cloud Backup Service, and Cloud Sync) and storage tiering and archiving tools (FabricPool) for ONTAP to help reduce operating expenses and leverage the broad reach of the cloud.
- **And more.** Take advantage of the extreme performance of NetApp AFF A-Series arrays to accelerate your virtualized infrastructure while managing costs. Enjoy completely nondisruptive operations, from maintenance to upgrades to complete replacement of your storage system, using scale-out ONTAP clusters. Protect data at rest with NetApp encryption capabilities at no additional cost. Make sure performance meets business service levels through fine-grained quality of service capabilities. They are all part of the broad range of capabilities that come with ONTAP, the industry's leading enterprise data management software.

## ONTAP capabilities for vSphere

### Protocols

ONTAP supports all major storage protocols used for virtualization, such as iSCSI, Fibre Channel (FC), Fibre Channel over Ethernet (FCoE), or Non-Volatile Memory Express over Fibre Channel (NVMe/FC) for SAN environments, as well as NFS (v3 and v4.1), and SMB or S3 for guest connections. Customers are free to pick what works best for their environment and can combine protocols as needed on a single system (for example, augmenting general use of NFS datastores with a few iSCSI LUNs or guest shares).

### Features

There are many ONTAP features that are useful for managing virtualized workloads. Some that require additional product licenses are described in the next section. Others packaged as standalone tools, some for ONTAP and others for the entire NetApp portfolio, are described after that.

Here are further details about base ONTAP features:

- **NetApp Snapshot copies.** ONTAP offers instant Snapshot copies of a VM or datastore with zero performance effect when you create or use a Snapshot copy. They can be used to create a restoration point for a VM prior to patching or for simple data protection. Note that these are different from VMware (consistency) snapshots. The easiest way to make an ONTAP Snapshot copy is to use the SnapCenter Plug-In for VMware vSphere to back up VMs and datastores.
- **Storage efficiency.** ONTAP supports inline and background deduplication and compression, zero-block deduplication, and data compaction.
- **Volume and LUN move.** Allows nondisruptive movement of volumes and LUNs supporting vSphere datastores and vVols within the ONTAP cluster to balance performance and capacity or support nondisruptive maintenance and upgrades.
- **QoS.** QoS allows for managing performance on an individual LUN, volume, or file. This function can be used to limit an unknown or bully VM or to make sure an important VM gets sufficient performance resources.
- **NetApp Volume Encryption, NetApp Aggregate Encryption.** NetApp encryption options offer easy software-based encryption to protect data at rest.
- **FabricPool.** This feature tiers colder data automatically at the block level to a separate object store, freeing up expensive flash storage.
- **REST, Ansible.** Use [ONTAP REST APIs](#) to automate storage and data management, and [Ansible modules](#) for configuration management of your ONTAP systems. Note that some ONTAP features are not well-suited for vSphere workloads. For example, FlexGroup prior to ONTAP 9.8 did not have full cloning support and was not tested with vSphere (see the FlexGroup section for the latest on using it with vSphere). FlexCache is also not optimal for vSphere as it is designed for read-mostly workloads. Writes can be problematic when the cache is disconnected from the origin, resulting in NFS datastore errors on both sides.

### ONTAP licensing

Some ONTAP features that are valuable for managing virtualized workloads require an additional license, whether available at no additional cost, in a license bundle, or a la carte. For many customers, the most cost-effective approach is with a license bundle. Here are the key licenses relevant to vSphere and how they are used:

- **FlexClone.** FlexClone enables instant, space-efficient clones of ONTAP volumes and files. This cloning is used when operations are offloaded to the storage system by VMware vSphere Storage APIs – Array Integration (VAAI), for backup verification and recovery (SnapCenter software), and for vVols cloning and

Snapshot copies. Here is how they are used:

(video)

- VAAI is supported with ONTAP for offloaded copy in support of vSphere (Storage Appliance/Migration) operations. The FlexClone license allows for fast clones within NetApp FlexVol volume, but, if not licensed, it still allows clones using slower block copies.

- A FlexClone license is required for vVols functionality. It enables cloning of vVols within a single datastore or between datastores, and it enables vSphere-managed Snapshot copies of vVols, which are offloaded to the storage system.

- The storage replication adapter (SRA) is used with VMware Site Recovery Manager, and a FlexClone license is required to test recovery in both NAS and SAN environments. SRA may be used without FlexClone for discovery, recovery, and reprottection workflows.

- **SnapRestore.** SnapRestore technology enables instant recovery of a volume in place without copying data. It is required by NetApp backup and recovery tools such as SnapCenter where it is used to mount the datastore for verification and restore operations.

- **SnapMirror.** SnapMirror technology allows for simple, fast replication of data between ONTAP systems on-premises and in the cloud. SnapMirror supports the version flexibility of logical replication with the performance of block replication, sending only changed data to the secondary system. Data can be protected with mirror and/or vault policies, allowing for disaster recovery as well as long-term data retention for backup. SnapMirror supports asynchronous as well as synchronous relationships, and ONTAP 9.8 introduces transparent application failover with SnapMirror Business Continuity.

SnapMirror is required for SRA replication with Site Recovery Manager. It is also required for SnapCenter to enable replication of Snapshot copies to a secondary storage system.

- **SnapCenter.** SnapCenter software provides a unified, scalable platform and plug-in suite for application-consistent data protection and clone management. A SnapCenter license is included with the data protection license bundles for AFF and FAS systems. SnapCenter Plug-in for VMware vSphere is a free product if you are using the following storage systems: FAS, AFF, Cloud Volumes ONTAP, or ONTAP Select. However, SnapRestore and FlexClone licenses are required.

- **MetroCluster.** NetApp MetroCluster is a synchronous replication solution combining high availability and disaster recovery in a campus or metropolitan area to protect against both site disasters and hardware outages. It provides solutions with transparent recovery from failure, with zero data loss (0 RPO) and fast recovery (RTO within minutes). It is used in vSphere environments as part of a vSphere Metro Storage Cluster configuration.

#### Virtualization tools for ONTAP

NetApp offers several standalone software tools that can be used together with ONTAP and vSphere to manage your virtualized environment. The following tools are included with the ONTAP license at no additional cost. See Figure 1 for a depiction of how these tools work together in your vSphere environment.

#### ONTAP tools for VMware vSphere

ONTAP tools for VMware vSphere is a set of tools for using ONTAP storage together with vSphere. The vCenter plug-in, formerly known as the Virtual Storage Console (VSC), simplifies storage management and efficiency features, enhances availability, and reduces storage costs and operational overhead, whether you are using SAN or NAS. It uses best practices for provisioning datastores and optimizes ESXi host settings for NFS and block storage environments. For all these benefits, NetApp recommends using these ONTAP tools as a best practice when using vSphere with systems running ONTAP software. It includes both a server appliance and user interface extensions for vCenter.

## NFS Plug-In for VMware VAAI

The NetApp NFS Plug-In for VMware is a plug-in for ESXi hosts that allows them to use VAAI features with NFS datastores on ONTAP. It supports copy offload for clone operations, space reservation for thick virtual disk files, and Snapshot copy offload. Offloading copy operations to storage is not necessarily faster to complete, but it does offload host resources such as CPU cycles, buffers, and queues. You can use ONTAP tools for VMware vSphere to install the plug-in on ESXi hosts.

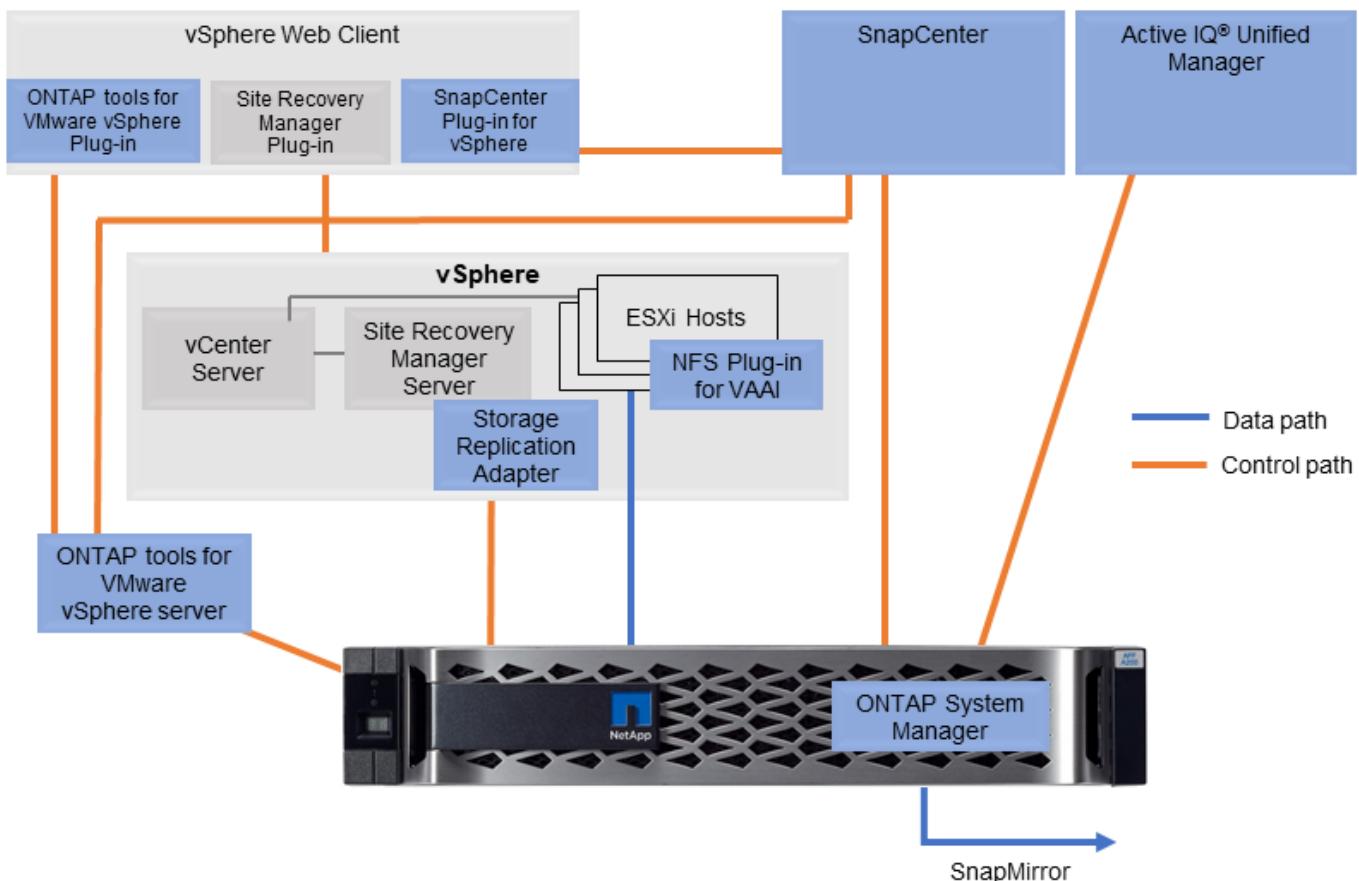
## VASA Provider for ONTAP

The VASA Provider for ONTAP supports the VMware vStorage APIs for Storage Awareness (VASA) framework. It is supplied as part of ONTAP tools for VMware vSphere as a single virtual appliance for ease of deployment. VASA Provider connects vCenter Server with ONTAP to aid in provisioning and monitoring VM storage. It enables VMware Virtual Volumes (vVols) support, management of storage capability profiles and individual VM vVols performance, and alarms for monitoring capacity and compliance with the profiles.

## Storage Replication Adapter

The SRA is used together with VMware Site Recovery Manager (SRM) to manage data replication between production and disaster recovery sites and test the DR replicas nondisruptively. It helps automate the tasks of discovery, recovery, and reprotection. It includes both an SRA server appliance and SRA adapters for the Windows SRM server and SRM appliance. The SRA is supplied as part of ONTAP tools for VMware vSphere.

The following figure depicts ONTAP tools for vSphere.



## Best practices

### vSphere datastore and protocol features

Five protocols are used to connect VMware vSphere to datastores on a system running ONTAP software:

- FC
- FCoE
- NVMe/FC
- iSCSI
- NFS

FC, FCoE, NVMe/FC, and iSCSI are block protocols that use the vSphere Virtual Machine File System (VMFS) to store VMs inside ONTAP LUNs or namespaces that are contained in an ONTAP volume. Note that, starting from vSphere 7.0, VMware no longer supports software FCoE in production environments. NFS is a file protocol that places VMs into datastores (which are simply ONTAP volumes) without the need for VMFS. SMB, iSCSI, or NFS can also be used directly from a guest OS to ONTAP.

The following tables presents vSphere supported traditional datastore features with ONTAP. This information does not apply to vVols datastores, but it does generally applies to vSphere 6.x and 7.x releases using supported ONTAP releases. You can also consult [VMware Configuration Maximums](#) for specific vSphere releases to confirm specific limits.

Capability/Feature	FC/FCoE	iSCSI	NFS
Format	VMFS or raw device mapping (RDM)	VMFS or RDM	N/A
Maximum number of datastores or LUNs	256 targets/HBA	256 targets	256 mounts Default NFS. MaxVolumes is 8. Use ONTAP tools for VMware vSphere to increase to 256.
Maximum datastore size	64TB	64TB	100TB FlexVol volume or greater with FlexGroup volume
Maximum datastore file size (for VMDKs using vSphere version 5.5 and VMFS 5 or later)	62TB	62TB	16TB 62TB is the maximum size supported by vSphere.
Optimal queue depth per LUN or file system	64	64	N/A

The following table lists supported VMware storage-related functionalities.

Capacity/Feature	FC/FCoE	iSCSI	NFS
vMotion	Yes	Yes	Yes
Storage vMotion	Yes	Yes	Yes
VMware HA	Yes	Yes	Yes

Capacity/Feature	FC/FCoE	iSCSI	NFS
Storage Distributed Resource Scheduler (SDRS)	Yes	Yes	Yes
VMware vStorage APIs for Data Protection (VADP)–enabled backup software	Yes	Yes	Yes
Microsoft Cluster Service (MSCS) or failover clustering within a VM	Yes	Yes*	Not supported
Fault Tolerance	Yes	Yes	Yes
Site Recovery Manager	Yes	Yes	Yes
Thin-provisioned VMs (virtual disks)	Yes	Yes	Yes This setting is the default for all VMs on NFS when not using VAAI.
VMware native multipathing	Yes	Yes	N/A

\*NetApp recommends using in-guest iSCSI for Microsoft clusters rather than multi-writer enabled VMDKs in a VMFS datastore. This approach is fully supported by Microsoft and VMware, offers great flexibility with ONTAP (SnapMirror to ONTAP systems on-premises or in the cloud), is easy to configure and automate, and can be protected with SnapCenter. vSphere 7 adds a new clustered VMDK option. This is different from multi-writer enabled VMDKs but requires a datastore presented via the FC protocol, which has clustered VMDK support enabled. Other restrictions apply. See VMware's [Setup for Windows Server Failover Clustering](#) documentation for configuration guidelines.

The following table lists supported ONTAP storage management features.

Capability/Feature	FC/FCoE	iSCSI	NFS
Data deduplication	Savings in the array	Savings in the array	Savings in the datastore
Thin provisioning	Datastore or RDM	Datastore or RDM	Datastore
Resize datastore	Grow only	Grow only	Grow, autogrow, and shrink
SnapCenter plug-ins for Windows, Linux applications (in guest)	Yes	Yes	Yes
Monitoring and host configuration using ONTAP tools for VMware vSphere	Yes	Yes	Yes
Provisioning using ONTAP tools for VMware vSphere	Yes	Yes	Yes

The following table lists supported backup features.

Capability/Feature	FC/FCoE	iSCSI	NFS
ONTAP Snapshot copies	Yes	Yes	Yes
SRM supported by replicated backups	Yes	Yes	Yes
Volume SnapMirror	Yes	Yes	Yes
VMDK image access	VADP-enabled backup software	VADP-enabled backup software	VADP-enabled backup software, vSphere Client, and vSphere Web Client datastore browser
VMDK file-level access	VADP-enabled backup software, Windows only	VADP-enabled backup software, Windows only	VADP-enabled backup software and third-party applications
NDMP granularity	Datastore	Datastore	Datastore or VM

### Selecting a storage protocol

Systems running ONTAP software support all major storage protocols, so customers can choose what is best for their environment, depending on existing and planned networking infrastructure and staff skills. NetApp testing has generally shown little difference between protocols running at similar line speeds, so it is best to focus on your network infrastructure and staff capabilities over raw protocol performance.

The following factors might be useful in considering a choice of protocol:

- **Current customer environment.** Although IT teams are generally skilled at managing Ethernet IP infrastructure, not all are skilled at managing an FC SAN fabric. However, using a general-purpose IP network that's not designed for storage traffic might not work well. Consider the networking infrastructure you have in place, any planned improvements, and the skills and availability of staff to manage them.
- **Ease of setup.** Beyond initial configuration of the FC fabric (additional switches and cabling, zoning, and the interoperability verification of HBA and firmware), block protocols also require creation and mapping of LUNs and discovery and formatting by the guest OS. After the NFS volumes are created and exported, they are mounted by the ESXi host and ready to use. NFS has no special hardware qualification or firmware to manage.
- **Ease of management.** With SAN protocols, if more space is needed, several steps are necessary, including growing a LUN, rescanning to discover the new size, and then growing the file system). Although growing a LUN is possible, reducing the size of a LUN is not, and recovering unused space can require additional effort. NFS allows easy sizing up or down, and this resizing can be automated by the storage system. SAN offers space reclamation through guest OS TRIM/UNMAP commands, allowing space from deleted files to be returned to the array. This type of space reclamation is more difficult with NFS datastores.
- **Storage space transparency.** Storage utilization is typically easier to see in NFS environments because thin provisioning returns savings immediately. Likewise, deduplication and cloning savings are immediately available for other VMs in the same datastore or for other storage system volumes. VM density is also typically greater in an NFS datastore, which can improve deduplication savings as well as reduce management costs by having fewer datastores to manage.

### Datastore layout

ONTAP storage systems offer great flexibility in creating datastores for VMs and virtual disks. Although many ONTAP best practices are applied when using the VSC to provision datastores for vSphere (listed in the

section [Recommended ESXi host and other ONTAP settings](#)), here are some additional guidelines to consider:

- Deploying vSphere with ONTAP NFS datastores results in a high-performing, easy-to-manage implementation that provides VM-to-datastore ratios that cannot be obtained with block-based storage protocols. This architecture can result in a tenfold increase in datastore density with a correlating reduction in the number of datastores. Although a larger datastore can benefit storage efficiency and provide operational benefits, consider using at least four datastores (FlexVol volumes) to store your VMs on a single ONTAP controller to get maximum performance from the hardware resources. This approach also allows you to establish datastores with different recovery policies. Some can be backed up or replicated more frequently than others, based on business needs. Multiple datastores are not required with FlexGroup volumes for performance as it scales by design.
- NetApp recommends the use of FlexVol volumes and, starting with ONTAP 9.8 FlexGroup volumes, NFS datastores. Other ONTAP storage containers such as qtrees are not generally recommended because these are not currently supported by ONTAP tools for VMware vSphere. Deploying datastores as multiple qtrees in a single volume might be useful for highly automated environments that can benefit from datastore-level quotas or VM file clones.
- A good size for a FlexVol volume datastore is around 4TB to 8TB. This size is a good balance point for performance, ease of management, and data protection. Start small (say, 4TB) and grow the datastore as needed (up to the maximum 100TB). Smaller datastores are faster to recover from backup or after a disaster and can be moved quickly across the cluster. Consider the use of ONTAP autosize to automatically grow and shrink the volume as used space changes. The ONTAP tools for VMware vSphere Datastore Provisioning Wizard use autosize by default for new datastores. Additional customization of the grow and shrink thresholds and maximum and minimum size can be done with System Manager or the command line.
- Alternately, VMFS datastores can be configured with LUNs that are accessed by FC, iSCSI, or FCoE. VMFS allows traditional LUNs to be accessed simultaneously by every ESX server in a cluster. VMFS datastores can be up to 64TB in size and consist of up to 32 2TB LUNs (VMFS 3) or a single 64TB LUN (VMFS 5). The ONTAP maximum LUN size is 16TB on most systems, and 128TB on All SAN Array systems. Therefore, a maximum size VMFS 5 datastore on most ONTAP systems can be created by using four 16TB LUNs. While there can be performance benefit for high-I/O workloads with multiple LUNs (with high-end FAS or AFF systems), this benefit is offset by added management complexity to create, manage, and protect the datastore LUNs and increased availability risk. NetApp generally recommends using a single, large LUN for each datastore and only span if there is a special need to go beyond a 16TB datastore. As with NFS, consider using multiple datastores (volumes) to maximize performance on a single ONTAP controller.
- Older guest operating systems (OSs) needed alignment with the storage system for best performance and storage efficiency. However, modern vendor-supported OSs from Microsoft and Linux distributors such as Red Hat no longer require adjustments to align the file system partition with the blocks of the underlying storage system in a virtual environment. If you are using an old OS that might require alignment, search the NetApp Support Knowledgebase for articles using “VM alignment” or request a copy of TR-3747 from a NetApp sales or partner contact.
- Avoid the use of defragmentation utilities within the guest OS, as this offers no performance benefit and affects storage efficiency and Snapshot copy space usage. Also consider turning off search indexing in the guest OS for virtual desktops.
- ONTAP has led the industry with innovative storage efficiency features, allowing you to get the most out of your usable disk space. AFF systems take this efficiency further with default inline deduplication and compression. Data is deduplicated across all volumes in an aggregate, so you no longer need to group similar operating systems and similar applications within a single datastore to maximize savings.
- In some cases, you might not even need a datastore. For the best performance and manageability, avoid using a datastore for high-I/O applications such as databases and some applications. Instead, consider guest-owned file systems such as NFS or iSCSI file systems managed by the guest or with RDMs. For

specific application guidance, see NetApp technical reports for your application. For example, [TR-3633: Oracle Databases on Data ONTAP](#) has a section about virtualization with helpful details.

- First Class Disks (or Improved Virtual Disks) allow for vCenter-managed disks independent of a VM with vSphere 6.5 and later. While primarily managed by API, they can be useful with vVols, especially when managed by OpenStack or Kubernetes tools. They are supported by ONTAP as well as ONTAP tools for VMware vSphere.

### Datastore and VM migration

When migrating VMs from an existing datastore on another storage system to ONTAP, here are some practices to keep in mind:

- Use Storage vMotion to move the bulk of your virtual machines to ONTAP. Not only is this approach nondisruptive to running VMs, it also allows ONTAP storage efficiency features such as inline deduplication and compression to process the data as it migrates. Consider using vCenter capabilities to select multiple VMs from the inventory list and then schedule the migration (use Ctrl key while clicking Actions) at an appropriate time.
- While you could carefully plan a migration to appropriate destination datastores, it is often simpler to migrate in bulk and then organize later as needed. If you have specific data protection needs, such as different Snapshot schedules, you might want to use this approach to guide your migration to different datastores.
- Most VMs and their storage may be migrated while running (hot), but migrating attached (not in datastore) storage such as ISOs, LUNs, or NFS volumes from another storage system might require cold migration.
- Virtual machines that need more careful migration include databases and applications that use attached storage. In general, consider the use of the application's tools to manage migration. For Oracle, consider using Oracle tools such as RMAN or ASM to migrate the database files. See [TR-4534](#) for more information. Likewise, for SQL Server, consider using either SQL Server Management Studio or NetApp tools such as SnapManager for SQL Server or SnapCenter.

### ONTAP tools for VMware vSphere

The most important best practice when using vSphere with systems running ONTAP software is to install and use the ONTAP tools for VMware vSphere plug-in (formerly known as Virtual Storage Console). This vCenter plug-in simplifies storage management, enhances availability, and reduces storage costs and operational overhead, whether using SAN or NAS. It uses best practices for provisioning datastores and optimizes ESXi host settings for multipath and HBA timeouts (these are described in Appendix B). Because it's a vCenter plug-in, it's available to all vSphere web clients that connect to the vCenter server.

The plug-in also helps you use other ONTAP tools in vSphere environments. It allows you to install the NFS Plug-In for VMware VAAI, which enables copy offload to ONTAP for VM cloning operations, space reservation for thick virtual disk files, and ONTAP Snapshot copy offload.

The plug-in is also the management interface for many functions of the VASA Provider for ONTAP, supporting storage policy-based management with vVols. After ONTAP tools for VMware vSphere is registered, use it to create storage capability profiles, map them to storage, and make sure of datastore compliance with the profiles over time. The VASA Provider also provides an interface to create and manage vVol datastores.

In general, NetApp recommends using the ONTAP tools for VMware vSphere interface within vCenter to provision traditional and vVols datastores to make sure best practices are followed.

### General Networking

Configuring network settings when using vSphere with systems running ONTAP software is straightforward and

similar to other network configuration. Here are some things to consider:

- Separate storage network traffic from other networks. A separate network can be achieved by using a dedicated VLAN or separate switches for storage. If the storage network shares physical paths such as uplinks, you might need QoS or additional uplink ports to make sure of sufficient bandwidth. Don't connect hosts directly to storage; use switches to have redundant paths and allow VMware HA to work without intervention.
- Jumbo frames can be used if desired and supported by your network, especially when using iSCSI. If they are used, make sure they are configured identically on all network devices, VLANs, and so on in the path between storage and the ESXi host. Otherwise, you might see performance or connection problems. The MTU must also be set identically on the ESXi virtual switch, the VMkernel port, and also on the physical ports or interface groups of each ONTAP node.
- NetApp only recommends disabling network flow control on the cluster network ports within an ONTAP cluster. NetApp makes no other recommendations for best practices for the remaining network ports used for data traffic. You should enable or disable as necessary. See [TR-4182](#) for more background on flow control.
- When ESXi and ONTAP storage arrays are connected to Ethernet storage networks, NetApp recommends configuring the Ethernet ports to which these systems connect as Rapid Spanning Tree Protocol (RSTP) edge ports or by using the Cisco PortFast feature. NetApp recommends enabling the Spanning-Tree PortFast trunk feature in environments that use the Cisco PortFast feature and that have 802.1Q VLAN trunking enabled to either the ESXi server or the ONTAP storage arrays.
- NetApp recommends the following best practices for link aggregation:
  - Use switches that support link aggregation of ports on two separate switch chassis, using a multichassis link aggregation group approach such as Cisco's Virtual PortChannel (vPC).
  - Disable LACP for switch ports connected to ESXi unless using dvSwitches 5.1 or later with LACP configured.
  - Use LACP to create link aggregates for ONTAP storage systems, with dynamic multimode interface groups with IP hash.
  - Use IP hash teaming policy on ESXi.

The following table provides a summary of network configuration items and indicates where the settings are applied.

Item	ESXi	Switch	Node	SVM
IP address	VMkernel	No**	No**	Yes
Link aggregation	Virtual switch	Yes	Yes	No*
VLAN	VMkernel and VM port groups	Yes	Yes	No*
Flow control	NIC	Yes	Yes	No*
Spanning tree	No	Yes	No	No
MTU (for jumbo frames)	Virtual switch and VMkernel port (9000)	Yes (set to max)	Yes (9000)	No*
Failover groups	No	No	Yes (create)	Yes (select)

\*SVM LIFs connect to ports, interface groups, or VLAN interfaces that have VLAN, MTU, and other settings,

but the settings are not managed at the SVM level.

\*\*These devices have IP addresses of their own for management, but these addresses are not used in the context of ESXi storage networking.

#### SAN (FC, FCoE, NVMe/FC, iSCSI), RDM

In vSphere, there are three ways to use block storage LUNs:

- With VMFS datastores
- With raw device mapping (RDM)
- As a LUN accessed and controlled by a software initiator from a VM guest OS

VMFS is a high-performance clustered file system that provides datastores that are shared storage pools. VMFS datastores can be configured with LUNs that are accessed using FC, iSCSI, FCoE, or NVMe namespaces accessed by the NVMe/FC protocol. VMFS allows traditional LUNs to be accessed simultaneously by every ESX server in a cluster. The ONTAP maximum LUN size is generally 16TB; therefore, a maximum-size VMFS 5 datastore of 64TB (see the first table in this section) is created by using four 16TB LUNs (All SAN Array systems support the maximum VMFS LUN size of 64TB). Because the ONTAP LUN architecture does not have small individual queue depths, VMFS datastores in ONTAP can scale to a greater degree than with traditional array architectures in a relatively simple manner.

vSphere includes built-in support for multiple paths to storage devices, referred to as native multipathing (NMP). NMP can detect the type of storage for supported storage systems and automatically configures the NMP stack to support the capabilities of the storage system in use.

Both NMP and NetApp ONTAP support Asymmetric Logical Unit Access (ALUA) to negotiate optimized and nonoptimized paths. In ONTAP, an ALUA-optimized path follows a direct data path, using a target port on the node that hosts the LUN being accessed. ALUA is turned on by default in both vSphere and ONTAP. The NMP recognizes the ONTAP cluster as ALUA, and it uses the ALUA storage array type plug-in ([VMW\\_SATP\\_ALUA](#)) and selects the round robin path selection plug-in ([VMW\\_PSP\\_RR](#)).

ESXi 6 supports up to 256 LUNs and up to 1,024 total paths to LUNs. Any LUNs or paths beyond these limits are not seen by ESXi. Assuming the maximum number of LUNs, the path limit allows four paths per LUN. In a larger ONTAP cluster, it is possible to reach the path limit before the LUN limit. To address this limitation, ONTAP supports selective LUN map (SLM) in release 8.3 and later.

SLM limits the nodes that advertise paths to a given LUN. It is a NetApp best practice to have at least one LIF per node per SVM and to use SLM to limit the paths advertised to the node hosting the LUN and its HA partner. Although other paths exist, they aren't advertised by default. It is possible to modify the paths advertised with the add and remove reporting node arguments within SLM. Note that LUNs created in releases prior to 8.3 advertise all paths and need to be modified to only advertise the paths to the hosting HA pair. For more information about SLM, review section 5.9 of [TR-4080](#). The previous method of portsets can also be used to further reduce the available paths for a LUN. Portsets help by reducing the number of visible paths through which initiators in an igroup can see LUNs.

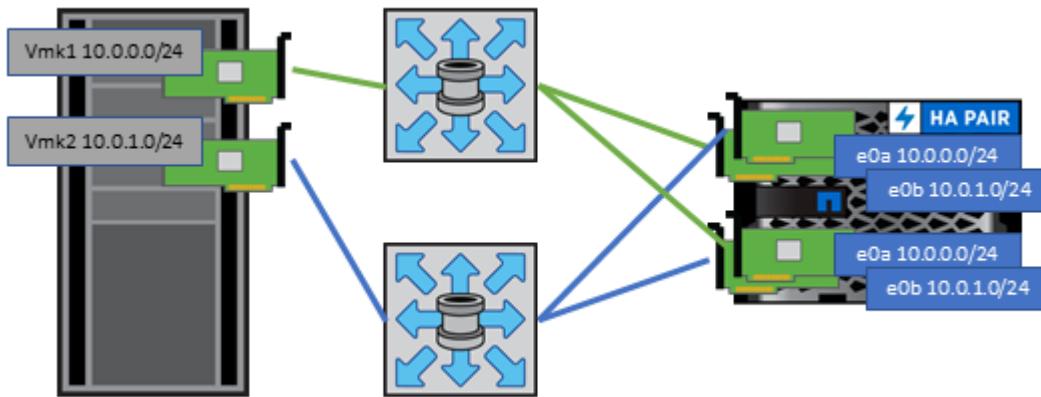
- SLM is enabled by default. Unless you are using portsets, no additional configuration is required.
- For LUNs created prior to Data ONTAP 8.3, manually apply SLM by running the `lun mapping remove-reporting-nodes` command to remove the LUN reporting nodes and restrict LUN access to the LUN-owning node and its HA partner.

Block protocols (iSCSI, FC, and FCoE) access LUNs by using LUN IDs and serial numbers, along with unique names. FC and FCoE use worldwide names (WWNNs and WWPNs), and iSCSI uses iSCSI qualified names (IQNs). The path to LUNs inside the storage is meaningless to the block protocols and is not presented

anywhere in the protocol. Therefore, a volume that contains only LUNs does not need to be internally mounted at all, and a junction path is not needed for volumes that contain LUNs used in datastores. The NVMe subsystem in ONTAP works similarly.

Other best practices to consider:

- Make sure that a logical interface (LIF) is created for each SVM on each node in the ONTAP cluster for maximum availability and mobility. ONTAP SAN best practice is to use two physical ports and LIFs per node, one for each fabric. ALUA is used to parse paths and identify active optimized (direct) paths versus active nonoptimized paths. ALUA is used for FC, FCoE, and iSCSI.
- For iSCSI networks, use multiple VMkernel network interfaces on different network subnets with NIC teaming when multiple virtual switches are present. You can also use multiple physical NICs connected to multiple physical switches to provide HA and increased throughput. The following figure provides an example of multipath connectivity. In ONTAP, configure either a single-mode interface group for failover with two or more links that are connected to two or more switches, or use LACP or other link-aggregation technology with multimode interface groups to provide HA and the benefits of link aggregation.
- If the Challenge-Handshake Authentication Protocol (CHAP) is used in ESXi for target authentication, it must also be configured in ONTAP using the CLI (`vserver iscsi security create`) or with System Manager (edit Initiator Security under Storage > SVMs > SVM Settings > Protocols > iSCSI).
- Use ONTAP tools for VMware vSphere to create and manage LUNs and igroups. The plug-in automatically determines the WWPNs of servers and creates appropriate igroups. It also configures LUNs according to best practices and maps them to the correct igroups.
- Use RDMs with care because they can be more difficult to manage, and they also use paths, which are limited as described earlier. ONTAP LUNs support both [physical and virtual compatibility mode RDMs](#).
- For more on using NVMe/FC with vSphere 7.0, see this [ONTAP NVMe/FC Host Configuration guide](#) and [TR-4684](#). The following figure depicts multipath connectivity from a vSphere host to an ONTAP LUN.



## NFS

vSphere allows customers to use enterprise-class NFS arrays to provide concurrent access to datastores to all the nodes in an ESXi cluster. As mentioned in the datastore section, there are some ease of use and storage efficiency visibility benefits when using NFS with vSphere.

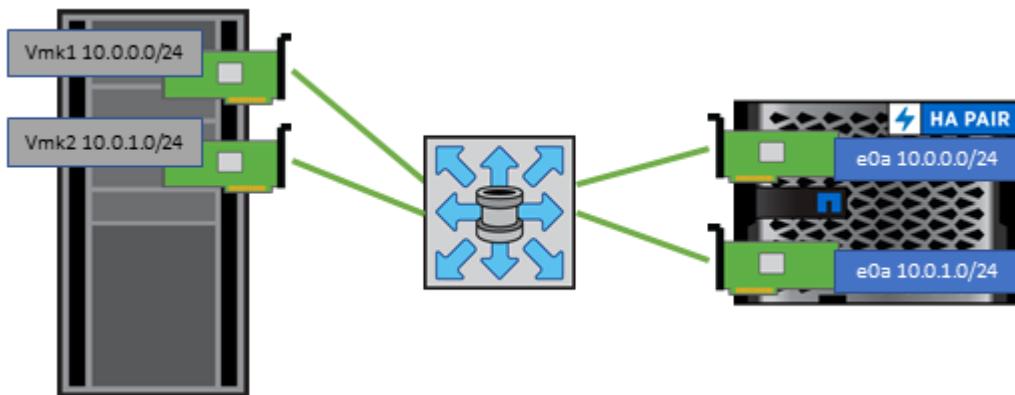
The following best practices are recommended when using ONTAP NFS with vSphere:

- Use a single logical interface (LIF) for each SVM on each node in the ONTAP cluster. Past recommendations of a LIF per datastore are no longer necessary. While direct access (LIF and datastore on same node) is best, don't worry about indirect access because the performance effect is generally minimal (microseconds).

- VMware has supported NFSv3 since VMware Infrastructure 3. vSphere 6.0 added support for NFSv4.1, which enables some advanced capabilities such as Kerberos security. Where NFSv3 uses client-side locking, NFSv4.1 uses server-side locking. Although an ONTAP volume can be exported through both protocols, ESXi can only mount through one protocol. This single protocol mount does not preclude other ESXi hosts from mounting the same datastore through a different version. Make sure to specify the protocol version to use when mounting so that all hosts use the same version and, therefore, the same locking style. Do not mix NFS versions across hosts. If possible, use host profiles to check compliancy.
  - Because there is no automatic datastore conversion between NFSv3 and NFSv4.1, create a new NFSv4.1 datastore and use Storage vMotion to migrate VMs to the new datastore.
  - At the time that this report was written, NetApp is continuing to work with VMware to resolve problems with NFSv4.1 datastores and storage failover. We expect to resolve these issues shortly.
- NFS export policies are used to control access by vSphere hosts. You can use one policy with multiple volumes (datastores). With NFSv3, ESXi uses the sys (UNIX) security style and requires the root mount option to execute VMs. In ONTAP, this option is referred to as superuser, and when the superuser option is used, it is not necessary to specify the anonymous user ID. Note that export policy rules with different values for `-anon` and `-allow-suid` can cause SVM discovery problems with the ONTAP tools. Here's a sample policy:
  - Access Protocol: nfs3
  - Client Match Spec: 192.168.42.21
  - RO Access Rule: sys
  - RW Access Rule: sys
  - Anonymous UID:
  - Superuser: sys
- If the NetApp NFS Plug-In for VMware VAAI is used, the protocol should be set as `nfs` when the export policy rule is created or modified. The NFSv4 protocol is required for VAAI copy offload to work, and specifying the protocol as `nfs` automatically includes both the NFSv3 and the NFSv4 versions.
- NFS datastore volumes are junctioned from the root volume of the SVM; therefore, ESXi must also have access to the root volume to navigate and mount datastore volumes. The export policy for the root volume, and for any other volumes in which the datastore volume's junction is nested, must include a rule or rules for the ESXi servers granting them read-only access. Here's a sample policy for the root volume, also using the VAAI plug-in:
  - Access Protocol. nfs (which includes both nfs3 and nfs4)
  - Client Match Spec. 192.168.42.21
  - RO Access Rule. sys
  - RW Access Rule. never (best security for root volume)
  - Anonymous UID.
  - Superuser. sys (also required for root volume with VAAI)
- Use ONTAP tools for VMware vSphere (the most important best practice):
  - Use ONTAP tools for VMware vSphere to provision datastores because it simplifies management of export policies automatically.
  - When creating datastores for VMware clusters with the plug- in, select the cluster rather than a single ESX server. This choice triggers it to automatically mount the datastore to all hosts in the cluster.
  - Use the plug- in mount function to apply existing datastores to new servers.
  - When not using ONTAP tools for VMware vSphere, use a single export policy for all servers or for each

cluster of servers where additional access control is needed.

- Although ONTAP offers a flexible volume namespace structure to arrange volumes in a tree using junctions, this approach has no value for vSphere. It creates a directory for each VM at the root of the datastore, regardless of the namespace hierarchy of the storage. Thus, the best practice is to simply mount the junction path for volumes for vSphere at the root volume of the SVM, which is how ONTAP tools for VMware vSphere provisions datastores. Not having nested junction paths also means that no volume is dependent on any volume other than the root volume and that taking a volume offline or destroying it, even intentionally, does not affect the path to other volumes.
- A block size of 4K is fine for NTFS partitions on NFS datastores. The following figure depicts connectivity from a vSphere host to an ONTAP NFS datastore.



The following table lists NFS versions and supported features.

vSphere Features	NFSv3	NFSv4.1
vMotion and Storage vMotion	Yes	Yes
High availability	Yes	Yes
Fault tolerance	Yes	Yes
DRS	Yes	Yes
Host profiles	Yes	Yes
Storage DRS	Yes	No
Storage I/O control	Yes	No
SRM	Yes	No
Virtual volumes	Yes	No
Hardware acceleration (VAAI)	Yes	Yes (vSphere 6.5 and later, NetApp VAAI Plug-in 1.1.2)
Kerberos authentication	No	Yes (enhanced with vSphere 6.5 and later to support AES, krb5i)
Multipathing support	No	No (ESXi 6.5 and later supports through session trunking; ONTAP supports through pNFS)

## FlexGroup

ONTAP 9.8 adds support for FlexGroup datastores in vSphere, along with the ONTAP tools for VMware vSphere 9.8 release. FlexGroup simplifies the creation of large datastores and automatically creates a number of constituent volumes to get maximum performance from an ONTAP system. Use FlexGroup with vSphere for a single, scalable vSphere datastore with the power of a full ONTAP cluster.

In addition to extensive system testing with vSphere workloads, ONTAP 9.8 also adds a new copy offload mechanism for FlexGroup datastores. This uses an improved copy engine to copy files between constituents in the background while allowing access on both source and destination. Multiple copies use instantly available, space-efficient file clones within a constituent when needed based on scale.

ONTAP 9.8 also adds new file-based performance metrics (IOPS, throughput, and latency) for FlexGroup files, and these metrics can be viewed in the ONTAP tools for VMware vSphere dashboard and VM reports. The ONTAP tools for VMware vSphere plug-in also allows you to set Quality of Service (QoS) rules using a combination of maximum and/or minimum IOPS. These can be set across all VMs in a datastore or individually for specific VMs.

Here are some additional best practices that NetApp has developed:

- Use FlexGroup provisioning defaults. While ONTAP tools for VMware vSphere is recommended because it creates and mounts the FlexGroup within vSphere, ONTAP System Manager or the command line might be used for special needs. Even then, use the defaults such as the number of constituent members per node because this is what has been tested with vSphere.
- When sizing a FlexGroup datastore, keep in mind that the FlexGroup consists of multiple smaller FlexVol volumes that create a larger namespace. As such, size the datastore to be at least 8x the size of your largest virtual machine. For example, if you have a 6TB VM in your environment, size the FlexGroup datastore no smaller than 48TB.
- Allow FlexGroup to manage datastore space. Autosize and Elastic Sizing have been tested with vSphere datastores. Should the datastore get close to full capacity, use ONTAP tools for VMware vSphere or another tool to resize the FlexGroup volume. FlexGroup keeps capacity and inodes balanced across constituents, prioritizing files within a folder (VM) to the same constituent if capacity allows.
- VMware and NetApp do not currently support a common multipath networking approach. For NFSv4.1, NetApp supports pNFS, whereas VMware supports session trunking. NFSv3 does not support multiple physical paths to a volume. For FlexGroup with ONTAP 9.8, our recommended best practice is to let ONTAP tools for VMware vSphere make the single mount, because the effect of indirect access is typically minimal (microseconds). It's possible to use round-robin DNS to distribute ESXi hosts across LIFs on different nodes in the FlexGroup, but this would require the FlexGroup to be created and mounted without ONTAP tools for VMware vSphere. Then the performance management features would not be available.
- FlexGroup vSphere datastore support has been tested up to 1500 VMs with the 9.8 release.
- Use the NFS Plug-In for VMware VAAI for copy offload. Note that while cloning is enhanced within a FlexGroup datastore, ONTAP does not provide significant performance advantages versus ESXi host copy when copying VMs between FlexVol and/or FlexGroup volumes.
- Use ONTAP tools for VMware vSphere 9.8 to monitor performance of FlexGroup VMs using ONTAP metrics (dashboard and VM reports), and to manage QoS on individual VMs. These metrics are not currently available through ONTAP commands or APIs.
- QoS (max/min IOPS) can be set on individual VMs or on all VMs in a datastore at that time. Setting QoS on all VMs replaces any separate per-VM settings. Settings do not extend to new or migrated VMs in the future; either set QoS on the new VMs or re-apply QoS to all VMs in the datastore.
- SnapCenter Plug-In for VMware vSphere release 4.4 supports backup and recovery of VMs in a FlexGroup datastore on the primary storage system. While SnapMirror may be used manually to replicate a FlexGroup

to a secondary system, SCV 4.4 does not manage the secondary copies.

## Other capabilities for vSphere

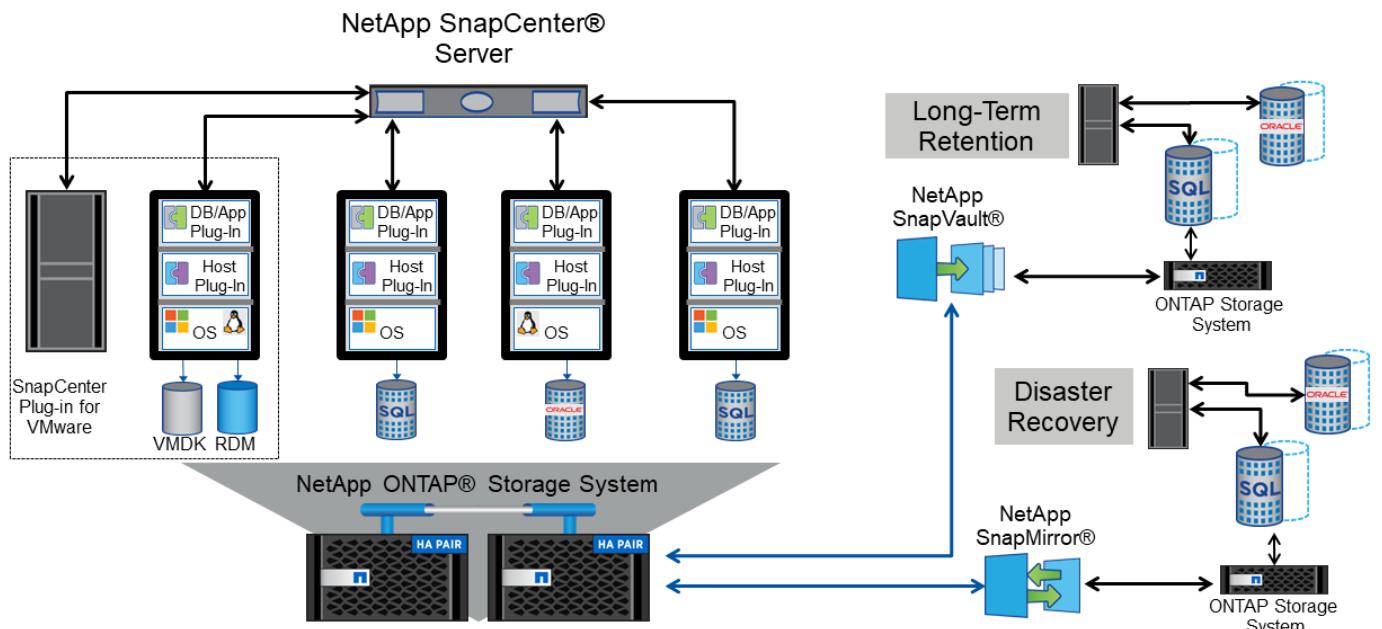
### Data protection

Backing up your VMs and quickly recovering them are among the great strengths of ONTAP for vSphere, and it is easy to manage this ability inside vCenter with the SnapCenter Plug-In for VMware vSphere. Use Snapshot copies to make quick copies of your VM or datastore without affecting performance, and then send them to a secondary system using SnapMirror for longer-term off-site data protection. This approach minimizes storage space and network bandwidth by only storing changed information.

SnapCenter allows you to create backup policies that can be applied to multiple jobs. These policies can define schedule, retention, replication, and other capabilities. They continue to allow optional selection of VM-consistent snapshots, which leverages the hypervisor's ability to quiesce I/O before taking a VMware snapshot. However, due to the performance effect of VMware snapshots, they are generally not recommended unless you need the guest file system to be quiesced. Instead, use ONTAP Snapshot copies for general protection, and use application tools such as SnapCenter plug-ins to protect transactional data such as SQL Server or Oracle. These Snapshot copies are different from VMware (consistency) snapshots and are suitable for longer term protection. VMware snapshots are only [recommended](#) for short term use due to performance and other effects.

These plug-ins offer extended capabilities to protect the databases in both physical and virtual environments. With vSphere, you can use them to protect SQL Server or Oracle databases where data is stored on RDM LUNs, iSCSI LUNs directly connected to the guest OS, or VMDK files on either VMFS or NFS datastores. The plug-ins allow specification of different types of database backups, supporting online or offline backup, and protecting database files along with log files. In addition to backup and recovery, the plug-ins also support cloning of databases for development or test purposes.

The following figure depicts an example of SnapCenter deployment.



For enhanced disaster recovery capabilities, consider using the NetApp SRA for ONTAP with VMware Site Recovery Manager. In addition to support for the replication of datastores to a DR site, it also enables nondisruptive testing in the DR environment by cloning the replicated datastores. Recovery from a disaster and

reprotecting production after the outage has been resolved are also made easy by automation built into SRA.

Finally, for the highest level of data protection, consider a VMware vSphere Metro Storage Cluster (vMSC) configuration using NetApp MetroCluster. vMSC is a VMware-certified solution that combines synchronous replication with array-based clustering, giving the same benefits of a high-availability cluster but distributed across separate sites to protect against site disaster. NetApp MetroCluster offers cost-effective configurations for synchronous replication with transparent recovery from any single storage component failure as well as single-command recovery in the event of a site disaster. vMSC is described in greater detail in [TR-4128](#).

### Space reclamation

Space can be reclaimed for other uses when VMs are deleted from a datastore. When using NFS datastores, space is reclaimed immediately when a VM is deleted (of course, this approach only makes sense when the volume is thin provisioned, that is, the volume guarantee is set to none). However, when files are deleted within the VM guest OS, space is not automatically reclaimed with an NFS datastore. For LUN-based VMFS datastores, ESXi as well as the guest OS can issue VAAI UNMAP primitives to the storage (again, when using thin provisioning) to reclaim space. Depending on the release, this support is either manual or automatic.

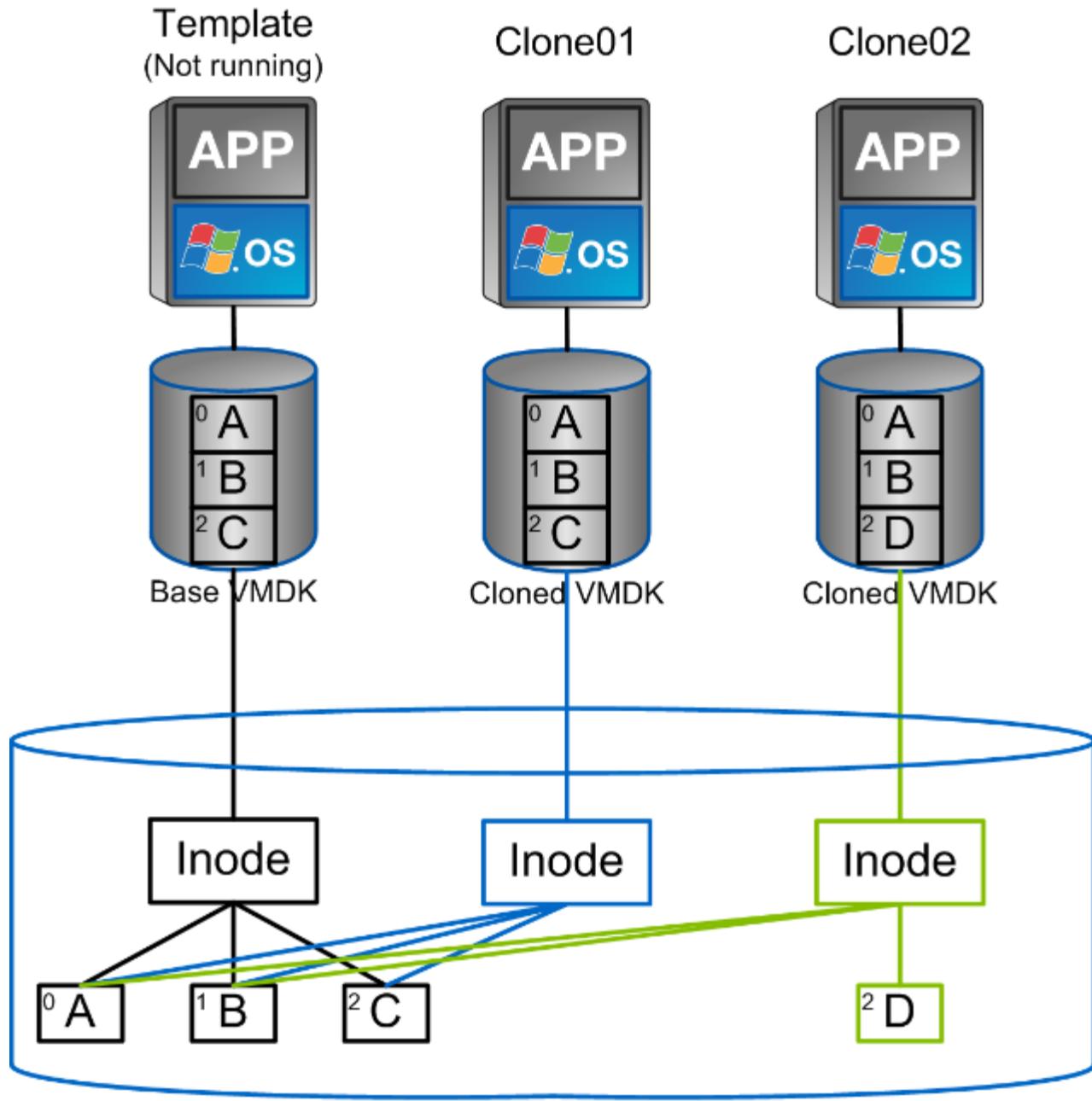
In vSphere 5.5 and later, the `vmkfstools -y` command is replaced by the `esxcli storage vmfs unmap` command, which specifies the number of free blocks (see VMware KB [2057513](#) for more info). In vSphere 6.5 and later when using VMFS 6, space should be automatically reclaimed asynchronously (see [Storage Space Reclamation](#) in the vSphere documentation), but can also be run manually if needed. This automatic UNMAP is supported by ONTAP, and ONTAP tools for VMware vSphere sets it to low priority.

### VM and datastore cloning

Cloning a storage object allows you to quickly create copies for further use, such as provisioning additional VMs, backup/recovery operations, and so on. In vSphere, you can clone a VM, virtual disk, vVol, or datastore. After being cloned, the object can be further customized, often through an automated process. vSphere supports both full copy clones, as well as linked clones, where it tracks changes separately from the original object.

Linked clones are great for saving space, but they increase the amount of I/O that vSphere handles for the VM, affecting performance of that VM and perhaps the host overall. That's why NetApp customers often use storage system-based clones to get the best of both worlds: efficient use of storage and increased performance.

The following figure depicts ONTAP cloning.



## NetApp FlexVol Volume

Cloning can be offloaded to systems running ONTAP software through several mechanisms, typically at the VM, vVol, or datastore level. These include the following:

- vVols using the NetApp vSphere APIs for Storage Awareness (VASA) Provider. ONTAP clones are used to support vVol Snapshot copies managed by vCenter that are space-efficient with minimal I/O effect to create and delete them. VMs can also be cloned using vCenter, and these are also offloaded to ONTAP, whether within a single datastore/volume or between datastores/volumes.
- vSphere cloning and migration using vSphere APIs – Array Integration (VAAI). VM cloning operations can be offloaded to ONTAP in both SAN and NAS environments (NetApp supplies an ESXi plug-in to enable VAAI for NFS). vSphere only offloads operations on cold (powered off) VMs in a NAS datastore, whereas operations on hot VMs (cloning and storage vMotion) are also offloaded for SAN. ONTAP uses the most efficient approach based on source, destination, and installed product licenses. This capability is also used by VMware Horizon View.

- SRA (used with VMware Site Recovery Manager). Here, clones are used to test recovery of the DR replica nondisruptively.
- Backup and recovery using NetApp tools such as SnapCenter. VM clones are used to verify backup operations as well as to mount a VM backup so that individual files can be copied.

ONTAP offloaded cloning can be invoked by VMware, NetApp, and third-party tools. Clones that are offloaded to ONTAP have several advantages. They are space-efficient in most cases, needing storage only for changes to the object; there is no additional performance effect to read and write them, and in some cases performance is improved by sharing blocks in high-speed caches. They also offload CPU cycles and network I/O from the ESXi server. Copy offload within a traditional datastore using a FlexVol volume can be fast and efficient with FlexClone licensed, but copies between FlexVol volumes might be slower. If you maintain VM templates as a source of clones, consider placing them within the datastore volume (use folders or content libraries to organize them) for fast, space efficient clones.

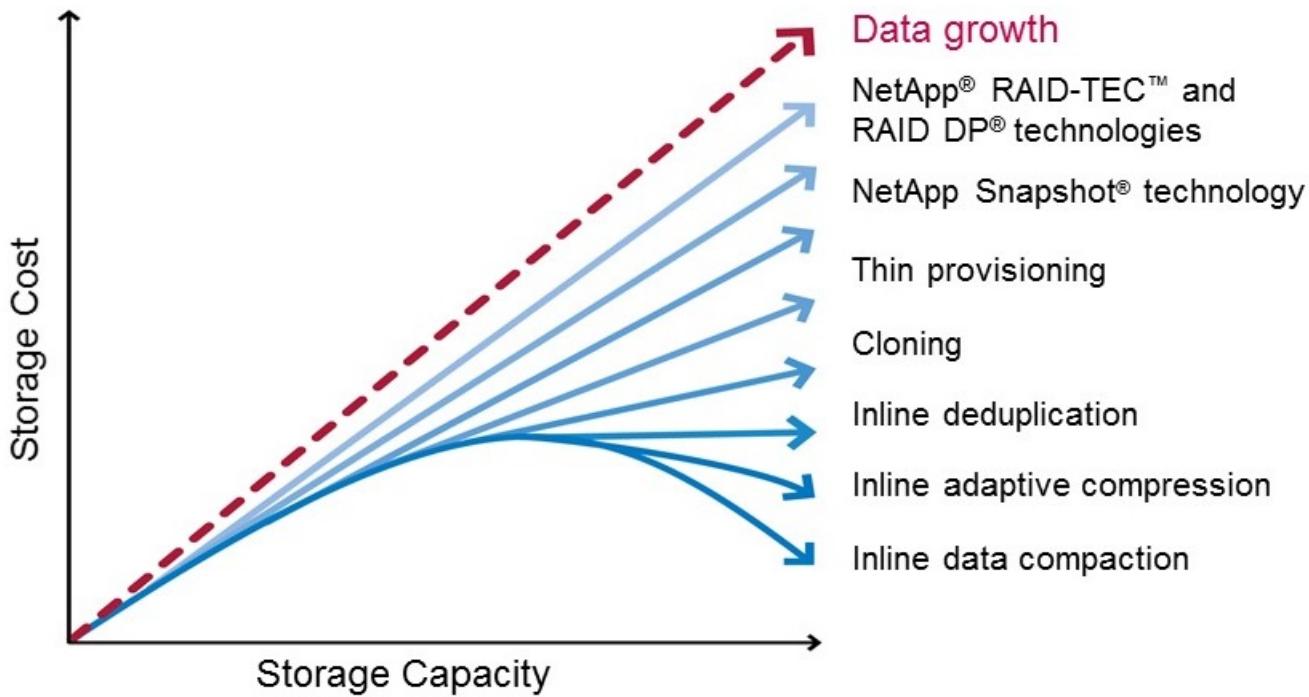
You can also clone a volume or LUN directly within ONTAP to clone a datastore. With NFS datastores, FlexClone technology can clone an entire volume, and the clone can be exported from ONTAP and mounted by ESXi as another datastore. For VMFS datastores, ONTAP can clone a LUN within a volume or a whole volume, including one or more LUNs within it. A LUN containing a VMFS must be mapped to an ESXi initiator group (igroup) and then resignatured by ESXi to be mounted and used as a regular datastore. For some temporary use cases, a cloned VMFS can be mounted without resignaturing. After a datastore is cloned, VMs inside it can be registered, reconfigured, and customized as if they were individually cloned VMs.

In some cases, additional licensed features can be used to enhance cloning, such as SnapRestore for backup or FlexClone. These licenses are often included in license bundles at no additional cost. A FlexClone license is required for vVol cloning operations as well as to support managed Snapshot copies of a vVol (which are offloaded from the hypervisor to ONTAP). A FlexClone license can also improve certain VAAI-based clones when used within a datastore/volume (creates instant, space-efficient copies instead of block copies). It is also used by the SRA when testing recovery of a DR replica, and SnapCenter for clone operations and to browse backup copies to restore individual files.

### **Storage efficiency and thin provisioning**

NetApp has led the industry with storage-efficiency innovation such as the first deduplication for primary workloads, and inline data compaction, which enhances compression and stores small files and I/O efficiently. ONTAP supports both inline and background deduplication, as well as inline and background compression.

The following figure depicts the combined effect of ONTAP storage efficiency features.



Here are recommendations on using ONTAP storage efficiency in a vSphere environment:

- The amount of data deduplication savings realized is based on the commonality of the data. With ONTAP 9.1 and earlier, data deduplication operated at the volume level, but with aggregate deduplication in ONTAP 9.2 and later, data is deduplicated across all volumes in an aggregate on AFF systems. You no longer need to group similar operating systems and similar applications within a single datastore to maximize savings.
- To realize the benefits of deduplication in a block environment, the LUNs must be thin provisioned. Although the LUN is still seen by the VM administrator as taking the provisioned capacity, the deduplication savings are returned to the volume to be used for other needs. NetApp recommends deploying these LUNs in FlexVol volumes that are also thin provisioned (ONTAP tools for VMware vSphere size the volume about 5% larger than the LUN).
- Thin provisioning is also recommended (and is the default) for NFS FlexVol volumes. In an NFS environment, deduplication savings are immediately visible to both storage and VM administrators with thin-provisioned volumes.
- Thin provisioning applies to the VMs as well, where NetApp generally recommends thin-provisioned VMDKs rather than thick. When using thin provisioning, make sure you monitor available space with ONTAP tools for VMware vSphere, ONTAP, or other available tools to avoid out-of-space problems.
- Note that there is no performance penalty when using thin provisioning with ONTAP systems; data is written to available space so that write performance and read performance are maximized. Despite this fact, some products such as Microsoft failover clustering or other low-latency applications might require guaranteed or fixed provisioning, and it is wise to follow these requirements to avoid support problems.
- For maximum deduplication savings, consider scheduling background deduplication on hard disk-based systems or automatic background deduplication on AFF systems. However, the scheduled processes use system resources when running, so ideally they should be scheduled during less active times (such as weekends) or run more frequently to reduce the amount of changed data to be processed. Automatic background deduplication on AFF systems has much less effect on foreground activities. Background compression (for hard disk-based systems) also consumes resources, so it should only be considered for secondary workloads with limited performance requirements.

- NetApp AFF systems primarily use inline storage efficiency capabilities. When data is moved to them using NetApp tools that use block replication such as the 7-Mode Transition Tool, SnapMirror, or Volume Move, it can be useful to run compression and compaction scanners to maximize efficiency savings. Review this NetApp Support [KB article](#) for additional details.
- Snapshot copies might lock blocks that could be reduced by compression or deduplication. When using scheduled background efficiency or one-time scanners, make sure that they run and complete before the next Snapshot copy is taken. Review your Snapshot copies and retention to make sure you only retain needed Snapshot copies, especially before a background or scanner job is run.

The following table provide storage efficiency guidelines for virtualized workloads on different types of ONTAP storage:

Workload	Storage efficiency guidelines		
	AFF	Flash Pool	Hard Disk Drives
VDI and SVI	For primary and secondary workloads, use: <ul style="list-style-type: none"> <li>Adaptive compression</li> <li>Inline deduplication</li> <li>Background deduplication</li> <li>Inline data compaction</li> </ul>	For primary and secondary workloads, use: <ul style="list-style-type: none"> <li>Adaptive compression</li> <li>Inline deduplication</li> <li>Background deduplication</li> <li>Inline data compaction</li> </ul>	For primary workloads, use: <ul style="list-style-type: none"> <li>Background deduplication</li> </ul> For secondary workloads, use: <ul style="list-style-type: none"> <li>Adaptive compression</li> <li>Adaptive background compression</li> <li>Inline deduplication</li> <li>Background deduplication</li> <li>Inline data compaction</li> </ul>

#### Quality of service (QoS)

Systems running ONTAP software can use the ONTAP storage QoS feature to limit throughput in MBps and/or I/Os per second (IOPS) for different storage objects such as files, LUNs, volumes, or entire SVMs.

Throughput limits are useful in controlling unknown or test workloads before deployment to make sure they don't affect other workloads. They can also be used to constrain a bully workload after it is identified. Minimum levels of service based on IOPS are also supported to provide consistent performance for SAN objects in ONTAP 9.2 and for NAS objects in ONTAP 9.3.

With an NFS datastore, a QoS policy can be applied to the entire FlexVol volume or individual VMDK files within it. With VMFS datastores using ONTAP LUNs, the QoS policies can be applied to the FlexVol volume that contains the LUNs or individual LUNs, but not individual VMDK files because ONTAP has no awareness of the VMFS file system. When using vVols, minimum and/or maximum QoS can be set on individual VMs using the storage capability profile and VM storage policy.

The QoS maximum throughput limit on an object can be set in MBps and/or IOPS. If both are used, the first limit reached is enforced by ONTAP. A workload can contain multiple objects, and a QoS policy can be applied to one or more workloads. When a policy is applied to multiple workloads, the workloads share the total limit of the policy. Nested objects are not supported (for example, files within a volume cannot each have their own policy). QoS minimums can only be set in IOPS.

The following tools are currently available for managing ONTAP QoS policies and applying them to objects:

- ONTAP CLI
- ONTAP System Manager
- OnCommand Workflow Automation
- Active IQ Unified Manager
- NetApp PowerShell Toolkit for ONTAP
- ONTAP tools for VMware vSphere VASA Provider

To assign a QoS policy to a VMDK on NFS, note the following guidelines:

- The policy must be applied to the `vmname-flat.vmdk` that contains the actual virtual disk image, not the `vmname.vmdk` (virtual disk descriptor file) or `vmname.vmx` (VM descriptor file).
- Do not apply policies to other VM files such as virtual swap files (`vmname.vswp`).
- When using the vSphere web client to find file paths (Datastore > Files), be aware that it combines the information of the `- flat.vmdk` and `. vmdk` and simply shows one file with the name of the `. vmdk` but the size of the `- flat.vmdk`. Add `-flat` into the file name to get the correct path.

To assign a QoS policy to a LUN, including VMFS and RDM, the ONTAP SVM (displayed as Vserver), LUN path, and serial number can be obtained from the Storage Systems menu on the ONTAP tools for VMware vSphere home page. Select the storage system (SVM), and then Related Objects > SAN. Use this approach when specifying QoS using one of the ONTAP tools.

Maximum and minimum QoS can be easily assigned to a vVol-based VM with ONTAP tools for VMware vSphere or Virtual Storage Console 7.1 and later. When creating the storage capability profile for the vVol container, specify a max and/or min IOPS value under the performance capability and then reference this SCP with the VM's storage policy. Use this policy when creating the VM or apply the policy to an existing VM.

FlexGroup datastores offer enhanced QoS capabilities when using ONTAP tools for VMware vSphere 9.8 and later. You can easily set QoS on all VMs in a datastore or on specific VMs. See the FlexGroup section of this report for more information.

## ONTAP QoS and VMware SIOC

ONTAP QoS and VMware vSphere Storage I/O Control (SIOC) are complementary technologies that vSphere and storage administrators can use together to manage performance of vSphere VMs hosted on systems running ONTAP software. Each tool has its own strengths, as shown in the following table. Because of the different scopes of VMware vCenter and ONTAP, some objects can be seen and managed by one system and not the other.

Property	ONTAP QoS	VMware SIOC
When active	Policy is always active	Active when contention exists (datastore latency over threshold)
Type of units	IOPS, MBps	IOPS, shares
vCenter or application scope	Multiple vCenter environments, other hypervisors and applications	Single vCenter server
Set QoS on VM?	VMDK on NFS only	VMDK on NFS or VMFS
Set QoS on LUN (RDM)?	Yes	No

Property	ONTAP QoS	VMware SIOC
Set QoS on LUN (VMFS)?	Yes	No
Set QoS on volume (NFS datastore)?	Yes	No
Set QoS on SVM (tenant)?	Yes	No
Policy-based approach?	Yes; can be shared by all workloads in the policy or applied in full to each workload in the policy.	Yes, with vSphere 6.5 and later.
License required	Included with ONTAP	Enterprise Plus

### VMware Storage Distributed Resource Scheduler

VMware Storage Distributed Resource Scheduler (SDRS) is a vSphere feature that places VMs on storage based on the current I/O latency and space usage. It then moves the VM or VMDKs nondisruptively between the datastores in a datastore cluster (also referred to as a pod), selecting the best datastore in which to place the VM or VMDKs in the datastore cluster. A datastore cluster is a collection of similar datastores that are aggregated into a single unit of consumption from the vSphere administrator's perspective.

When using SDRS with the NetApp ONTAP tools for VMware vSphere, you must first create a datastore with the plug-in, use vCenter to create the datastore cluster, and then add the datastore to it. After the datastore cluster is created, additional datastores can be added to the datastore cluster directly from the provisioning wizard on the Details page.

Other ONTAP best practices for SDRS include the following:

- All datastores in the cluster should use the same type of storage (such as SAS, SATA, or SSD), be either all VMFS or NFS datastores, and have the same replication and protection settings.
- Consider using SDRS in default (manual) mode. This approach allows you to review the recommendations and decide whether to apply them or not. Be aware of these effects of VMDK migrations:
  - When SDRS moves VMDKs between datastores, any space savings from ONTAP cloning or deduplication are lost. You can rerun deduplication to regain these savings.
  - After SDRS moves VMDKs, NetApp recommends recreating the Snapshot copies at the source datastore because space is otherwise locked by the VM that was moved.
  - Moving VMDKs between datastores on the same aggregate has little benefit, and SDRS does not have visibility into other workloads that might share the aggregate.

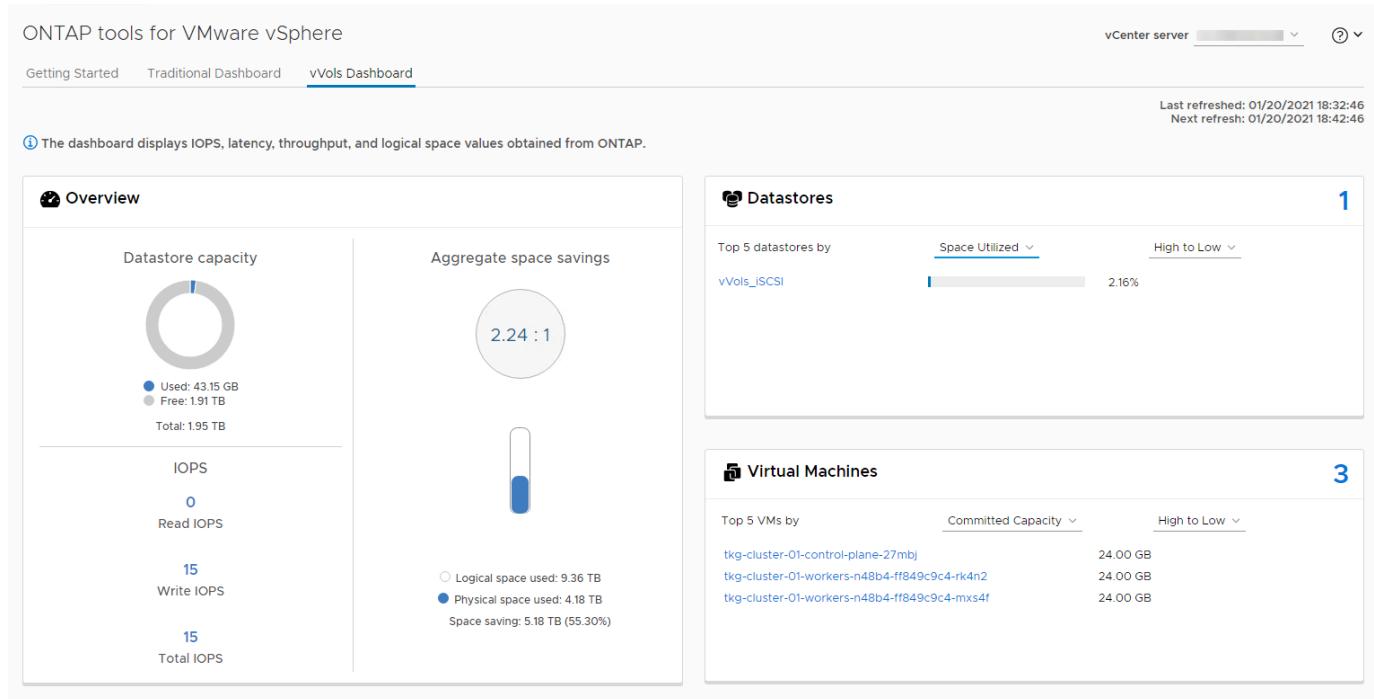
### Storage policy-based management and vVols

VMware vSphere APIs for Storage Awareness (VASA) make it easy for a storage administrator to configure datastores with well-defined capabilities and let the VM administrator use those whenever needed to provision VMs without having to interact with each other. It's worth taking a look at this approach to see how it can streamline your virtualization storage operations and avoid a lot of trivial work.

Prior to VASA, VM administrators could define VM storage policies, but they had to work with the storage administrator to identify appropriate datastores, often by using documentation or naming conventions. With VASA, the storage administrator can define a range of storage capabilities, including performance, tiering, encryption, and replication. A set of capabilities for a volume or a set of volumes is called a storage capability profile (SCP).

The SCP supports minimum and/or maximum QoS for a VM's data vVols. Minimum QoS is supported only on AFF systems. ONTAP tools for VMware vSphere includes a dashboard that displays VM granular performance and logical capacity for vVols on ONTAP systems.

The following figure depicts ONTAP tools for VMware vSphere 9.8 vVols dashboard.



After the storage capability profile is defined, it can be used to provision VMs using the storage policy that identifies its requirements. The mapping between the VM storage policy and the datastore storage capability profile allows vCenter to display a list of compatible datastores for selection. This approach is known as storage policy-based management.

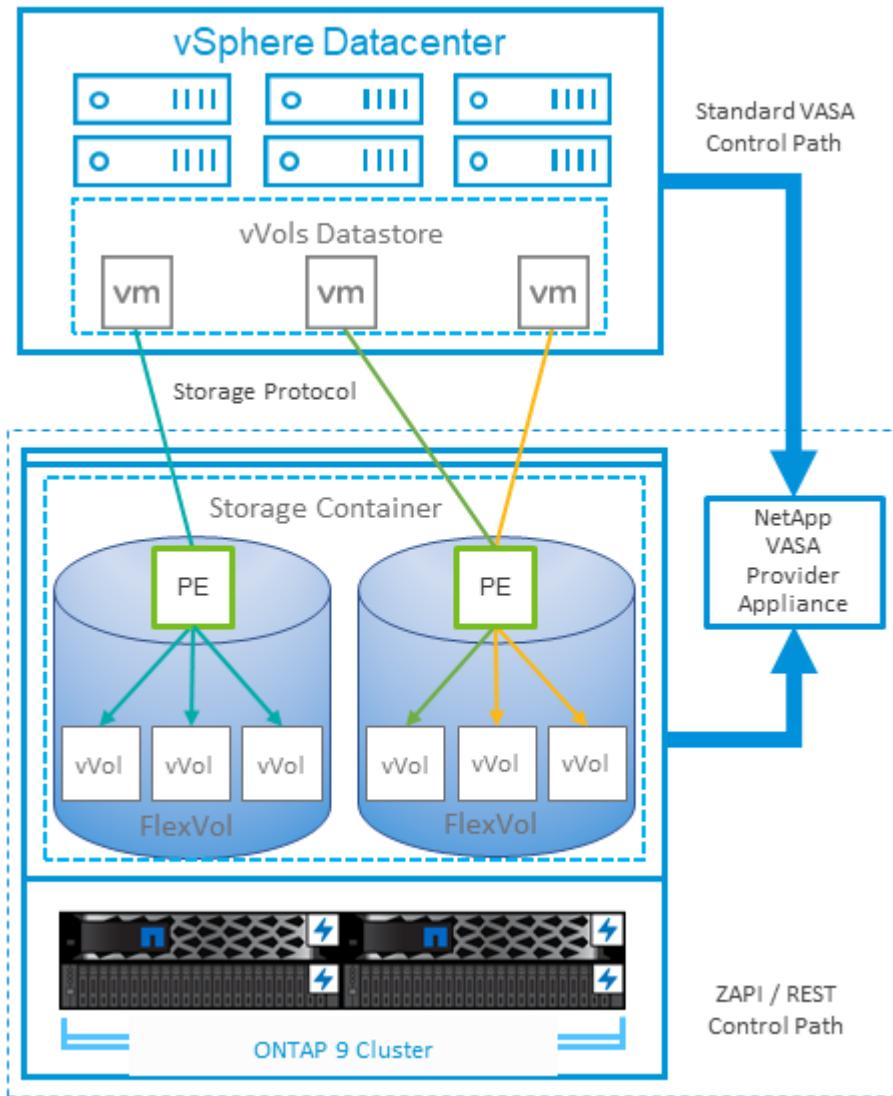
VASA provides the technology to query storage and return a set of storage capabilities to vCenter. VASA vendor providers supply the translation between the storage system APIs and constructs and the VMware APIs that are understood by vCenter. NetApp's VASA Provider for ONTAP is offered as part of the ONTAP tools for VMware vSphere appliance VM, and the vCenter plug-in provides the interface to provision and manage vVol datastores, as well as the ability to define storage capability profiles (SCPs).

ONTAP supports both VMFS and NFS vVol datastores. Using vVols with SAN datastores brings some of the benefits of NFS such as VM-level granularity. Here are some best practices to consider, and you can find additional information in [TR-4400](#):

- A vVol datastore can consist of multiple FlexVol volumes on multiple cluster nodes. The simplest approach is a single datastore, even when the volumes have different capabilities. SPBM makes sure that a compatible volume is used for the VM. However, the volumes must all be part of a single ONTAP SVM and accessed using a single protocol. One LIF per node for each protocol is sufficient. Avoid using multiple ONTAP releases within a single vVol datastore because the storage capabilities might vary across releases.
- Use the ONTAP tools for VMware vSphere plug-in to create and manage vVol datastores. In addition to managing the datastore and its profile, it automatically creates a protocol endpoint to access the vVols if needed. If LUNs are used, note that LUN PEs are mapped using LUN IDs 300 and higher. Verify that the ESXi host advanced system setting `Disk.MaxLUN` allows a LUN ID number that is higher than 300 (the default is 1,024). Do this step by selecting the ESXi host in vCenter, then the Configure tab, and find `Disk.MaxLUN` in the list of Advanced System Settings.

- Do not install or migrate VASA Provider, vCenter Server (appliance or Windows based), or ONTAP tools for VMware vSphere itself onto a vVols datastore, because they are then mutually dependent, limiting your ability to manage them in the event of a power outage or other data center disruption.
- Back up the VASA Provider VM regularly. At a minimum, create hourly Snapshot copies of the traditional datastore that contains VASA Provider. For more about protecting and recovering the VASA Provider, see this [KB article](#).

The following figure shows vVols components.



#### Cloud migration and backup

Another ONTAP strength is broad support for the hybrid cloud, merging systems in your on-premises private cloud with public cloud capabilities. Here are some NetApp cloud solutions that can be used in conjunction with vSphere:

- **Cloud Volumes.** NetApp Cloud Volumes Service for AWS or GCP and Azure NetApp Files for ANF provide high-performance, multi-protocol managed storage services in the leading public cloud environments. They can be used directly by VMware Cloud VM guests.
- **Cloud Volumes ONTAP.** NetApp Cloud Volumes ONTAP data management software delivers control, protection, flexibility, and efficiency to your data on your choice of cloud. Cloud Volumes ONTAP is cloud-

native data management software built on NetApp ONTAP storage software. Use together with Cloud Manager to deploy and manage Cloud Volumes ONTAP instances together with your on-premises ONTAP systems. Take advantage of advanced NAS and iSCSI SAN capabilities together with unified data management, including snapshot copies and SnapMirror replication.

- **Cloud Services.** Use Cloud Backup Service or SnapMirror Cloud to protect data from on-premises systems using public cloud storage. Cloud Sync helps migrate and keep your data in sync across NAS, object stores, and Cloud Volumes Service storage.
- **FabricPool.** FabricPool offers quick and easy tiering for ONTAP data. Cold blocks in Snapshot copies can be migrated to an object store in either public clouds or a private StorageGRID object store and are automatically recalled when the ONTAP data is accessed again. Or use the object tier as a third level of protection for data that is already managed by SnapVault. This approach can allow you to [store more Snapshot copies of your VMs](#) on primary and/or secondary ONTAP storage systems.
- **ONTAP Select.** Use NetApp software-defined storage to extend your private cloud across the Internet to remote facilities and offices, where you can use ONTAP Select to support block and file services as well as the same vSphere data management capabilities you have in your enterprise data center.

When designing your VM-based applications, consider future cloud mobility. For example, rather than placing application and data files together use a separate LUN or NFS export for the data. This allows you to migrate the VM and data separately to cloud services.

#### Encryption for vSphere data

Today, there are increasing demands to protect data at rest through encryption. Although the initial focus was on financial and healthcare information, there is growing interest in protecting all information, whether it's stored in files, databases, or other data types.

Systems running ONTAP software make it easy to protect any data with at-rest encryption. NetApp Storage Encryption (NSE) uses self-encrypting disk drives with ONTAP to protect SAN and NAS data. NetApp also offers NetApp Volume Encryption and NetApp Aggregate Encryption as a simple, software-based approach to encrypt volumes on any disk drives. This software encryption doesn't require special disk drives or external key managers and is available to ONTAP customers at no additional cost. You can upgrade and start using it without any disruption to your clients or applications, and they are validated to the FIPS 140-2 level 1 standard, including the onboard key manager.

There are several approaches for protecting the data of virtualized applications running on VMware vSphere. One approach is to protect the data with software inside the VM at the guest OS level. Newer hypervisors such as vSphere 6.5 now support encryption at the VM level as another alternative. However, NetApp software encryption is simple and easy and has these benefits:

- **No effect on the virtual server CPU.** Some virtual server environments need every available CPU cycle for their applications, yet tests have shown up to 5x CPU resources are needed with hypervisor-level encryption. Even if the encryption software supports Intel's AES-NI instruction set to offload encryption workload (as NetApp software encryption does), this approach might not be feasible due to the requirement for new CPUs that are not compatible with older servers.
- **Onboard key manager included.** NetApp software encryption includes an onboard key manager at no additional cost, which makes it easy to get started without high-availability key management servers that are complex to purchase and use.
- **No effect on storage efficiency.** Storage efficiency techniques such as deduplication and compression are widely used today and are key to using flash disk media cost-effectively. However, encrypted data cannot typically be deduplicated or compressed. NetApp hardware and storage encryption operate at a lower level and allow full use of industry-leading NetApp storage efficiency features, unlike other approaches.

- **Easy datastore granular encryption.** With NetApp Volume Encryption, each volume gets its own AES 256-bit key. If you need to change it, you can do so with a single command. This approach is great if you have multiple tenants or need to prove independent encryption for different departments or apps. This encryption is managed at the datastore level, which is a lot easier than managing individual VMs.

It's simple to get started with software encryption. After the license is installed, simply configure the onboard key manager by specifying a passphrase and then either create a new volume or do a storage-side volume move to enable encryption. NetApp is working to add more integrated support for encryption capabilities in future releases of its VMware tools.

### Active IQ Unified Manager

Active IQ Unified Manager provides visibility into the VMs in your virtual infrastructure and enables monitoring and troubleshooting storage and performance issues in your virtual environment.

A typical virtual infrastructure deployment on ONTAP has various components that are spread across compute, network, and storage layers. Any performance lag in a VM application might occur due to a combination of latencies faced by the various components at the respective layers.

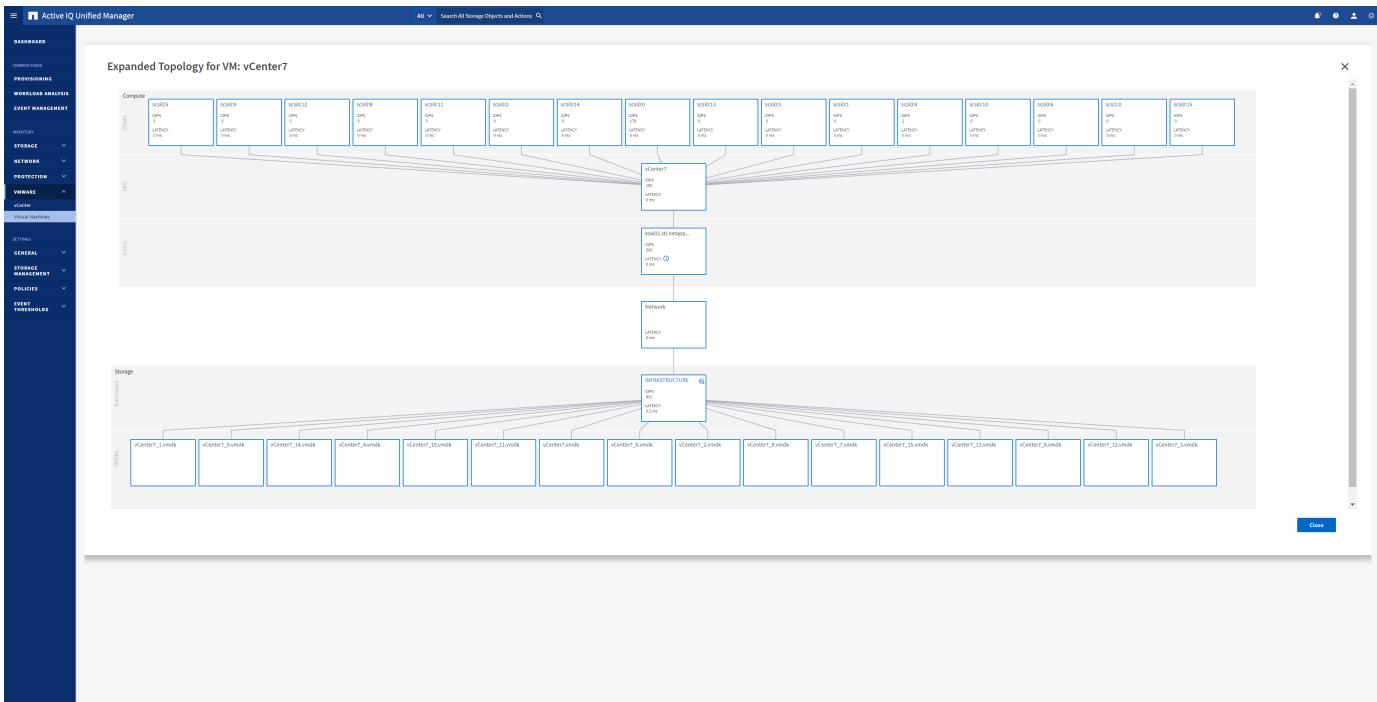
The following screenshot shows the Active IQ Unified Manager Virtual Machines view.

Name	Status	Power State	Protocol	Capacity (Used   Allocated)	≤ 4 IOPS	VM Latency (ms)	Host IOPS	Host Latency (ms)	Network Latency (ms)	Datastore IOPS	Datastore Latency (ms)
vCenter7	ON	NFS		160 GB   712 GB	183	0	243	0	0	831	0.3
VCENTER-SERVER											
vcenter7.stl.netapp.com											
<b>Compute</b>											
VDISK (16)											
Worst Latency/VDisk											
VM											
vCenter7											
HOST											
esxi02.stl.netapp.com											
NETWORK											
LATENCY											
IOPS											
HOST											
esxi02.stl.netapp.com											
LATENCY											
IOPS											
DATASTORE											
INFRASTRUCTURE											
VMDK (16)											
IOPS											
LATENCY											
<b>Storage</b>											
DATASTORE INFRASTRUCTURE											
VMDK (16)											
IOPS											
LATENCY											
AD	ON	NFS		8.05 GB   100 GB	167	0	306	0	0	831	0.3
BluePaddle-01	ON	NFS		398 GB   2.26 TB	44	0	149	0	0	831	0.3
AIQUM	ON	NFS		92 GB   400 GB	41	0	149	0	0	831	0.3
DirtWolf-02	ON	NFS		138 GB   2.26 TB	39	0	306	0	0	831	0.3
BluePaddle-02	ON	NFS		398 GB   2.26 TB	38	0	149	0	0	831	0.3

Showing all 44 Virtual Machines

Unified Manager presents the underlying sub-system of a virtual environment in a topological view for determining whether a latency issue has occurred in the compute node, network, or storage. The view also highlights the specific object that causes the performance lag for taking remedial steps and addressing the underlying issue.

The following screenshot shows the AIQUM expanded topology.



## ONTAP and vSphere release-specific information

This section provides guidance on capabilities supported by specific releases of ONTAP and vSphere. NetApp recommends confirming a specific combination of releases with the [NetApp Interoperability Matrix](#).

### ONTAP releases

At the time of publication, NetApp provides full support for these release families:

- ONTAP 9.5
- ONTAP 9.6
- ONTAP 9.7
- ONTAP 9.8

### vSphere and ESXi support

NetApp ONTAP has broad support for vSphere ESXi hosts. The four major release families just described (9.5, 9.6, 9.7, and 9.8) are fully supported as data storage platforms for recent vSphere releases, including 6.0, 6.5, and 7.0 (including updates for these releases). NFS v3 interoperability is broadly defined, and NetApp supports any client, including hypervisors, that is compliant with the NFS v3 standard. NFSv4.1 support is limited to vSphere 6.0 through 7.0.

For SAN environments, NetApp conducts extensive testing of SAN components. In general, NetApp supports standard X86-64 rack servers and Cisco UCS servers together with standard Ethernet adapters for iSCSI connections. FC, FCoE, and NVMe/FC environments have more specifically defined support due to the HBA firmware and drivers needed.

Always check the [NetApp Interoperability Matrix](#) to confirm support for a specific hardware and software configuration.

## NFS Plug-In for VMware VAAI

This plug-in for ESXi hosts helps by offloading operations to ONTAP using VAAI. The latest release, 1.1.2, includes support for NFSv4.1 datastores, including Kerberos (krb5 and krb5i) support. It is supported with ESXi 6.0, 6.5, and 7.0 together with ONTAP 9.5-9.8.

## VASA Provider

NetApp's VASA Provider supports vVol provisioning and management (see section 3.7). Recent VASA Provider releases support ESXi 6.0, 6.5, and 7.0 together with ONTAP 9.5-9.8.

## ONTAP tools for VMware vSphere

ONTAP tools for VMware vSphere is key for managing ONTAP storage together with vSphere (using it is a best practice). The latest release, 9.8, is supported with vSphere 6.5 and 7.0 together with ONTAP 9.5-9.8.

## Recommended ESXi host and other ONTAP settings

NetApp has developed a set of ESXi host multipathing and HBA timeout settings for proper behavior with ONTAP based on NetApp testing. These are easily set using ONTAP tools for VMware vSphere. From the Summary dashboard, click Edit Settings in the Host Systems portlet or right-click the host in vCenter, then navigate to ONTAP tools > Set Recommended Values. Here are the currently recommended host settings with the 9.8 release.

Host setting
NetApp recommended value
ESXi advanced configuration
VMFS3.HardwareAcceleratedLocking
Leave as set (VMware default is 1).
VMFS3.EnableBlockDelete
Leave as set (VMware default is 0, but this is not needed for VMFS6). For more information, see VMware KB article .
NFS Settings
Net.TcpipHeapSize
vSphere 6.0 or later, set to 32. All other NFS configurations, set to 30.
Net.TcpipHeapMax
Set to 1536 for vSphere 6.0 and later.
NFS.MaxVolumes
vSphere 6.0 or later, set to 256. All other NFS configurations, set to 64.
NFS41.MaxVolumes
vSphere 6.0 or later, set to 256.

NFS.MaxQueueDepth	vSphere 6.0 or later, set to 128.
NFS.HeartbeatMaxFailures	Set to 10 for all NFS configurations.
NFS.HeartbeatFrequency	Set to 12 for all NFS configurations.
NFS.HeartbeatTimeout	Set to 5 for all NFS configurations.
SunRPC.MaxConnPerIP	vSphere 7.0 or later, set to 128.
FC/FCoE Settings	
Path selection policy	Set to RR (round robin) when FC paths with ALUA are used. Set to FIXED for all other configurations. Setting this value to RR helps provide load balancing across all active/optimized paths. The value FIXED is for older, non-ALUA configurations and helps prevent proxy I/O. In other words, it helps keep I/O from going to the other node of a high-availability (HA) pair in an environment that has Data ONTAP operating in 7-Mode.
Disk.QFullSampleSize	Set to 32 for all configurations. Setting this value helps prevent I/O errors.
Disk.QFullThreshold	Set to 8 for all configurations. Setting this value helps prevent I/O errors.
Emulex FC HBA timeouts	Use the default value.
QLogic FC HBA timeouts	Use the default value.
iSCSI Settings	
Path selection policy	Set to RR (round robin) for all iSCSI paths. Setting this value to RR helps provide load balancing across all active/optimized paths.
Disk.QFullSampleSize	Set to 32 for all configurations. Setting this value helps prevent I/O errors.
Disk.QFullThreshold	Set to 8 for all configurations. Setting this value helps prevent I/O errors.

ONTAP tools also specify certain default settings when creating ONTAP FlexVol volumes and LUNs:

ONTAP tool
------------

Default setting
Snapshot reserve (-percent-snapshot-space)
0
Fractional reserve (-fractional-reserve)
0
Access time update (-atime-update)
False
Minimum readahead (-min-readahead)
False
Scheduled Snapshot copies
None
Storage efficiency
Enabled
Volume guarantee
None (thin provisioned)
Volume Autosize
grow_shrink
LUN space reservation
Disabled
LUN space allocation
Enabled

#### Other host multipath configuration considerations

While not currently configured by available ONTAP tools, NetApp suggests considering these configuration options:

- In high-performance environments or when testing performance with a single LUN datastore, consider changing the load balance setting of the round-robin (VMW\_PSP\_RR) path selection policy (PSP) from the default IOPS setting of 1000 to a value of 1. See VMware KB [2069356](#) for more info.
- In vSphere 6.7 Update 1, VMware introduced a new latency load balance mechanism for the Round Robin PSP. The new option considers I/O bandwidth and path latency when selecting the optimal path for I/O. You might benefit from using it in environments with non-equivalent path connectivity, such as cases where there are more network hops on one path than another, or when using a NetApp All SAN Array system. See [Path Selection Plug-Ins and Policies](#) for more information.

#### Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- VMware Product Documentation

<https://www.vmware.com/support/pubs/>

- NetApp Product Documentation  
<https://docs.netapp.com>

## Contact us

Do you have comments about this technical report?

Send them to us at [doccomments@netapp.com](mailto:doccomments@netapp.com) and include TR-4597 in the subject line.

## TR-4900: VMware Site Recovery Manager with NetApp ONTAP 9

Chance Bingen, NetApp

### ONTAP for vSphere

NetApp ONTAP has been a leading storage solution for VMware vSphere environments since its introduction into the modern datacenter in 2002, and it continues to add innovative capabilities to simplify management while reducing costs. This document introduces the ONTAP solution for VMware Site Recovery Manager (SRM), VMware's industry leading disaster recovery (DR) software, including the latest product information and best practices to streamline deployment, reduce risk, and simplify ongoing management.

Best practices supplement other documents such as guides and compatibility tools. They are developed based on lab testing and extensive field experience by NetApp engineers and customers. In some cases, recommended best practices might not be the right fit for your environment; however, they are generally the simplest solutions that meet the needs of the most customers.

This document is focused on capabilities in recent releases of ONTAP 9 when used in conjunction with supported versions of ONTAP tools for VMware vSphere (which includes the NetApp Storage Replication Adapter [SRA] and VASA Provider [VP]), as well as VMware Site Recovery Manager 8.4.

### Why use ONTAP with SRM?

NetApp data management platforms powered by ONTAP software are some of the most widely adopted storage solutions for SRM. The reasons are plentiful: A secure, high performance, unified protocol (NAS and SAN together) data management platform that provides industry defining storage efficiency, multitenancy, quality of service controls, data protection with space-efficient Snapshot copies and replication with SnapMirror. All leveraging native hybrid multi-cloud integration for the protection of VMware workloads and a plethora of automation and orchestration tools at your fingertips.

When you use SnapMirror for array-based replication, you take advantage of one of ONTAP's most proven and mature technologies. SnapMirror gives you the advantage of secure and highly efficient data transfers, copying only changed file system blocks, not entire VMs or datastores. Even those blocks take advantage of space savings, such as deduplication, compression, and compaction. Modern ONTAP systems now use version-independent SnapMirror, allowing you flexibility in selecting your source and destination clusters. SnapMirror has truly become one of the most powerful tools available for disaster recovery.

Whether you are using traditional NFS, iSCSI, or Fibre Channel- attached datastores (now with support for vVols datastores), SRM provides a robust first party offering that leverages the best of ONTAP capabilities for disaster recovery or datacenter migration planning and orchestration.

### How SRM leverages ONTAP 9

SRM leverages the advanced data management technologies of ONTAP systems by integrating with ONTAP

tools for VMware vSphere, a virtual appliance that includes three primary components:

- The vCenter plug-in, formerly known as Virtual Storage Console (VSC), simplifies storage management and efficiency features, enhances availability, and reduces storage costs and operational overhead, whether you are using SAN or NAS. It uses best practices for provisioning datastores and optimizes ESXi host settings for NFS and block storage environments. For all these benefits, NetApp recommends this plug-in when using vSphere with systems running ONTAP software.
- The VASA Provider for ONTAP supports the VMware vStorage APIs for Storage Awareness (VASA) framework. VASA Provider connects vCenter Server with ONTAP to aid in provisioning and monitoring VM storage. It enables VMware Virtual Volumes (vVols) support and the management of storage capability profiles (including vVols replication capabilities) and individual VM vVols performance. It also provides alarms for monitoring capacity and compliance with the profiles. When used in conjunction with SRM, the VASA Provider for ONTAP enables support for vVols-based virtual machines without requiring the installation of an SRA adapter on the SRM server.
- The SRA is used together with SRM to manage the replication of VM data between production and disaster recovery sites for traditional VMFS and NFS datastores and also for the nondisruptive testing of DR replicas. It helps automate the tasks of discovery, recovery, and reprotection. It includes both an SRA server appliance and SRA adapters for the Windows SRM server and the SRM appliance.

After you have installed and configured the SRA adapters on the SRM server for protecting non-vVols datastores and/or enabled vVols replication in the VASA Provider settings, you can begin the task of configuring your vSphere environment for disaster recovery.

The SRA and VASA Provider deliver a command-and-control interface for the SRM server to manage the ONTAP FlexVols that contain your VMware Virtual Machines (VMs), as well as the SnapMirror replication protecting them.

Starting with SRM 8.3, a new SRM vVols Provider control path was introduced into the SRM server, allowing it to communicate with the vCenter server and, through it, to the VASA Provider without needing an SRA. This enabled the SRM server to leverage much deeper control over the ONTAP cluster than was possible before, because VASA provides a complete API for closely coupled integration.

SRM can test your DR plan nondisruptively using NetApp's proprietary FlexClone technology to make nearly instantaneous clones of your protected datastores at your DR site. SRM creates a sandbox to safely test so that your organization, and your customers, are protected in the event of a true disaster, giving you confidence in your organizations ability to execute a failover during a disaster.

In the event of a true disaster or even a planned migration, SRM allows you to send any last-minute changes to the dataset via a final SnapMirror update (if you choose to do so). It then breaks the mirror and mounts the datastore to your DR hosts. At that point, your VMs can be automatically powered up in any order according to your pre-planned strategy.

### **SRM with ONTAP and other use cases: hybrid cloud and migration**

Integrating your SRM deployment with ONTAP advanced data management capabilities allows for vastly improved scale and performance when compared with local storage options. But more than that, it brings the flexibility of the hybrid cloud. The hybrid cloud enables you to save money by tiering unused data blocks from your high-performance array to your preferred hyperscaler using FabricPool, which could be an on-premises S3 store such as NetApp StorageGRID. You can also use SnapMirror for edge-based systems with software-defined ONTAP Select or cloud-based DR using Cloud Volumes ONTAP (CVO) or [NetApp Private Storage in Equinix](#) for Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) to create a fully integrated storage, networking, and compute- services stack in the cloud.

You could then perform test failover inside a cloud service provider's datacenter with near-zero storage

footprint thanks to FlexClone. Protecting your organization can now cost less than ever before.

SRM can also be used to execute planned migrations by leveraging SnapMirror to efficiently transfer your VMs from one datacenter to another or even within the same datacenter, whether your own, or via any number of NetApp partner service providers.

## New features with SRM and ONTAP Tools

With the transition from the legacy virtual appliance, ONTAP tools brings a wealth of new features, higher limits, and new vVols support.

### Latest versions of vSphere and Site Recovery Manager

With the release of SRM 8.3 and later and the 9.7.1 and later releases of ONTAP tools, you are now able to protect VMs running on VMware vSphere 7.

NetApp has shared a deep partnership with VMware for nearly two decades and strives to provide support for the latest releases as soon as possible. Always check the NetApp Interoperability Matrix Tool (IMT) for the latest qualified combinations of software.

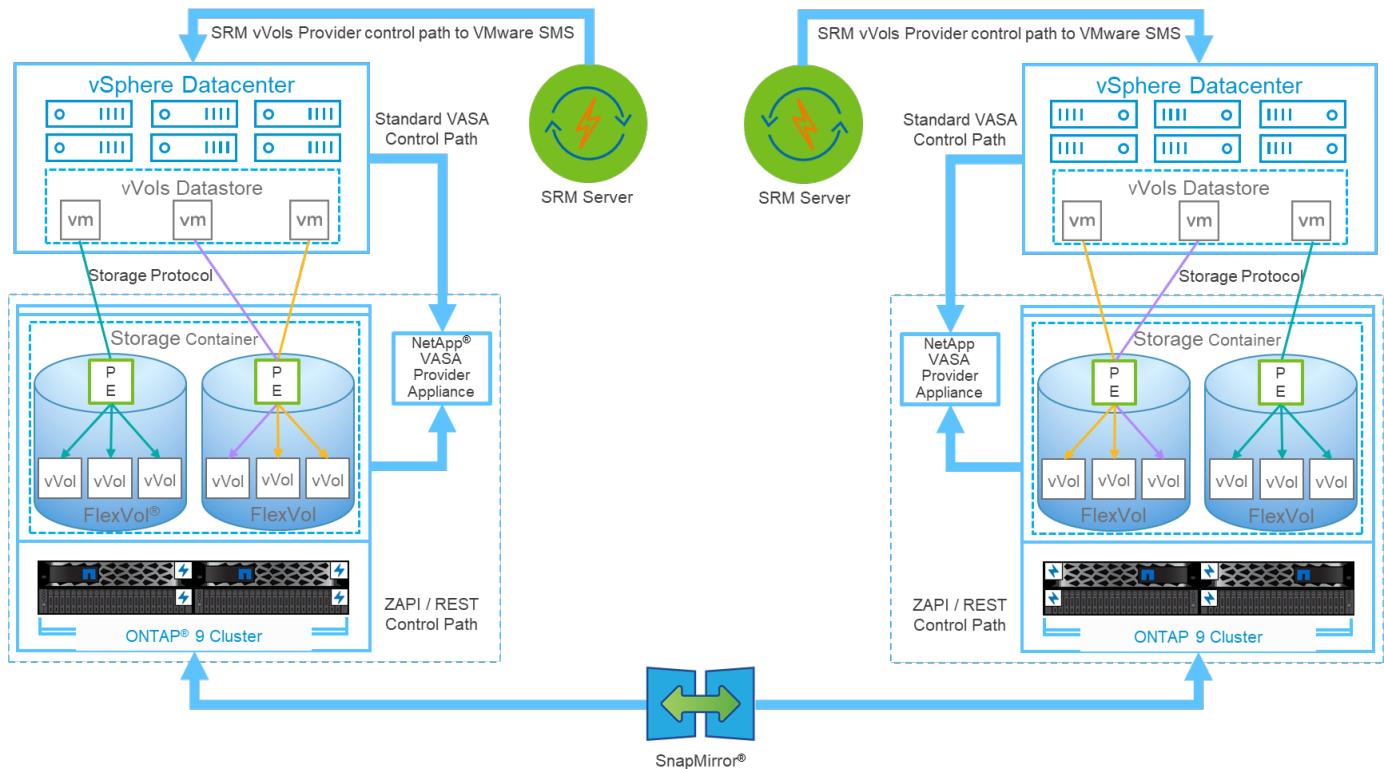
The NetApp IMT can be found [here](#).

### vVols support (and why SPBM matters, even with SRM)

Starting with the 8.3 release, SRM now supports storage policy-based management (SPBM) of replication leveraging vVols and array-based replication. To accomplish this, the SRM server was updated to include a new SRM vVols provider service, which communicates to the vCenter server's SMS service for VASA related tasks.

One advantage to this architecture is that an SRA is no longer needed since everything is handled using VASA.

SPBM is a powerful tool in the vSphere toolbox, allowing simplified, predictable, and consistent storage services for consumption by automation frameworks in private and hybrid cloud environments. Fundamentally, SPBM allows you to define classes of service that meet the needs of your diverse customer base. SRM now allows you to expose replication capabilities to your customers for critical workloads requiring robust industry-standard disaster-recovery orchestration and automation.



### vVols Architecture 2.3 Support for appliance-based SRM servers

Photon OS-based SRM servers are now supported, in addition to legacy Windows-based platforms.

You can now install SRA adapters regardless of your preferred SRM server type.

### Support for IPv6

IPv6 is now supported with the following limitations:

- vCenter 6.7 or later
- Not supported with SRM 8.2 (8.1, 8.3, and 8.4 are supported)
- Check the [Interoperability Matrix Tool](#) for the latest qualified versions.

### Improved performance

Operational performance is a key requirement for SRM task execution. To meet the requirements of modern RTOs and RPOs, the SRA with ONTAP tools has added two new improvements.

- **Support for concurrent reprotect operations.** First introduced in SRA 9.7.1, enabling this feature allows you to run reprotect on two or more recovery plans concurrently, thus reducing the time required to reprotect datastores after a failover or migration and remain within your RTO and RPO parameters.
- **ONTAP Tools 9.8 adds a new NAS- only optimized mode.** When you use SVM- scoped accounts and connections to ONTAP clusters with only NFS based datastores, you can enable NAS-only optimized mode for peak performance in supported environments.

### Greater scale

The ONTAP tools SRA can now support up to 500 protection groups (PGs) when used with SRM 8.3 and later.

## Synchronous replication

A long awaited and much anticipated new feature is SnapMirror Synchronous (SM-S) with ONTAP 9.5 and later which delivers a volume granular zero RPO data replication solution for your mission-critical applications. SM-S requires ONTAP tools 9.8 or later.

## REST API support

SRA server configuration can now be managed by REST APIs. A Swagger UI has been added to assist in building your automation workflows and can be found on your ONTAP tools appliance at <https://<appliance>:8143/api/rest/swagger-ui.html#/>.

## Deployment best practices

### SVM layout and segmentation for SMT

With ONTAP, the concept of the storage virtual machine (SVM) provides strict segmentation in secure multitenant environments. SVM users on one SVM cannot access or manage resources from another. In this way, you can leverage ONTAP technology by creating separate SVMs for different business units who manage their own SRM workflows on the same cluster for greater overall storage efficiency.

Consider managing ONTAP using SVM-scoped accounts and SVM management LIFs to not only improve security controls, but also improve performance. Performance is inherently greater when using SVM-scoped connections because the SRA is not required to process all the resources in an entire cluster, including physical resources. Instead, it only needs to understand the logical assets that are abstracted to the particular SVM.

When using NAS protocols only (no SAN access), you can even leverage the new NAS optimized mode by setting the following parameter (note that the name is such because SRA and VASA use the same backend services in the appliance):

1. Log into the control panel at <https://<IP address>:9083> and click Web based CLI interface.
2. Run the command `vp updateconfig -key=enable.qtree.discovery -value=true`.
3. Run the command `vp updateconfig -key=enable.optimised.sra -value=true`.
4. Run the command `vp reloadconfig`.

### Deploy ONTAP tools and considerations for vVols

If you intend to use SRM with vVols, you must manage the storage using cluster- scoped credentials and a cluster management LIF. This is because the VASA Provider must understand the underlying physical architecture to satisfy the policy requires for VM storage policies. For example, if you have a policy that requires all- flash storage, the VASA Provider must be able to see which systems are all flash.

Another deployment best practice is to never store your ONTAP tools appliance on a vVols datastore that it is managing. This could lead to a situation whereby you cannot power on the VASA Provider because you cannot create the swap vVol for the appliance because the appliance is offline.

### Best practices for managing ONTAP 9 systems

As previously mentioned, you can manage ONTAP clusters using either cluster or SVM scoped credentials and management LIFs. For optimum performance, you may want to consider using SVM- scoped credentials whenever you aren't using vVols. However, in doing so, you should be aware of some requirements, and that you do lose some functionality.

- The default vsadmin SVM account does not have the required access level to perform ONTAP tools tasks. Therefore, you need to create a new SVM account.
- If you are using ONTAP 9.8 or later, NetApp recommends creating an RBAC least privileged user account using ONTAP System Manager's users menu together with the JSON file available on your ONTAP tools appliance at <https://<IP address>:9083/vsc/config/>. Use your administrator password to download the JSON file. This can be used for SVM or cluster scoped accounts.

If you are using ONTAP 9.6 or earlier, you should use the RBAC User Creator (RUC) tool available in the [NetApp Support Site Toolchest](#).

- Because the vCenter UI plugin, VASA Provider, and SRA server are all fully integrated services, you must add storage to the SRA adapter in SRM the same way you add storage in the vCenter UI for ONTAP tools. Otherwise, the SRA server might not recognize the requests being sent from SRM via the SRA adapter.
- NFS path checking is not performed when using SVM-scoped credentials. This is because the physical location is logically abstracted from the SVM. This is not a cause for concern though, as modern ONTAP systems no longer suffer any noticeable performance decline when using indirect paths.
- Aggregate space savings due to storage efficiency might not be reported.
- Where supported, load-sharing mirrors cannot be updated.
- EMS logging might not be performed on ONTAP systems managed with SVM scoped credentials.

## **Operational best practices**

### **Datastores and protocols**

If possible, always use ONTAP tools to provision datastores and volumes. This makes sure that volumes, junction paths, LUNs, igroups, export policies, and other settings are configured in a compatible manner.

SRM supports iSCSI, Fibre Channel, and NFS version 3 with ONTAP 9 when using array-based replication through SRA. SRM does not support array-based replication for NFS version 4.1 with either traditional or vVols datastores.

To confirm connectivity, always verify that you can mount and unmount a new test datastore at the DR site from the destination ONTAP cluster. Test each protocol you intend to use for datastore connectivity. A best practice is to use ONTAP tools to create your test datastore, since it is doing all the datastore automation as directed by SRM.

SAN protocols should be homogeneous for each site. You can mix NFS and SAN, but the SAN protocols should not be mixed within a site. For example, you can use FCP in site A, and iSCSI in site B. You should not use both FCP and iSCSI at site A. The reason for this is that the SRA does not create mixed igroups at the recovery site and SRM does not filter the initiator list given to the SRA.

Previous guides advised to create LIF to data locality. That is to say, always mount a datastore using a LIF located on the node that physically owns the volume. That is no longer a requirement in modern versions of ONTAP 9. Whenever possible, and if given cluster scoped credentials, ONTAP tools will still choose to load balance across LIFs local to the data, but it is not a requirement for high availability or performance.

NetApp ONTAP 9 can be configured to automatically remove Snapshot copies to preserve uptime in the event of an out-of-space condition when autosize is not able to supply sufficient emergency capacity. The default setting for this capability does not automatically delete the Snapshot copies that are created by SnapMirror. If SnapMirror Snapshot copies are deleted, then the NetApp SRA cannot reverse and resynchronize replication for the affected volume. To prevent ONTAP from deleting SnapMirror Snapshot copies, configure the Snapshot autodelete capability to try.

```
snap autodelete modify -volume -commitment try
```

Volume autosize should be set to `grow` for volumes containing SAN datastores and `grow_shrink` for NFS datastores. Refer to the [ONTAP 9 Documentation Center](#) for specific syntax.

## SPBM and vVols

Starting with SRM 8.3, protection of VMs using vVols datastores is supported. SnapMirror schedules are exposed to VM storage policies by the VASA Provider when vVols replication is enabled in the ONTAP tools settings menu, as shown in the following screenshots.

The following example show the enablement of vVols replication.

## Manage Capabilities



### Enable VASA Provider

vStorage APIs for Storage Awareness (VASA) is a set of application program interfaces (APIs) that enables vSphere vCenter to recognize the capabilities of storage arrays.



### Enable vVols replication

Enables replication of vVols when used with VMware Site Recovery Manager 8.3 or later.



### Enable Storage Replication Adapter (SRA)

Storage Replication Adapter (SRA) allows VMware Site Recovery Manager (SRM) to integrate with third party storage array technology.

Enter authentication details for VASA Provider and SRA server:

IP address or hostname: 192.168.64.7

Username: Administrator

Password: \_\_\_\_\_

CANCEL

APPLY

The following screenshot provides an example of SnapMirror schedules displayed in the Create VM Storage Policy wizard.

## Create VM Storage Policy

- 1 Name and description
- 2 Policy structure
- 3 NetApp.clustered.Data.ONTAP.VP...**
- 4 Storage compatibility
- 5 Review and finish

## NetApp.clustered.Data.ONTAP.VP.vvol rules

Placement    Replication    Tags

Disabled  
 Custom

Provider: NetApp.clustered.Data.ONTAP.VP.vvolReplication

Replication	Asynchronous	REMOVE
Replication Schedule	[Select Value] [Select Value] <b>hourly</b>	REMOVE

CANCEL    BACK    **NEXT**

The ONTAP VASA Provider supports failover to dissimilar storage. For example, the system can fail over from ONTAP Select at an edge location to an AFF system in the core datacenter. Regardless of storage similarity, you must always configure storage policy mappings and reverse mappings for replication-enabled VM storage policies to make sure that services provided at the recovery site meet expectations and requirements. The following screenshot highlights a sample policy mapping.

## New Storage Policy Mappings

- 1 Creation mode
- 2 Recovery storage policies**
- 3 Reverse mappings
- 4 Ready to complete

## Recovery storage policies

Configure recovery storage policy mappings for one or more storage policies.

Search...

- vc1.demo.netapp.com
- Host-local PMem Default Storage Policy
- VC1 Storage Policy \*
- VM Encryption Policy
- vSAN Default Storage Policy
- VVol No Requirements Policy

Search...

- vc2.demo.netapp.com
- Host-local PMem Default Storage Policy
- VC2 Storage Policy
- VM Encryption Policy
- vSAN Default Storage Policy

**↓ ADD MAPPINGS**

vc1.demo.netapp.com	vc2.demo.netapp.com
: VC1 Storage Policy	: VC2 Storage Policy
1 mapping(s)	

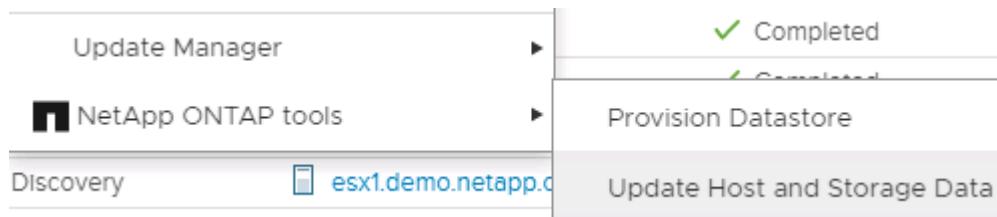
CANCEL    BACK    **NEXT**

## Create replicated volumes for vVols datastores

Unlike previous vVols datastores, replicated vVols datastores must be created from the start with replication enabled, and they must use volumes that were pre-created on the ONTAP systems with SnapMirror

relationships. This requires pre-configuring things like cluster peering and SVM peering. These activities should be performed by your ONTAP administrator, because this facilitates a strict separation of responsibilities between those who manage the ONTAP systems across multiple sites and those who are primarily responsible for vSphere operations.

This does come with a new requirement on behalf of the vSphere administrator. Because volumes are being created outside the scope of ONTAP tools, it is unaware of the changes your ONTAP administrator has made until the regularly scheduled rediscovery period. For that reason, it is a best practice to always run rediscovery whenever you create a volume or SnapMirror relationship to be used with vVols. Simply right click on the host or cluster and select NetApp ONTAP tools > Update Host and Storage Data, as shown in the following screenshot.



One caution should be taken when it comes to vVols and SRM. Never mix protected and unprotected VMs in the same vVols datastore. The reason for this is that when you use SRM to failover to your DR site, only those VMs that are part of the protection group are brought online in DR. Therefore, when you reprotect (reverse the SnapMirror from DR back to production again), you may overwrite the VMs that were not failed over and could contain valuable data.

#### About array pairs

An array manager is created for each array pair. With SRM and ONTAP tools, each array pairing is done with the scope of an SVM, even if you are using cluster credentials. This allows you to segment DR workflows between tenants based on which SVMs they have been assigned to manage. You can create multiple array managers for a given cluster, and they can be asymmetric in nature. You can fan out or fan in between different ONTAP 9 clusters. For example, you can have SVM-A and SVM-B on Cluster-1 replicating to SVM-C on Cluster-2, SVM-D on Cluster-3, or vice-versa.

When configuring array pairs in SRM, you should always add them in SRM the same way as you added them to ONTAP Tools, meaning, they must use the same username, password, and management LIF. This requirement ensures that SRA communicates properly with the array. The following screenshot illustrates how a cluster might appear in ONTAP Tools and how it might be added to an array manager.

The screenshot shows the vSphere Client interface with the 'Storage Systems' tab selected. The main pane displays a table with columns 'Name', 'Type', and 'IP Address'. One row is selected, showing 'cluster2' under 'Name', 'Cluster' under 'Type', and 'cluster2.demo.netapp.com' under 'IP Address'. Below this, a red arrow points from the 'cluster2' entry in the table to the 'cluster2 demo.netapp.com' input field in the 'Edit Local Array Manager' dialog.

## Edit Local Array Manager

Enter a name for the array manager on "vc2.demo.netapp.com":

vc2\_array\_manager

Storage Array Parameters

Storage Management IP Address or Hostname

cluster2 demo.netapp.com

Enter the cluster management IP address/hostname. To connect directly to a Storage Virtual Machine(SVM), enter the SVM management IP address/hostname.

## About replication groups

Replication groups contain logical collections of virtual machines that are recovered together. The ONTAP tools VASA Provider automatically creates replication groups for you. Because ONTAP SnapMirror replication occurs at the volume level, all VMs in a volume are in the same replication group.

There are several factors to consider with replication groups and how you distribute VMs across FlexVol volumes. Grouping similar VMs in the same volume can increase storage efficiency with older ONTAP systems that lack aggregate-level deduplication, but grouping increases the size of the volume and reduces volume I/O concurrency. The best balance of performance and storage efficiency can be achieved in modern ONTAP systems by distributing VMs across FlexVol volumes in the same aggregate, thereby leveraging aggregate level deduplication and gaining greater I/O parallelization across multiple volumes. You can recover VMs in the volumes together because a protection group (discussed below) can contain multiple replication groups. The downside to this layout is that blocks might be transmitted over the wire multiple times because volume SnapMirror doesn't take aggregate deduplication into account.

One final consideration for replication groups is that each one is by its nature a logical consistency group (not to be confused with SRM consistency groups). This is because all VMs in the volume are transferred together using the same snapshot. So if you have VMs that must be consistent with each other, consider storing them in the same FlexVol.

## About protection groups

Protection groups define VMs and datastores in groups that are recovered together from the protected site. The protected site is where the VMs that are configured in a protection group exist during normal steady-state operations. It is important to note that even though SRM might display multiple array managers for a protection group, a protection group cannot span multiple array managers. For this reason, you should not span VM files across datastores on different SVMs.

## About recovery plans

Recovery plans define which protection groups are recovered in the same process. Multiple protection groups can be configured in the same recovery plan. Also, to enable more options for the execution of recovery plans,

a single protection group can be included in multiple recovery plans.

Recovery plans allow SRM administrators to define recovery workflows by assigning VMs to a priority group from 1 (highest) to 5 (lowest), with 3 (medium) being the default. Within a priority group, VMs can be configured for dependencies.

For example, your company could have a tier-1 business critical application that relies on a Microsoft SQL server for its database. So, you decide to place your VMs in priority group 1. Within priority group 1, you begin planning the order to bring up services. You probably want your Microsoft Windows domain controller to boot up before your Microsoft SQL server, which would need to be online before your application server, and so on. You would add all these VMs to the priority group and then set the dependencies, because dependencies only apply within a given priority group.

NetApp strongly recommends working with your application teams to understand the order of operations required in a failover scenario and to construct your recovery plans accordingly.

#### **Test failover**

As a best practice, always perform a test failover whenever a change is made to the configuration of a protected VM storage. This ensures that, in the event of a disaster, you can trust that Site Recovery Manager is able to restore services within the expected RTO target.

NetApp also recommends confirming in-guest application functionality occasionally, especially after reconfiguring VM storage.

When a test recovery operation is performed, a private test bubble network is created on the ESXi host for the VMs. However, this network is not automatically connected to any physical network adapters and therefore does not provide connectivity between the ESXi hosts. To allow communication among VMs that are running on different ESXi hosts during DR testing, a physical private network is created between the ESXi hosts at the DR site. To verify that the test network is private, the test bubble network can be separated physically or by using VLANs or VLAN tagging. This network must be segregated from the production network because as the VMs are recovered, they cannot be placed on the production network with IP addresses that could conflict with actual production systems. When a recovery plan is created in SRM, the test network that was created can be selected as the private network to connect the VMs to during the test.

After the test has been validated and is no longer required, perform a cleanup operation. Running cleanup returns the protected VMs to their initial state and resets the recovery plan to the Ready state.

#### **Failover considerations**

There are several other considerations when it comes to failing over a site in addition to the order of operations mentioned in this guide.

One issue you might have to contend with is networking differences between sites. Some environments might be able to use the same network IP addresses at both the primary site and the DR site. This ability is referred to as a stretched virtual LAN (VLAN) or stretched network setup. Other environments might have a requirement to use different network IP addresses (for example, in different VLANs) at the primary site relative to the DR site.

VMware offers several ways to solve this problem. For one, network virtualization technologies like VMware NSX-T Data Center abstract the entire networking stack from layers 2 through 7 from the operating environment, allowing for more portable solutions. You can read more about NSX-T options with SRM [here](#).

SRM also gives you the ability to change the network configuration of a VM as it is recovered. This reconfiguration includes settings such as IP addresses, gateway address, and DNS server settings. Different

network settings, which are applied to individual VMs as they are recovered, can be specified in the property's settings of a VM in the recovery plan.

To configure SRM to apply different network settings to multiple VMs without having to edit the properties of each one in the recovery plan, VMware provides a tool called the dr-ip-customizer. For information on how to use this utility, refer to VMware's documentation [here](#).

### **Reprotect**

After a recovery, the recovery site becomes the new production site. Because the recovery operation broke the SnapMirror replication, the new production site is not protected from any future disaster. A best practice is to protect the new production site to another site immediately after a recovery. If the original production site is operational, the VMware administrator can use the original production site as a new recovery site to protect the new production site, effectively reversing the direction of protection. Reprotection is available only in non-catastrophic failures. Therefore, the original vCenter Servers, ESXi servers, SRM servers, and corresponding databases must be eventually recoverable. If they are not available, a new protection group and a new recovery plan must be created.

### **Failback**

A failback operation is fundamentally a failover in a different direction than before. As a best practice, you verify that the original site is back to acceptable levels of functionality before attempting to failback, or, in other words, failover to the original site. If the original site is still compromised, you should delay failback until the failure is sufficiently remediated.

Another failback best practice is to always perform a test failover after completing reprotect and before doing your final failback. This verifies that the systems in place at the original site can complete the operation.

#### **Reprotecting the original site**

After failback, you should confirm with all stakeholders that their services have been returned to normal before running reprotect again.

Running reprotect after failback essentially puts the environment back in the state it was in at the beginning, with SnapMirror replication again running from the production site to the recovery site.

### **Replication topologies**

In ONTAP 9, the physical components of a cluster are visible to cluster administrators, but they are not directly visible to the applications and hosts that use the cluster. The physical components provide a pool of shared resources from which the logical cluster resources are constructed. Applications and hosts access data only through SVMs that contain volumes and LIFs.

Each NetApp SVM is treated as an array in VMware vCenter Site Recovery Manager. SRM supports certain array-to-array (or SVM-to-SVM) replication layouts.

A single VM cannot own data—Virtual Machine Disk (VMDK) or RDM—on more than one SRM array for the following reasons:

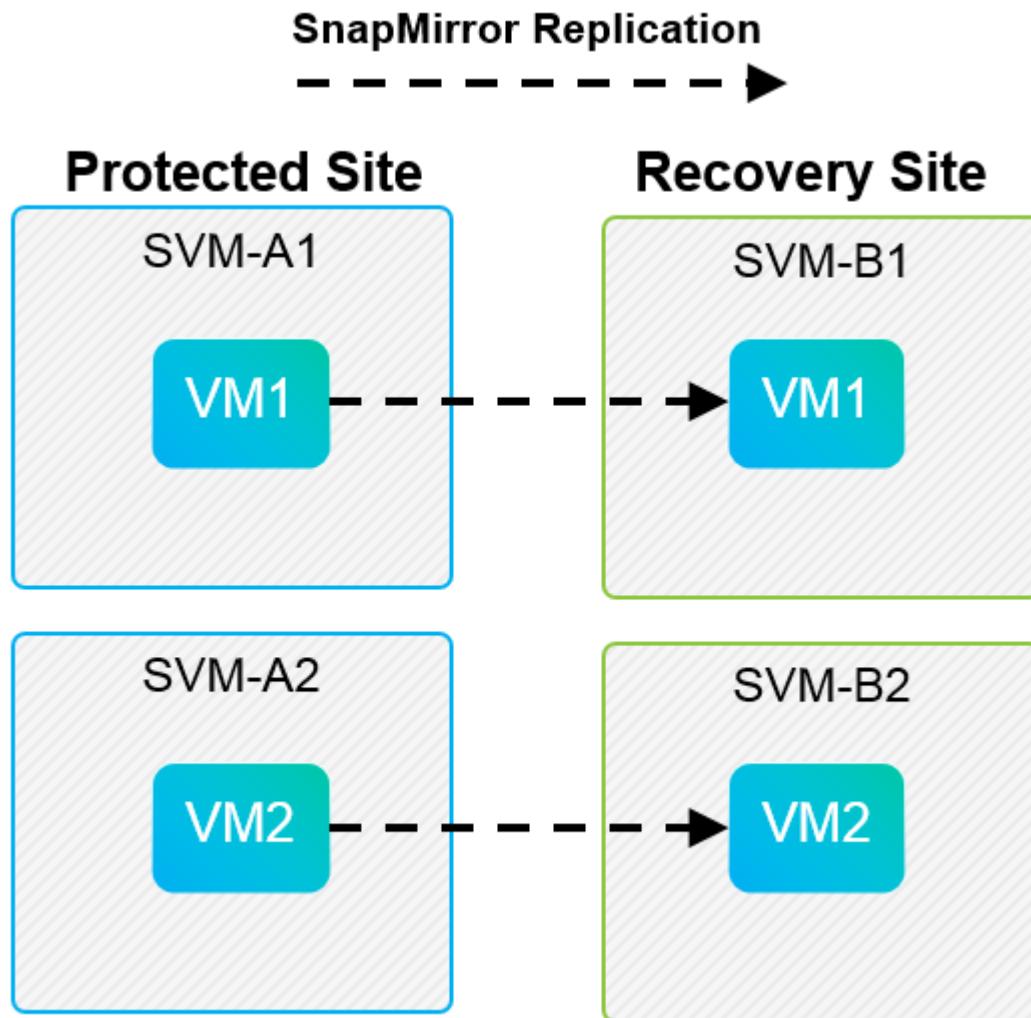
- SRM sees only the SVM, not an individual physical controller.
- An SVM can control LUNs and volumes that span multiple nodes in a cluster.

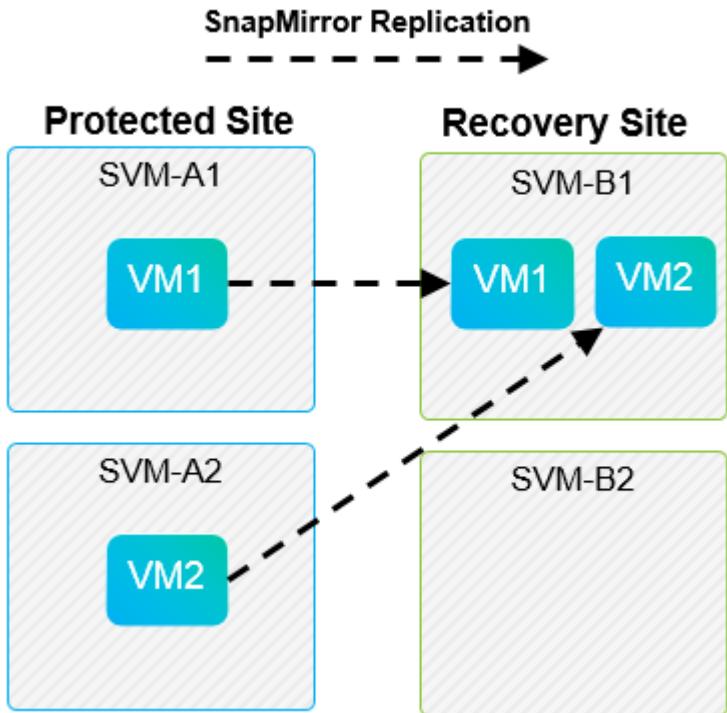
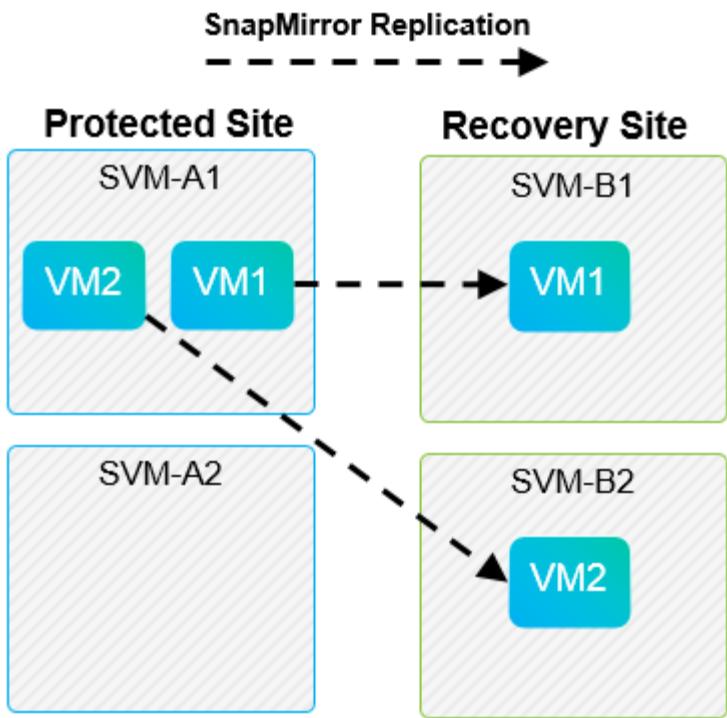
### **Best Practice**

To determine supportability, keep this rule in mind: to protect a VM by using SRM and the NetApp SRA, all parts of the VM must exist on only one SVM. This rule applies at both the protected site and the recovery site.

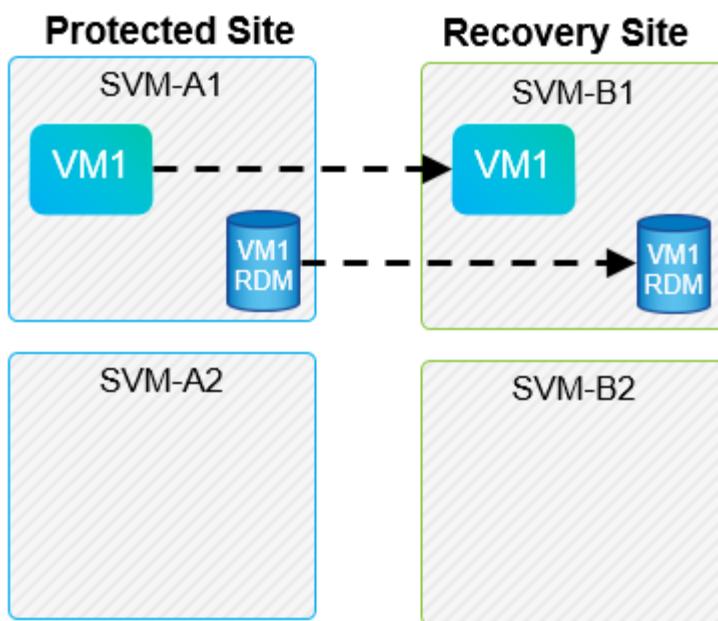
#### **Supported SnapMirror layouts**

The following figures show the SnapMirror relationship layout scenarios that SRM and SRA support. Each VM in the replicated volumes owns data on only one SRM array (SVM) at each site.





## SnapMirror Replication



### Supported Array Manager layouts

When you use array-based replication (ABR) in SRM, protection groups are isolated to a single array pair, as shown in the following screenshot. In this scenario, **SVM1** and **SVM2** are peered with **SVM3** and **SVM4** at the recovery site. However, you can select only one of the two array pairs when you create a protection group.

New Protection Group

1 Name and direction

**2 Type**

3 Datastore groups

4 Recovery plan

5 Ready to complete

Type

Select the type of protection group you want to create:

Datastore groups (array-based replication)  
Protect all virtual machines which are on specific datastores.

Individual VMs (vSphere Replication)  
Protect specific virtual machines, regardless of the datastores.

Virtual Volumes (vVol replication)  
Protect virtual machines which are on replicated vVol storage.

Storage policies (array-based replication)  
Protect virtual machines with specific storage policies.

Select array pair

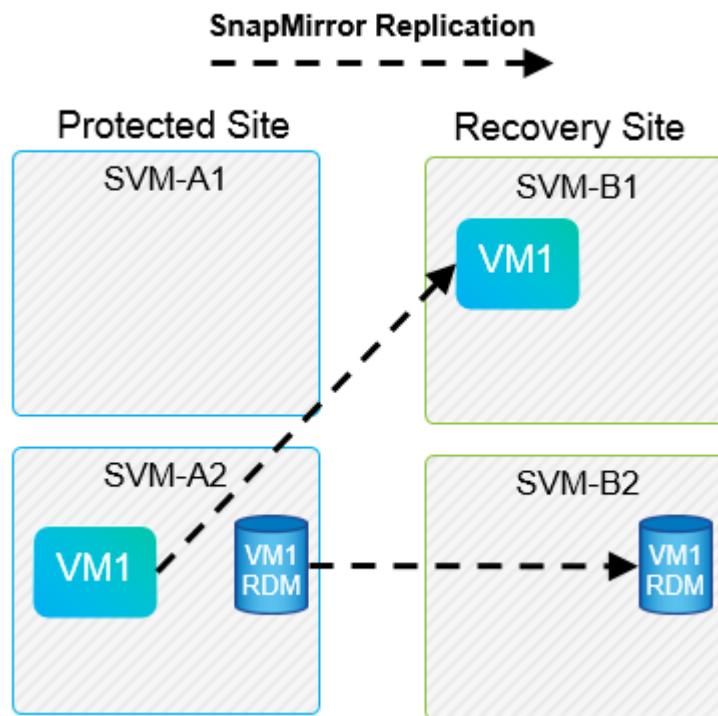
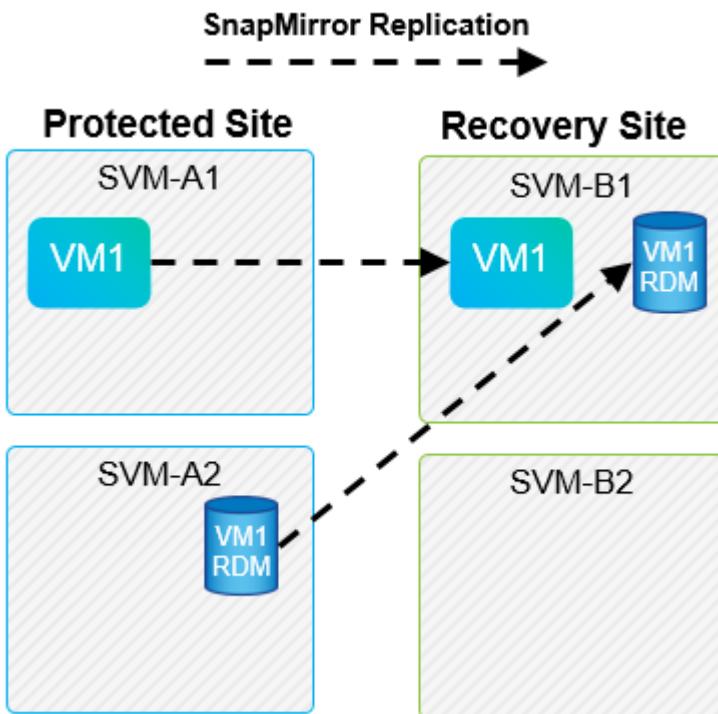
Array Pair	Array Manager Pair
<input type="radio"/> ✓ cluster1:svm1 ↔ cluster2:svm2	vc1 array manager ↔ vc2 array manager
<input type="radio"/> ✓ cluster1:svm3 ↔ cluster2:svm4	vc1 trad datastores ↔ vc2 trad datastores

CANCEL BACK NEXT

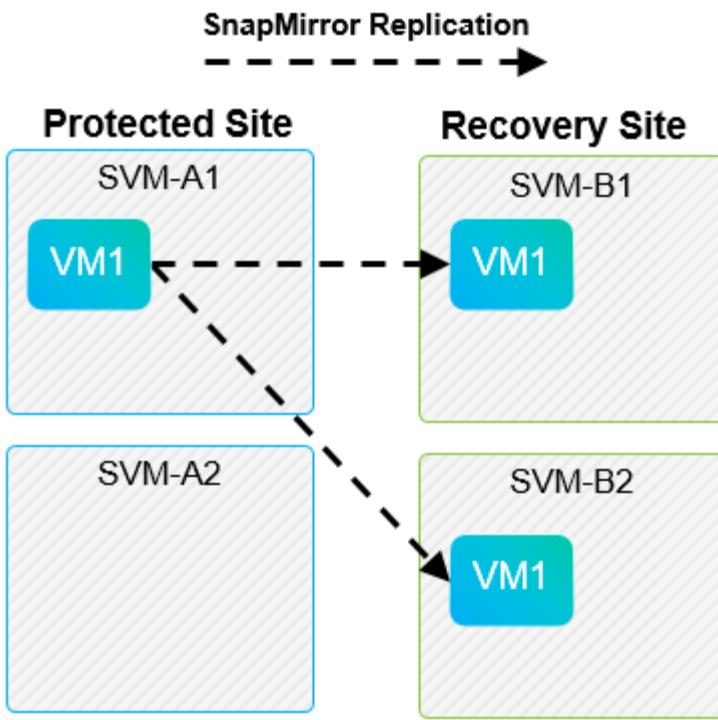
### Unsupported layouts

Unsupported configurations have data (VMDK or RDM) on multiple SVMs that is owned by an individual VM. In

In the examples shown in the following figures, **VM1** cannot be configured for protection with SRM because **VM1** has data on two SVMs.

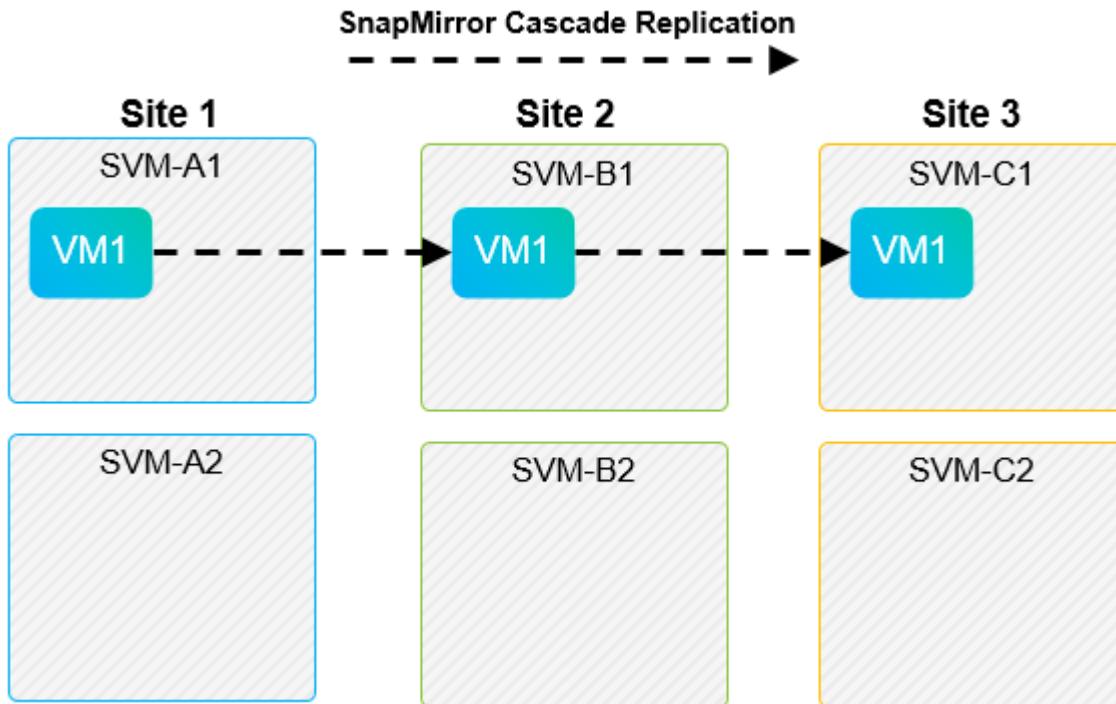


Any replication relationship in which an individual NetApp volume is replicated from one source SVM to multiple destinations in the same SVM or in different SVMs is referred to as SnapMirror fan-out. Fan-out is not supported with SRM. In the example shown in the following figure, **VM1** cannot be configured for protection in SRM because it is replicated with SnapMirror to two different locations.



#### SnapMirror cascade

SRM does not support cascading of SnapMirror relationships, in which a source volume is replicated to a destination volume and that destination volume is also replicated with SnapMirror to another destination volume. In the scenario shown in the following figure, SRM cannot be used for failover between any sites.



#### SnapMirror and SnapVault

NetApp SnapVault software enables disk-based backup of enterprise data between NetApp storage systems. SnapVault and SnapMirror can coexist in the same environment; however, SRM supports the failover of only

the SnapMirror relationships.



The NetApp SRA supports the `mirror-vault` policy type.

SnapVault was rebuilt from the ground up for ONTAP 8.2. Although former Data ONTAP 7-Mode users should find similarities, major enhancements have been made in this version of SnapVault. One major advance is the ability to preserve storage efficiencies on primary data during SnapVault transfers.

An important architectural change is that SnapVault in ONTAP 9 replicates at the volume level as opposed to at the qtree level, as is the case in 7-Mode SnapVault. This setup means that the source of a SnapVault relationship must be a volume, and that volume must replicate to its own volume on the SnapVault secondary system.

In an environment in which SnapVault is used, specifically named Snapshot copies are created on the primary storage system. Depending on the configuration implemented, the named Snapshot copies can be created on the primary system by a SnapVault schedule or by an application such as NetApp Active IQ Unified Manager. The named Snapshot copies that are created on the primary system are then replicated to the SnapMirror destination, and from there they are vaulted to the SnapVault destination.

A source volume can be created in a cascade configuration in which a volume is replicated to a SnapMirror destination in the DR site, and from there it is vaulted to a SnapVault destination. A source volume can also be created in a fan-out relationship in which one destination is a SnapMirror destination and the other destination is a SnapVault destination. However, SRA does not automatically reconfigure the SnapVault relationship to use the SnapMirror destination volume as the source for the vault when SRM failover or replication reversal occurs.

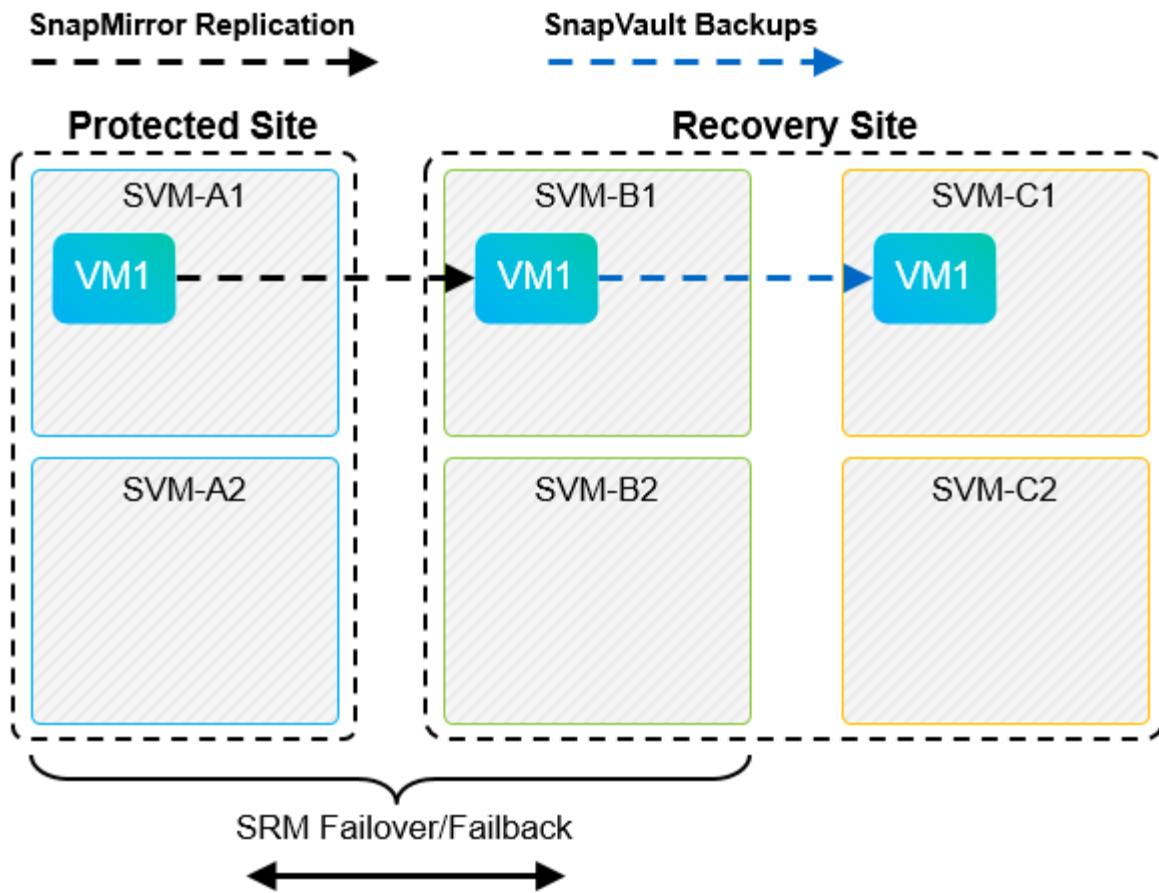
For the latest information about SnapMirror and SnapVault for ONTAP 9, see [TR-4015 SnapMirror Configuration Best Practice Guide for ONTAP 9](#).

### Best Practice

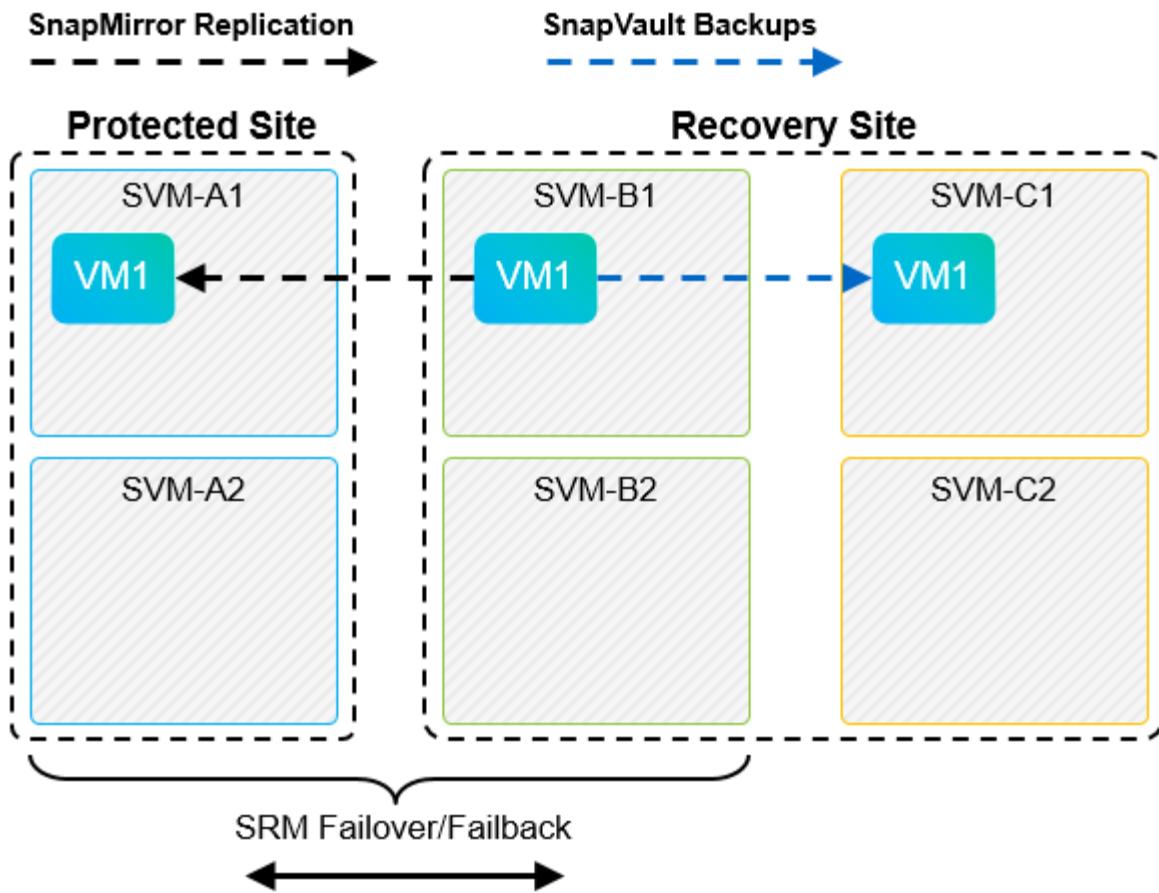
If SnapVault and SRM are used in the same environment, NetApp recommends using a SnapMirror to SnapVault cascade configuration in which SnapVault backups are normally performed from the SnapMirror destination at the DR site. In the event of a disaster, this configuration makes the primary site inaccessible. Keeping the SnapVault destination at the recovery site allows SnapVault backups to be reconfigured after failover so that SnapVault backups can continue while operating at the recovery site.

In a VMware environment, each datastore has a universal unique identifier (UUID), and each VM has a unique managed object ID (MOID). These IDs are not maintained by SRM during failover or fallback. Because datastore UUIDs and VM MOIDs are not maintained during failover by SRM, any applications that depend on these IDs must be reconfigured after SRM failover. An example application is NetApp Active IQ Unified Manager, which coordinates SnapVault replication with the vSphere environment.

The following figure depicts a SnapMirror to SnapVault cascade configuration. If the SnapVault destination is at the DR site or at a tertiary site that is not affected by an outage at the primary site, the environment can be reconfigured to allow backups to continue after failover.



The following figure depicts the configuration after SRM has been used to reverse SnapMirror replication back to the primary site. The environment has also been reconfigured such that SnapVault backups are occurring from what is now the SnapMirror source. This setup is a SnapMirror SnapVault fan-out configuration.



After SRM performs failover and a second reversal of the SnapMirror relationships, the production data is back at the primary site. This data is now protected in the same way that it was before the failover to the DR site—through SnapMirror and SnapVault backups.

#### Use of Qtrees in Site Recovery Manager environments

Qtrees are special directories that allow the application of file system quotas for NAS. ONTAP 9 allows the creation of qtrees, and qtrees can exist in volumes that are replicated with SnapMirror. However, SnapMirror does not allow replication of individual qtrees or qtree-level replication. All SnapMirror replication is at the volume level only. For this reason, NetApp does not recommend the use of qtrees with SRM.

#### Mixed FC and iSCSI environments

With the supported SAN protocols (FC, FCoE, and iSCSI), ONTAP 9 provides LUN services—that is, the ability to create and map LUNs to attached hosts. Because the cluster consists of multiple controllers, there are multiple logical paths that are managed by multipath I/O to any individual LUN. Asymmetric logical unit access (ALUA) is used on the hosts so that the optimized path to a LUN is selected and is made active for data transfer. If the optimized path to any LUN changes (for example, because the containing volume is moved), ONTAP 9 automatically recognizes and nondisruptively adjusts for this change. If the optimized path becomes unavailable, ONTAP can nondisruptively switch to any other available path.

VMware SRM and NetApp SRA support the use of the FC protocol at one site and the iSCSI protocol at the other site. It does not support having a mix of FC-attached datastores and iSCSI-attached datastores in the same ESXi host or in different hosts in the same cluster, however. This configuration is not supported with SRM because, during the SRM failover or test failover, SRM includes all FC and iSCSI initiators in the ESXi hosts in the request.

## **Best Practice**

SRM and SRA support mixed FC and iSCSI protocols between the protected and recovery sites. However, each site should be configured with only one protocol, either FC or iSCSI, not both protocols at the same site. If a requirement exists to have both FC and iSCSI protocols configured at the same site, NetApp recommends that some hosts use iSCSI and other hosts use FC. NetApp also recommends in this case that SRM resource mappings be set up so that the VMs are configured to fail over into one group of hosts or the other.

## **Troubleshooting SRM when using vVols replication**

The workflow within SRM is significantly different when using vVols replication from what is used with SRA and traditional datastores. For example, there is no array manager concept. As such, `discoverarrays` and `discoverdevices` commands are never seen.

When troubleshooting, it is beneficial to understand the new workflows, which are listed below:

1. `queryReplicationPeer`: Discovers the replication agreements between two fault domains.
2. `queryFaultDomain`: Discovers fault domain hierarchy.
3. `queryReplicationGroup`: Discovers the replication groups present in the source or target domains.
4. `syncReplicationGroup`: Synchronizes the data between source and target.
5. `queryPointInTimeReplica`: Discovers the point in time replicas on a target.
6. `testFailoverReplicationGroupStart`: Begins test failover.
7. `testFailoverReplicationGroupStop`: Ends test failover.
8. `promoteReplicationGroup`: Promotes a group currently in test to production.
9. `prepareFailoverReplicationGroup`: Prepares for a disaster recovery.
10. `failoverReplicationGroup`: Executes disaster recovery.
11. `reverseReplicateGroup`: Initiates reverse replication.
12. `queryMatchingContainer`: Finds containers (along with Hosts or Replication Groups) that might satisfy a provisioning request with a given policy.
13. `queryResourceMetadata`: Discovers the metadata of all resources from the VASA provider, the resource utilization can be returned as an answer to the `queryMatchingContainer` function.

The most common error seen when configuring vVols replication is a failure to discover the SnapMirror relationships. This occurs because the volumes and SnapMirror relationships are created outside of the purview of ONTAP Tools. Therefore, it is a best practice to always make sure your SnapMirror relationship is fully initialized and that you have run a rediscovery in ONTAP Tools at both sites before attempting to create a replicated vVols datastore.

## **Conclusion**

VMware vCenter Site Recovery Manager is a disaster recovery offering that provides automated orchestration and nondisruptive testing of centralized recovery plans to simplify disaster recovery management for all virtualized applications.

By deploying Site Recovery Manager on NetApp ONTAP systems, you can dramatically lower the cost and complexity of disaster recovery. With high-performance, easy-to-manage, and scalable storage appliances and robust software offerings, NetApp offers flexible storage and data management solutions to support vSphere environments.

The best practices and recommendations that are provided in this guide are not a one-size-fits-all solution. This document contains a collection of best practices and recommendations that provide guidelines to plan, deploy, and manage SRM DR plans. Consult with a local NetApp VMware expert when you plan and deploy VMware vCenter Site Recovery environments onto NetApp storage. NetApp VMware experts can quickly identify the needs and demands of any vSphere environment and can adjust the storage solution accordingly.

## Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- TR-4597: VMware vSphere for ONTAP  
[https://docs.netapp.com/us-en/netapp-solutions/hybrid-cloud/vsphere\\_ontap\\_ontap\\_for\\_vsphere.html](https://docs.netapp.com/us-en/netapp-solutions/hybrid-cloud/vsphere_ontap_ontap_for_vsphere.html)
- TR-4400: VMware vSphere Virtual Volumes with ONTAP  
<https://www.netapp.com/pdf.html?item=/media/13555-tr4400.pdf>
- TR-4015 SnapMirror Configuration Best Practice Guide for ONTAP 9  
<https://www.netapp.com/media/17229-tr4015.pdf?v=127202175503P>
- RBAC User Creator for ONTAP  
<https://mysupport.netapp.com/site/tools/tool-eula/rbac>
- ONTAP tools for VMware vSphere Resources  
<https://mysupport.netapp.com/site/products/all/details/otv/docsandkb-tab>
- VMware Site Recovery Manager Documentation  
<https://docs.vmware.com/en/Site-Recovery-Manager/index.html>

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

## Introduction to automation for ONTAP and vSphere

### VMware automation

Automation has been an integral part of managing VMware environments since the first days of VMware ESX. The ability to deploy infrastructure as code and extend practices to private cloud operations helps to alleviate concerns surrounding scale, flexibility, self-provisioning, and efficiency.

Automation can be organized into the following categories:

- **Virtual infrastructure deployment**
- **Guest machine operations**
- **Cloud operations**

There are many options available to administrators with respect to automating their infrastructure. Whether through using native vSphere features such as Host Profiles or Customization Specifications for virtual machines to available APIs on the VMware software components, operating systems, and NetApp storage systems; there is significant documentation and guidance available.

Data ONTAP 8.0.1 and later supports certain VMware vSphere APIs for Array Integration (VAAI) features when the ESX host is running ESX 4.1 or later. VAAI is a set of APIs that enable communication between VMware

vSphere ESXi hosts and storage devices. These features help offload operations from the ESX host to the storage system and increase network throughput. The ESX host enables the features automatically in the correct environment. You can determine the extent to which your system is using VAAI features by checking the statistics contained in the VAAI counters.

The most common starting point for automating the deployment of a VMware environment is provisioning block or file-based datastores. It is important to map out the requirements of the actual tasks prior to developing the corresponding automation.

For more information concerning the automation of VMware environments, see the following resources:

- [The NetApp Pub](#). NetApp configuration management and automation.
- [The Ansible Galaxy Community for VMware](#). A collection of Ansible resources for VMware.
- [VMware {code} Resources](#). Resources needed to design solutions for the software-defined data center, including forums, design standards, sample code, and developer tools.

## vSphere traditional block storage provisioning with ONTAP

VMware vSphere supports the following VMFS datastore options with ONTAP SAN protocol support indicated.

VMFS datastore options	ONTAP SAN protocol support
Fibre Channel (FC)	yes
Fibre Channel over Ethernet (FCoE)	yes
iSCSI	yes
iSCSI Extensions for RDMA (iSER)	no
NVMe over Fabric with FC (NVMe/FC)	yes
NVMe over Fabric with RDMA over Converged Ethernet (NVMe/RoCE)	no



If iSER or NVMe/RoCE VMFS is required, check SANtricity-based storage systems.

## vSphere VMFS datastore - Fibre Channel storage backend with ONTAP

### About this task

This section covers the creation of a VMFS datastore with ONTAP Fibre Channel (FC) storage.

For automated provisioning, use one of these scripts: [\[PowerShell\]](#), [Ansible Playbook](#), or [\[Terraform\]](#).

### What you need

- The basic skills necessary to manage a vSphere environment and ONTAP
- An ONTAP storage system (FAS/AFF/CVO/ONTAP Select/ASA) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, and password)
- ONTAP WWPN of host, target, and SVM and LUN information
- [The completed FC configuration worksheet](#)
- vCenter Server credentials

- vSphere host(s) information
  - vSphere 7.0 or later
- Fabric switch(es)
  - With connected ONTAP FC data ports and vSphere hosts
  - With the N\_port ID virtualization (NPIV) feature enabled
  - Create a single initiator single target zone.
    - Create one zone for each initiator (single initiator zone).
    - For each zone, include a target that is the ONTAP FC logical interface (WWPN) for the SVMs. There should be at least two logical interfaces per node per SVM. Do not use the WWPN of the physical ports.
- An ONTAP Tool for VMware vSphere deployed, configured, and ready to consume.

## Provisioning a VMFS datastore

To provision a VMFS datastore, complete the following steps:

1. Check compatibility with the [Interoperability Matrix Tool \(IMT\)](#)
2. Verify that the [FCP Configuration is supported](#).

## ONTAP tasks

1. [Verify that you have an ONTAP license for FCP.](#)
  - a. Use the `system license show` command to check that FCP is listed.
  - b. Use `license add -license-code <license code>` to add the license.
2. Make sure that the FCP protocol is enabled on the SVM.
  - a. [Verify the FCP on an existing SVM.](#)
  - b. [Configure the FCP on an existing SVM.](#)
  - c. [Create a new SVM with the FCP.](#)
3. Make sure that FCP logical interfaces are available on an SVM.
  - a. Use `Network Interface show` to verify the FCP adapter.
  - b. When an SVM is created with the GUI, logical interfaces are a part of that process.
  - c. To rename network interfaces, use `Network Interface modify`.
4. [Create and Map a LUN.](#) Skip this step if you are using ONTAP tools for VMware vSphere.

## VMware vSphere tasks

1. Verify that HBA drivers are installed. VMware supported HBAs have drivers deployed out of the box and should be visible in the [Storage Adapter Information](#).
2. [Provision a VMFS datastore with ONTAP Tools.](#)

**vSphere VMFS Datastore - Fibre Channel over Ethernet storage protocol with ONTAP**

## About this task

This section covers the creation of a VMFS datastore with the Fibre Channel over Ethernet (FCoE) transport protocol to ONTAP storage.

For automated provisioning, use one of these scripts: [\[PowerShell\]](#), [Ansible Playbook](#), or [\[Terraform\]](#).

## What you need

- The basic skills necessary to manage a vSphere environment and ONTAP
- An ONTAP storage system (FAS/AFF/CVO/ONTAP Select) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, and password)
- [A supported FCoE combination](#)
- [A completed configuration worksheet](#)
- vCenter Server credentials
- vSphere host(s) information
  - vSphere 7.0 or later
- Fabric switch(es)
  - With either ONTAP FC data ports or vSphere hosts connected
  - With the N\_port ID virtualization (NPIV) feature enabled
  - Create a single initiator single target zone.
  - [FC/FCoE zoning configured](#)
- Network switch(es)
  - FCoE support
  - DCB support
  - [Jumbo frames for FCoE](#)
- ONTAP Tool for VMware vSphere deployed, configured, and ready to consume

## Provision a VMFS datastore

- Check compatibility with the [Interoperability Matrix Tool \(IMT\)](#).
- [Verify that the FCoE configuration is supported](#).

## ONTAP tasks

1. [Verify the ONTAP license for FCP.](#)
  - a. Use the `system license show` command to verify that the FCP is listed.
  - b. Use `license add -license-code <license code>` to add a license.
2. Verify that the FCP protocol is enabled on the SVM.
  - a. [Verify the FCP on an existing SVM.](#)
  - b. [Configure the FCP on an existing SVM.](#)
  - c. [Create a new SVM with the FCP.](#)

3. Verify that FCP logical interfaces are available on the SVM.
  - a. Use `Network Interface show` to verify the FCP adapter.
  - b. When the SVM is created with the GUI, logical interfaces are a part of that process.
  - c. To rename the network interface, use `Network Interface modify`.
4. [Create and map a LUN](#); skip this step if you are using ONTAP tools for VMware vSphere.

## VMware vSphere tasks

1. Verify that HBA drivers are installed. VMware-supported HBAs have drivers deployed out of the box and should be visible in the [storage adapter information](#).
2. [Provision a VMFS datastore with ONTAP Tools](#).

### vSphere VMFS Datastore - iSCSI Storage backend with ONTAP

#### About this task

This section covers the creation of a VMFS datastore with ONTAP iSCSI storage.

For automated provisioning, use one of these scripts: [\[PowerShell\]](#), [Ansible Playbook](#), or [\[Terraform\]](#).

#### What you need

- The basic skills necessary to manage a vSphere environment and ONTAP.
- An ONTAP storage system (FAS/AFF/CVO/ONTAP Select/ASA) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, and password)
- ONTAP network port, SVM, and LUN information for iSCSI
- [A completed iSCSI configuration worksheet](#)
- vCenter Server credentials
- vSphere host(s) information
  - vSphere 7.0 or later
- iSCSI VMKernel adapter IP information
- Network switch(es)
  - With ONTAP system network data ports and connected vSphere hosts
  - VLAN(s) configured for iSCSI
  - (Optional) link aggregation configured for ONTAP network data ports
- ONTAP Tool for VMware vSphere deployed, configured, and ready to consume

#### Steps

1. Check compatibility with the [Interoperability Matrix Tool \(IMT\)](#).
2. [Verify that the iSCSI configuration is supported](#).
3. Complete the following ONTAP and vSphere tasks.

## ONTAP tasks

1. [Verify the ONTAP license for iSCSI.](#)
    - a. Use the `system license show` command to check if iSCSI is listed.
    - b. Use `license add -license-code <license code>` to add the license.
  2. [Verify that the iSCSI protocol is enabled on the SVM.](#)
  3. Verify that iSCSI network logical interfaces are available on the SVM.
-  When an SVM is created using the GUI, iSCSI network interfaces are also created.
4. Use the `Network interface` command to view or make changes to the network interface.
-  Two iSCSI network interfaces per node are recommended.
5. [Create an iSCSI network interface.](#) You can use the default-data-blocks service policy.
  6. [Verify that the data-iscsi service is included in the service policy.](#) You can use `network interface service-policy show` to verify.
  7. [Verify that jumbo frames are enabled.](#)
  8. [Create and map the LUN.](#) Skip this step if you are using ONTAP tools for VMware vSphere. Repeat this step for each LUN.

## VMware vSphere tasks

1. Verify that at least one NIC is available for the iSCSI VLAN. Two NICs are preferred for better performance and fault tolerance.
2. [Identify the number of physical NICs available on the vSphere host.](#)
3. [Configure the iSCSI initiator.](#) A typical use case is a software iSCSI initiator.
4. [Verify that the TCPIP stack for iSCSI is available.](#)
5. [Verify that iSCSI portgroups are available.](#)
  - We typically use a single virtual switch with multiple uplink ports.
  - Use 1:1 adapter mapping.
6. Verify that iSCSI VMKernel adapters are enabled to match the number of NICs and that IPs are assigned.
7. [Bind the iSCSI software adapter to the iSCSI VMKernel adapter\(s\).](#)
8. [Provision the VMFS datastore with ONTAP Tools.](#) Repeat this step for all datastores.
9. [Verify hardware acceleration support.](#)

## What's next?

After these the tasks are completed, the VMFS datastore is ready to consume for provisioning virtual machines.

## Ansible Playbook

```
## Disclaimer: Sample script for reference purpose only.
```

```

- hosts: '{{ vsphere_host }}'
  name: Play for vSphere iSCSI Configuration
  connection: local
  gather_facts: false
  tasks:
    # Generate Session ID for vCenter
    - name: Generate a Session ID for vCenter
      uri:
        url: "https://{{ vcenter_hostname }}/rest/com/vmware/cis/session"
        validate_certs: false
        method: POST
        user: "{{ vcenter_username }}"
        password: "{{ vcenter_password }}"
        force_basic_auth: yes
        return_content: yes
      register: vclogin

    # Generate Session ID for ONTAP tools with vCenter
    - name: Generate a Session ID for ONTAP tools with vCenter
      uri:
        url: "https://{{ ontap_tools_ip }}:8143/api/rest/2.0/security/user/login"
        validate_certs: false
        method: POST
        return_content: yes
        body_format: json
        body:
          vcenterUserName: "{{ vcenter_username }}"
          vcenterPassword: "{{ vcenter_password }}"
      register: login

    # Get existing registered ONTAP Cluster info with ONTAP tools
    - name: Get ONTAP Cluster info from ONTAP tools
      uri:
        url: "https://{{ ontap_tools_ip }}:8143/api/rest/2.0/storage/clusters"
        validate_certs: false
        method: Get
        return_content: yes
        headers:
          vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
      register: clusterinfo

    - name: Get ONTAP Cluster ID
      set_fact:
        ontap_cluster_id: "{{ clusterinfo.json | "

```

```

    json_query(clusteridquery) }}"
  vars:
    clusteridquery: "records[?ipAddress == '{{ netapp_hostname }}' &&
type=='Cluster'].id | [0]"

  - name: Get ONTAP SVM ID
    set_fact:
      ontap_svm_id: "{{ clusterinfo.json | json_query(svmidquery) }}"
    vars:
      svmidquery: "records[?ipAddress == '{{ netapp_hostname }}' &&
type=='SVM' && name == '{{ svm_name }}'].id | [0]"

  - name: Get Aggregate detail
    uri:
      url: "https://{{ ontap_tools_ip
}}:8143/api/rest/2.0/storage/clusters/{{ ontap_svm_id }}/aggregates"
      validate_certs: false
      method: GET
      return_content: yes
      headers:
        vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
        cluster-id: "{{ ontap_svm_id }}"
    when: ontap_svm_id != ''
    register: aggrinfo

  - name: Select Aggregate with max free capacity
    set_fact:
      aggr_name: "{{ aggrinfo.json | json_query(aggrquery) }}"
    vars:
      aggrquery: "max_by(records, &freeCapacity).name"

  - name: Convert datastore size in MB
    set_fact:
      datastoreSizeInMB: "{{ iscsi_datastore_size |
human_to_bytes/1024/1024 | int }}"

  - name: Get vSphere Cluster Info
    uri:
      url: "https://{{ vcenter_hostname }}/api/vcenter/cluster?names={{ vsphere_cluster }}"
      validate_certs: false
      method: GET
      return_content: yes
      body_format: json
      headers:
        vmware-api-session-id: "{{ vclogin.json.value }}"

```

```

when: vsphere_cluster != ''
register: vcenterclusterid

- name: Create iSCSI VMFS-6 Datastore with ONTAP tools
  uri:
    url: "https://{{ ontap_tools_ip }}:8143/api/rest/3.0/admin/datastore"
    validate_certs: false
  method: POST
  return_content: yes
  status_code: [200]
  body_format: json
  body:
    traditionalDatastoreRequest:
      name: "{{ iscsi_datastore_name }}"
      datastoreType: VMFS
      protocol: ISCSI
      spaceReserve: Thin
      clusterID: "{{ ontap_cluster_id }}"
      svmID: "{{ ontap_svm_id }}"
      targetMoref: ClusterComputeResource:{{ vcenterclusterid.json[0].cluster }}
      datastoreSizeInMB: "{{ datastoreSizeInMB | int }}"
      vmfsFileSystem: VMFS6
      aggrName: "{{ aggr_name }}"
      existingFlexVolName: ""
      volumeStyle: FLEXVOL
      datastoreClusterMoref: ""
  headers:
    vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
when: ontap_cluster_id != '' and ontap_svm_id != '' and aggr_name !=
  ''
  register: result
  changed_when: result.status == 200

```

## vSphere VMFS Datastore - NVMe/FC with ONTAP

### About this task

This section covers the creation of a VMFS datastore with ONTAP storage using NVMe/FC.

For automated provisioning, use one of these scripts: [\[PowerShell\]](#), [Ansible Playbook](#), or [\[Terraform\]](#).

### What you need

- Basic skills needed to manage a vSphere environment and ONTAP.
- [Basic understanding of NVMe/FC](#).

- An ONTAP Storage System (FAS/AFF/CVO/ONTAP Select/ASA) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, and password)
- ONTAP WWPN for host, target, and SVMs and LUN information
- [A completed FC configuration worksheet](#)
- vCenter Server
- vSphere host(s) information (vSphere 7.0 or later)
- Fabric switch(es)
  - With ONTAP FC data ports and vSphere hosts connected.
  - With the N\_port ID virtualization (NPIV) feature enabled.
  - Create a single initiator target zone.
  - Create one zone for each initiator (single initiator zone).
  - For each zone, include a target that is the ONTAP FC logical interface (WWPN) for the SVMs. There should be at least two logical interfaces per node per SVM. DO not use the WWPN of physical ports.

## Provision VMFS datastore

1. Check compatibility with the [Interoperability Matrix Tool \(IMT\)](#).
2. [Verify that the NVMe/FC configuration is supported](#).

## ONTAP tasks

1. [Verify the ONTAP license for FCP](#).  
Use the `system license show` command and check if NVMe\_oF is listed.  
Use `license add -license-code <license code>` to add a license.
2. Verify that NVMe protocol is enabled on the SVM.
  - a. [Configure SVMs for NVMe](#).
3. Verify that NVMe/FC Logical Interfaces are available on the SVMs.
  - a. Use `Network Interface show` to verify the FCP adapter.
  - b. When an SVM is created with the GUI, logical interfaces are as part of that process.
  - c. To rename the network interface, use the command `Network Interface modify`.
4. [Create NVMe namespace and subsystem](#)

## VMware vSphere Tasks

1. Verify that HBA drivers are installed. VMware supported HBAs have the drivers deployed out of the box and should be visible at [Storage Adapter Information](#)
2. [Perform vSphere Host NVMe driver installatioln and validation tasks](#)
3. [Create VMFS Datastore](#)

## vSphere traditional file storage provisioning with ONTAP

VMware vSphere supports following NFS protocols, both of which support ONTAP.

- [NFS Version 3](#)

- [NFS Version 4.1](#)

If you need help selecting the correct NFS version for vSphere, check [this comparison of NFS client versions](#).

## Reference

Unresolved directive in hybrid-cloud/vsphere\_ontap\_auto\_file.adoc - include::hybrid-cloud/vsphere\_ontap\_best\_practices.adoc[tag=nfs]

### vSphere NFS datastore - Version 3 with ONTAP

#### About this task

Creation of NFS version 3 datastore with ONTAP NAS storage.

For automated provisioning, use one of these scripts: [\[PowerShell\]](#), [Ansible Playbook](#), or [\[Terraform\]](#).

#### What you need

- The basic skill necessary to manage a vSphere environment and ONTAP.
- An ONTAP storage system (FAS/AFF/CVO/ONTAP Select/Cloud Volume Service/Azure NetApp Files) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, password)
- ONTAP network port, SVM, and LUN information for NFS
  - [A completed NFS configuration worksheet](#)
- vCenter Server credentials
- vSphere host(s) information for vSphere 7.0 or later
- NFS VMKernel adapter IP information
- Network switch(es)
  - with ONTAP system network data ports and connected vSphere hosts
  - VLAN(s) configured for NFS
  - (Optional) link aggregation configured for ONTAP network data ports
- ONTAP Tool for VMware vSphere deployed, configured, and ready to consume

#### Steps

- Check compatibility with the [Interoperability Matrix Tool \(IMT\)](#)
  - [Verify that the NFS configuration is supported.](#)
- Complete the following ONTAP and vSphere tasks.

#### ONTAP tasks

1. [Verify the ONTAP license for NFS.](#)
  - a. Use the `system license show` command and check that NFS is listed.
  - b. Use `license add -license-code <license code>` to add a license.
2. [Follow the NFS configuration workflow.](#)

## VMware vSphere Tasks

Follow the workflow for NFS client configuration for vSphere.

### Reference

Unresolved directive in hybrid-cloud/vsphere\_ontap\_auto\_file\_nfs.adoc - include::hybrid-cloud/vsphere\_ontap\_best\_practices.adoc[lines=315..390]

### What's next?

After these tasks are completed, the NFS datastore is ready to consume for provisioning virtual machines.

#### vSphere NFS Datastore - Version 4.1 with ONTAP

#### About this task

This section describes the creation of an NFS version 4.1 datastore with ONTAP NAS storage.

For automated provisioning, use one of these scripts: [\[PowerShell\]](#), [Ansible Playbook](#), or [\[Terraform\]](#).

#### What you need

- The basic skills necessary to manage a vSphere environment and ONTAP
- ONTAP Storage System (FAS/AFF/CVO/ONTAP Select/Cloud Volume Service/Azure NetApp Files) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, password)
- ONTAP network port, SVM, and LUN information for NFS
- [A completed NFS configuration worksheet](#)
- vCenter Server credentials
- vSphere host(s) information vSphere 7.0 or later
- NFS VMKernel adapter IP information
- Network switch(es)
  - with ONTAP system network data ports, vSphere hosts, and connected
  - VLAN(s) configured for NFS
  - (Optional) link aggregation configured for ONTAP network data ports
- ONTAP Tools for VMware vSphere deployed, configured, and ready to consume

#### Steps

- Check compatibility with the [Interoperability Matrix Tool \(IMT\)](#).
  - [Verify that the NFS configuration is supported](#).
- Complete the ONTAP and vSphere Tasks provided below.

#### ONTAP tasks

1. [Verify ONTAP license for NFS](#)

- a. Use the `system license show` command to check whether NFS is listed.
  - b. Use `license add -license-code <license code>` to add a license.
2. [Follow the NFS configuration workflow](#)

## VMware vSphere tasks

[Follow the NFS Client Configuration for vSphere workflow.](#)

### What's next?

After these tasks are completed, the NFS datastore is ready to consume for provisioning virtual machines.

## What's New with ONTAP for VMware Virtualization

:allow-uri-read

### VMware Virtualization

VMware integration and support in ONTAP 9.8 gets a boost with a number of new features including FlexGroup datastore support. ONTAP 9.8 allows you to provision a FlexGroup volume as a VMware NFS datastore, simplifying datastore management with a single, scalable datastore that provides the power of a full ONTAP cluster. Many of these new features are coming with the ONTAP tools for VMware vSphere 9.8 release.

This means the following applies:

- Validated performance and placement
- Interop qualification
- Enhanced VAAI copy offload that is faster and completes in the background
- Virtual Storage Console support, including FlexGroup provisioning, resize and deletion, setting QoS on individual VMs, and displaying performance metrics (latency, IOPS, and throughput) for VMs
- NetApp SnapCenter primary storage backup and recovery support
- Support for a maximum of 64TB VMFS LUNs. With support for 128TB LUNs/300TB FlexVol volumes with the NetApp All-SAN Array, you can provision the maximum 64TB VMFS datastore using the Virtual Storage Console in the ONTAP tools for VMware vSphere 9.8 release.
- Increased [Site Recovery Manager \(SRM\)](#) scale. The Storage Replication Adapter in the ONTAP tools for VMware vSphere 9.8 release increases the scale of datastores and protection groups supported up to 512.
- VMware vSphere vVols file metrics with REST APIs. REST API support for vVols file metrics is added to ONTAP 9.8, which allows the Virtual Storage Console to display ONTAP storage performance metrics for vVols in the dashboard and reports.
- [Storage Replication Adapter \(SRA\)](#) support for SnapMirror Synchronous
- Support for [VMware Tanzu](#) storage
- Improved support for vVols, including an enhanced SAN vVol rebalancing command and enhancements to Storage Capability Profiles. For more information on the latest VMware virtualization support, see the following resources:
  - [Tech ONTAP Podcast Episode 263: Virtualization in ONTAP – Fall 2020](#)
  - [TR-4597: VMware vSphere with ONTAP](#)

## **VMware Private Cloud**

## **Red Hat Private Cloud**

## **Workload Performance**

## **Demos and Tutorials**

### **Hybrid cloud, desktop virtualization and containers videos and demos**

See the following videos and demos highlighting specific features of the hybrid cloud, desktop virtualization, and container solutions.

#### **NetApp with VMware Tanzu**

VMware Tanzu enables customers to deploy, administer, and manage their Kubernetes environment through vSphere or the VMware Cloud Foundation. This portfolio of products from VMware allows customer to manage all their relevant Kubernetes clusters from a single control plane by choosing the VMware Tanzu edition that best suits their needs.

For more information about VMware Tanzu, see the [VMware Tanzu Overview](#). This review covers use cases, available additions, and more about VMware Tanzu.

#### **NetApp with VMware Tanzu video series**

- [How to use vVols with NetApp and VMware Tanzu Basic, part 1](#)
- [How to use vVols with NetApp and VMware Tanzu Basic, part 2](#)
- [How to use vVols with NetApp and VMware Tanzu Basic, part 3](#)

#### **NetApp with Red Hat OpenShift**

Red Hat OpenShift, an enterprise Kubernetes platform, enables you to run container-based applications with an open hybrid-cloud strategy. Available as a cloud service on leading public clouds or as self-managed software, Red Hat OpenShift provides customers with the flexibility they need when designing their container-based solution.

For more information regarding Red Hat OpenShift, see this [Red Hat OpenShift Overview](#). You can also review the product documentation and deployment options to learn more about Red Hat OpenShift.

#### **NetApp with Red Hat OpenShift videos**

- [Workload Migration - Red Hat OpenShift with NetApp](#)
- [Red Hat OpenShift Deployment on RHV: Red Hat OpenShift with NetApp](#)

# Virtual Desktops

## Virtual Desktop Services (VDS)

### TR-4861: Hybrid Cloud VDI with Virtual Desktop Service

Suresh Thoppay, NetApp

The NetApp Virtual Desktop Service (VDS) orchestrates Remote Desktop Services (RDS) in major public clouds as well as on private clouds. VDS supports Windows Virtual Desktop (WVD) on Microsoft Azure. VDS automates many tasks that must be performed after deployment of WVD or RDS, including setting up SMB file shares (for user profiles, shared data, and the user home drive), enabling Windows features, application and agent installation, firewall, and policies, and so on.

Users consume VDS for dedicated desktops, shared desktops, and remote applications. VDS provides scripted events for automating application management for desktops and reduces the number of images to manage.

VDS provides a single management portal for handling deployments across public and private cloud environments.

### Customer Value

The remote workforce explosion of 2020 has changed requirements for business continuity. IT departments are faced with new challenges to rapidly provision virtual desktops and thus require provisioning agility, remote management, and the TCO advantages of a hybrid cloud that makes it easy to provision on-premises and cloud resources. They need a hybrid-cloud solution that:

- Addresses the post-COVID workspace reality to enable flexible work models with global dynamics
- Enables shift work by simplifying and accelerating the deployment of work environments for all employees, from task workers to power users
- Mobilizes your workforce by providing rich, secure VDI resources regardless of the physical location
- Simplifies hybrid-cloud deployment
- Automates and simplifies risk reduction management

[Next: Use Cases](#)

### Use Cases

Hybrid VDI with NetApp VDS allows service providers and enterprise virtual desktop administrators to easily expand resources to other cloud environment without affecting their users. Having on-premises resources provides better control of resources and offers wide selection of choices (compute, GPU, storage, and network) to meet demand.

This solution applies to the following use cases:

- Bursting into the cloud for surges in demand for remote desktops and applications
- Reducing TCO for long running remote desktops and applications by hosting them on-premises with flash storage and GPU resources
- Ease of management of remote desktops and applications across cloud environments

- Experience remote desktops and applications by using a software-as-a-service model with on-premises resources

## Target Audience

The target audience for the solution includes the following groups:

- EUC/VDI architects who want to understand the requirements for a hybrid VDS
- NetApp partners who would like to assist customers with their remote desktop and application needs
- Existing NetApp HCI customers who want to address remote desktop and application demands

[Next: NetApp Virtual Desktop Service Overview](#)

## NetApp Virtual Desktop Service Overview

NetApp offers many cloud services, including the rapid provisioning of virtual desktop with WVD or remote applications and rapid integration with Azure NetApp Files.

Traditionally, it takes weeks to provision and deliver remote desktop services to customers. Apart from provisioning, it can be difficult to manage applications, user profiles, shared data, and group policy objects to enforce policies. Firewall rules can increase complexity and require a separate skillset and tools.

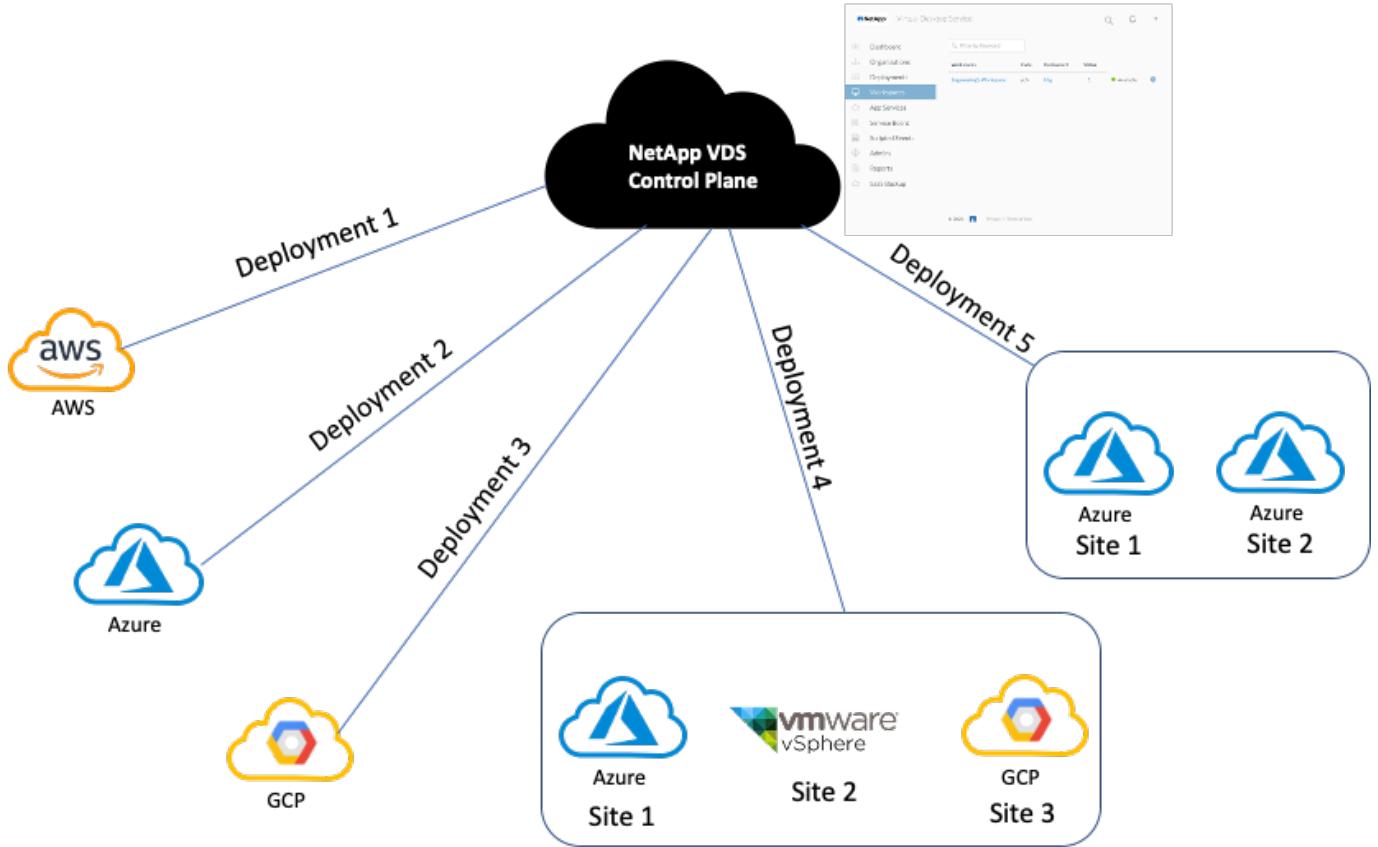
With Microsoft Azure Windows Virtual Desktop service, Microsoft takes care of maintenance for Remote Desktop Services components, allowing customers to focus on provisioning workspaces in the cloud. Customers must provision and manage the complete stack which requires special skills to manage VDI environments.

With NetApp VDS, customers can rapidly deploy virtual desktops without worrying about where to install the architecture components like brokers, gateways, agents, and so on. Customers who require complete control of their environment can work with a professional services team to achieve their goals. Customers consume VDS as a service and thus can focus on their key business challenges.

NetApp VDS is a software-as-a-service offering for centrally managing multiple deployments across AWS, Azure, GCP, or private cloud environments. Microsoft Windows Virtual Desktop is available only on Microsoft Azure. NetApp VDS orchestrates Microsoft Remote Desktop Services in other environments.

Microsoft offers multisession on Windows 10 exclusively for Windows Virtual Desktop environments on Azure. Authentication and identity are handled by the virtual desktop technology; WVD requires Azure Active Directory synced (with AD Connect) to Active Directory and session VMs joined to Active Directory. RDS requires Active Directory for user identity and authentication and VM domain join and management.

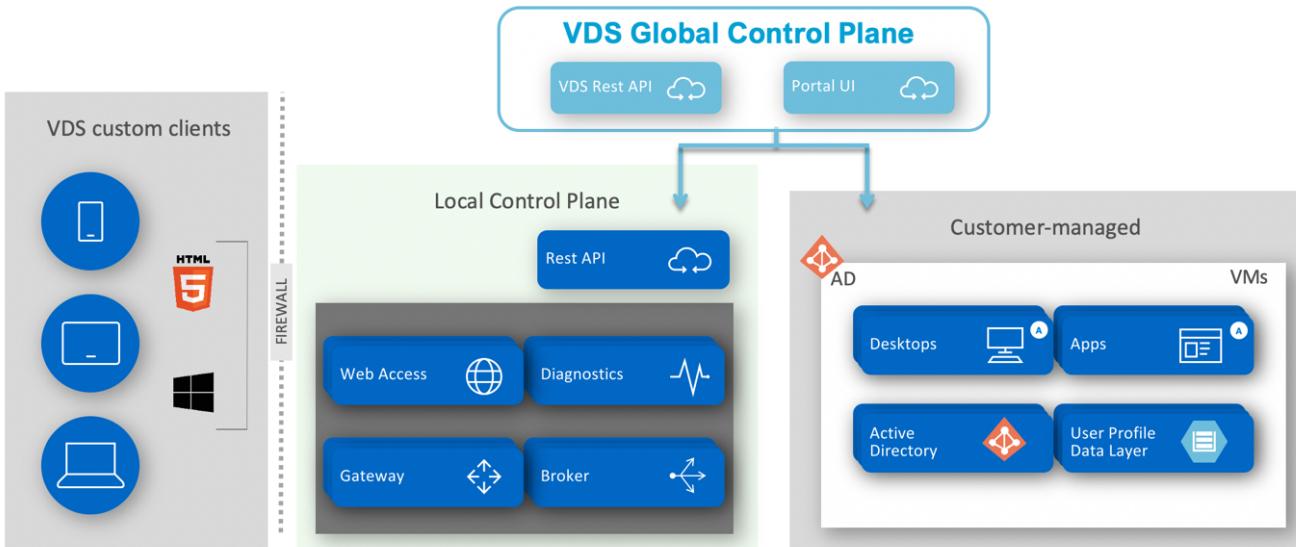
A sample deployment topology is shown in the following figure.



Each deployment is associated with an active directory domain and provides clients with an access entry point for workspaces and applications. A service provider or enterprise that has multiple active directory domains typically has more deployments. A single Active Directory domain that spans multiple regions typically has a single deployment with multiple sites.

For WVD in Azure, Microsoft provides a platform-as-a-service that is consumed by NetApp VDS. For other environments, NetApp VDS orchestrates the deployment and configuration of Microsoft Remote Desktop Services. NetApp VDS supports both WVD Classic and WVD ARM and can also be used to upgrade existing versions.

Each deployment has its own platform services, which consists of Cloud Workspace Manager (REST API endpoint), an HTML 5 Gateway (connect to VMs from a VDS management portal), RDS Gateways (Access point for clients), and a Domain Controller. The following figure depicts the VDS Control Plane architecture for RDS implementation.



For RDS implementations, NetApp VDS can be readily accessed from Windows and browsers using client software that can be customized to include customer logo and images. Based on user credentials, it provides user access to approved workspaces and applications. There is no need to configure the gateway details.

The following figure shows the NetApp VDS client.



## Virtual Desktop Service

Username

Password

Forgot Password

Save Username

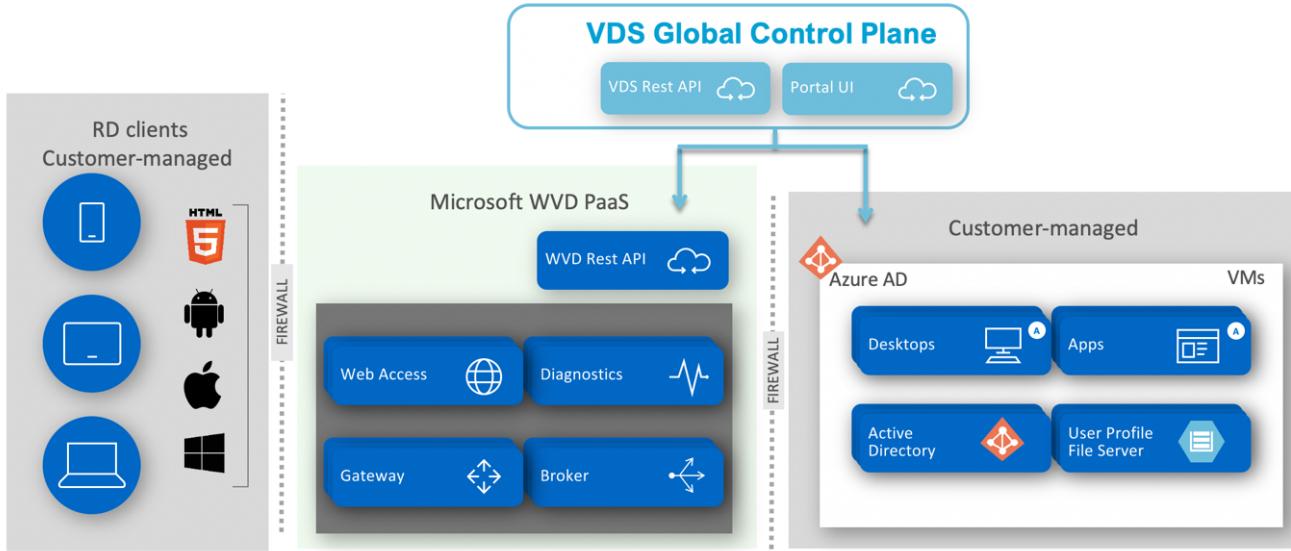


Workspace

Applications

In the Azure WVD implementation, Microsoft handles the access entry point for the clients and can be consumed by a Microsoft WVD client available natively for various OSs. It can also be accessed from a web-based portal. The configuration of client software must be handled by the Group Policy Object (GPO) or in other ways preferred by customers.

The following figure depicts the VDS Control Plane architecture for Azure WVD implementations.



In addition to the deployment and configuration of required components, NetApp VDS also handles user management, application management, resource scaling, and optimization.

NetApp VDS can create users or grant existing user accounts access to cloud workspace or application services. The portal can also be used for password resets and the delegation of administrating a subset of components. Helpdesk administrators or Level-3 technicians can shadow user sessions for troubleshooting or connect to servers from within the portal.

NetApp VDS can use image templates that you create, or it can use existing ones from the marketplace for cloud-based provisioning. To reduce the number of images to manage, you can use a base image, and any additional applications that you require can be provisioned using the provided framework to include any command-line tools like Chocolatey, MSIX app attach, PowerShell, and so on. Even custom scripts can be used as part of machine lifecycle events.

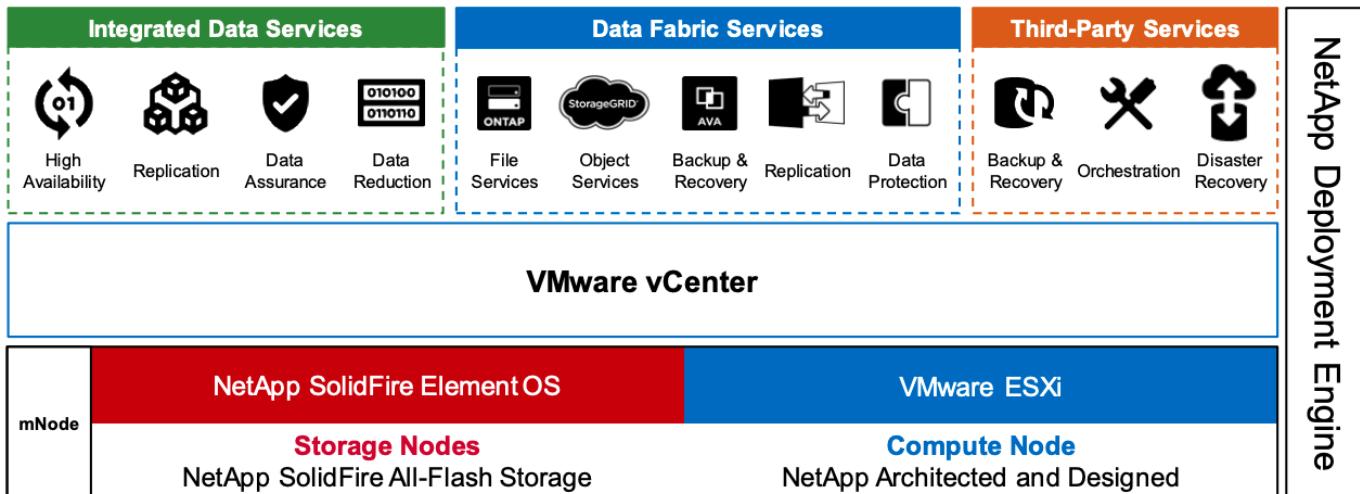
[Next: NetApp HCI Overview](#)

## NetApp HCI Overview

NetApp HCI is a hybrid cloud infrastructure that consists of a mix of storage nodes and compute nodes. It is available as either a two-rack unit or single-rack unit, depending on the model. The installation and configuration required to deploy VMs are automated with the NetApp Deployment Engine (NDE). Compute clusters are managed with VMware vCenter, and storage clusters are managed with the vCenter Plug-in deployed with NDE. A management VM called the mNode is deployed as part of the NDE.

NetApp HCI handles the following functions:

- Version upgrades
- Pushing events to vCenter
- vCenter Plug-In management
- A VPN tunnel for support
- The NetApp Active IQ collector
- The extension of NetApp Cloud Services to on the premises, enabling a hybrid cloud infrastructure. The following figure depicts HCI components.



## Storage Nodes

Storage nodes are available as either a half-width or full-width rack unit. A minimum of four storage nodes is required at first, and a cluster can expand to up to 40 nodes. A storage cluster can be shared across multiple compute clusters. All the storage nodes contain a cache controller to improve write performance. A single node provides either 50K or 100K IOPS at a 4K block size.

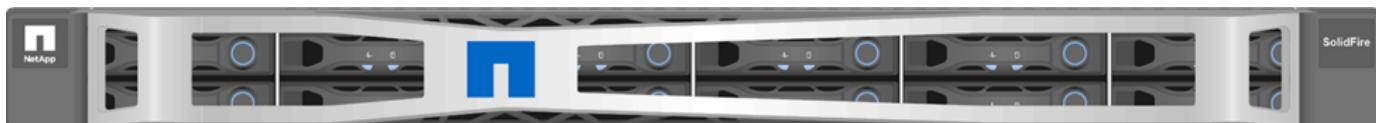
NetApp HCI storage nodes run NetApp Element software, which provides minimum, maximum, and burst QoS limits. The storage cluster supports a mix of storage nodes, although one storage node cannot exceed one-third of total capacity.

## Compute Nodes



NetApp supports its storage connected to any compute servers listed in the [VMware Compatibility Guide](#).

Compute nodes are available in half-width, full-width, and two rack-unit sizes. The NetApp HCI H410C and H610C are based on scalable Intel Skylake processors. The H615C is based on second-generation scalable Intel Cascade Lake processors. There are two compute models that contain GPUs: the H610C contains two NVIDIA M10 cards and the H615C contains three NVIDIA T4 cards.

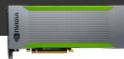


The NVIDIA T4 has 40 RT cores that provide the computation power needed to deliver real-time ray tracing. The same server model used by designers and engineers can now also be used by artists to create photorealistic imagery that features light bouncing off surfaces just as it would in real life. This RTX-capable GPU produces real-time ray tracing performance of up to five Giga Rays per second. The NVIDIA T4, when combined with Quadro Virtual Data Center Workstation (Quadro vDWS) software, enables artists to create photorealistic designs with accurate shadows, reflections, and refractions on any device from any location.

Tensor cores enable you to run deep learning inferencing workloads. When running these workloads, an NVIDIA T4 powered with Quadro vDWS can perform up to 25 times faster than a VM driven by a CPU-only server. A NetApp H615C with three NVIDIA T4 cards in one rack unit is an ideal solution for graphics and compute-intensive workloads.

The following figure lists NVIDIA GPU cards and compares their features.

## NVIDIA GPUs Recommended for Virtualization

	V100S	RTX 8000	RTX 6000	T4	M10	P6
						
<b>GPU</b>	1 NVIDIA Volta	1 NVIDIA Turing	1 NVIDIA Turing	1 NVIDIA Turing	4 NVIDIA Maxwell	1 NVIDIA Pascal
<b>CUDA Cores</b>	5,120	4,608	4,608	2,560	2,560 (640 per GPU)	2,048
<b>Tensor Cores</b>	640	576	576	320	—	—
<b>RT Cores</b>	—	72	72	40	—	—
<b>Guaranteed QoS [GPU Scheduler]</b>	✓	✓	✓	✓	—	✓
<b>Live Migration</b>	✓	✓	✓	✓	✓	✓
<b>Multi-vGPU</b>	✓	✓	✓	✓	✓	✓
<b>Memory Size</b>	32/16 GB HBM2	48 GB GDDR6	24 GB GDDR6	16 GB GDDR6	32 GB GDDR5 (8 GB per GPU)	16 GB GDDR5
<b>vGPU Profiles</b>	1 GB, 2 GB, 4 GB, 8 GB, 16 GB, 32 GB	1 GB, 2 GB, 3 GB, 4 GB, 6 GB, 8 GB, 12 GB, 16 GB, 24 GB, 48 GB	1 GB, 2 GB, 3 GB, 4 GB, 6 GB, 8 GB, 12 GB, 24 GB	1 GB, 2 GB, 4 GB, 8 GB, 16 GB	0.5 GB, 1 GB, 2 GB, 4 GB, 8 GB	1 GB, 2 GB, 4 GB, 8 GB, 16 GB
<b>Form Factor</b>	PCIe 3.0 dual slot and SXM2	PCIe 3.0 dual slot	PCIe 3.0 dual slot	PCIe 3.0 single slot	PCIe 3.0 dual slot	MXM (blade servers)
<b>Power</b>	250 W /300 W (SXM2)	250 W	250 W	70 W	225 W	90 W
<b>Thermal</b>	passive	passive	passive	passive	passive	bare board
<b>vGPU Software Support</b>	Quadro vDWS, GRID vPC, GRID vApps, vComputeServer	Quadro vDWS, GRID vPC, GRID vApps, vComputeServer	Quadro vDWS, GRID vPC, GRID vApps, vComputeServer	Quadro vDWS, GRID vPC, GRID vApps, vComputeServer	Quadro vDWS, GRID vPC, GRID vApps	Quadro vDWS, GRID vPC, GRID vApps, vComputeServer
<b>Use Case</b>	Ultra-high-end rendering, simulation, 3D design with Quadro vDWS; ideal upgrade path for V100	High-end rendering, 3D design and creative workflows with Quadro vDWS	Mid-range to high-end rendering, 3D design and creative workflows with Quadro vDWS	Entry-level to high-end 3D design and engineering workflows with Quadro vDWS. High-density, low power GPU acceleration for knowledge workers with NVIDIA GRID software.	Knowledge workers using modern productivity apps and Windows 10 requiring best density and total cost of ownership (TCO), multimonitor support with NVIDIA GRID vPC/vApps	For customers requiring GPUs in a blade server form factor; ideal upgrade path for M6

The M10 GPU remains the best TCO solution for knowledge-worker use cases. However, the T4 makes a great alternative when IT wants to standardize on a GPU that can be used across multiple use cases, such as virtual workstations, graphics performance, real-time interactive rendering, and inferencing. With the T4, IT can take advantage of the same GPU resources to run mixed workloads—for example, running VDI during the day and repurposing the resources to run compute workloads at night.

The H610C compute node is two rack units in size; the H615C is one rack unit in size and consumes less power. The H615C supports H.264 and H.265 (High Efficiency Video Coding [HEVC]) 4:4:4 encoding and decoding. It also supports the increasingly mainstream VP9 decoder; even the WebM container package served by YouTube uses the VP9 codec for video.

The number of nodes in a compute cluster is dictated by VMware; currently, it is 96 with VMware vSphere 7.0 Update 1. Mixing different models of compute nodes in a cluster is supported when Enhanced vMotion Compatibility (EVC) is enabled.

[Next: NVIDIA Licensing](#)

## NVIDIA Licensing

When using an H610C or H615C, the license for the GPU must be procured from NVIDIA partners that are authorized to resell the licenses. You can find NVIDIA partners with the [partner locator](#). Search for competencies such as virtual GPU (vGPU) or Tesla.

NVIDIA vGPU software is available in four editions:

- NVIDIA GRID Virtual PC (GRID vPC)
- NVIDIA GRID Virtual Applications (GRID vApps)
- NVIDIA Quadro Virtual Data Center Workstation (Quadro vDWS)
- NVIDIA Virtual ComputeServer (vComputeServer)

## **GRID Virtual PC**

This product is ideal for users who want a virtual desktop that provides a great user experience for Microsoft Windows applications, browsers, high-definition video, and multi-monitor support. The NVIDIA GRID Virtual PC delivers a native experience in a virtual environment, allowing you to run all your PC applications at full performance.

## **GRID Virtual Applications**

GRID vApps are for organizations deploying a Remote Desktop Session Host (RDSH) or other app-streaming or session-based solutions. Designed to deliver Microsoft Windows applications at full performance, Windows Server-hosted RDSH desktops are also supported by GRID vApps.

## **Quadro Virtual Data Center Workstation**

This edition is ideal for mainstream and high-end designers who use powerful 3D content creation applications like Dassault CATIA, SOLIDWORKS, 3Dexcite, Siemens NX, PTC Creo, Schlumberger Petrel, or Autodesk Maya. NVIDIA Quadro vDWS allows users to access their professional graphics applications with full features and performance anywhere on any device.

## **NVIDIA Virtual ComputeServer**

Many organizations run compute-intensive server workloads such as artificial intelligence (AI), deep learning (DL), and data science. For these use cases, NVIDIA vComputeServer software virtualizes the NVIDIA GPU, which accelerates compute-intensive server workloads with features such as error correction code, page retirement, peer-to-peer over NVLink, and multi-vGPU.



A Quadro vDWS license enables you to use GRID vPC and NVIDIA vComputeServer.

## [Next: Deployment](#)

### **Deployment**

NetApp VDS can be deployed to Microsoft Azure using a setup app available based on the required codebase. The current release is available [here](#) and the preview release of the upcoming product is available [here](#).

See [this video](#) for deployment instructions.



# NetApp Virtual Desktop Service

## Deployment & AD Connect

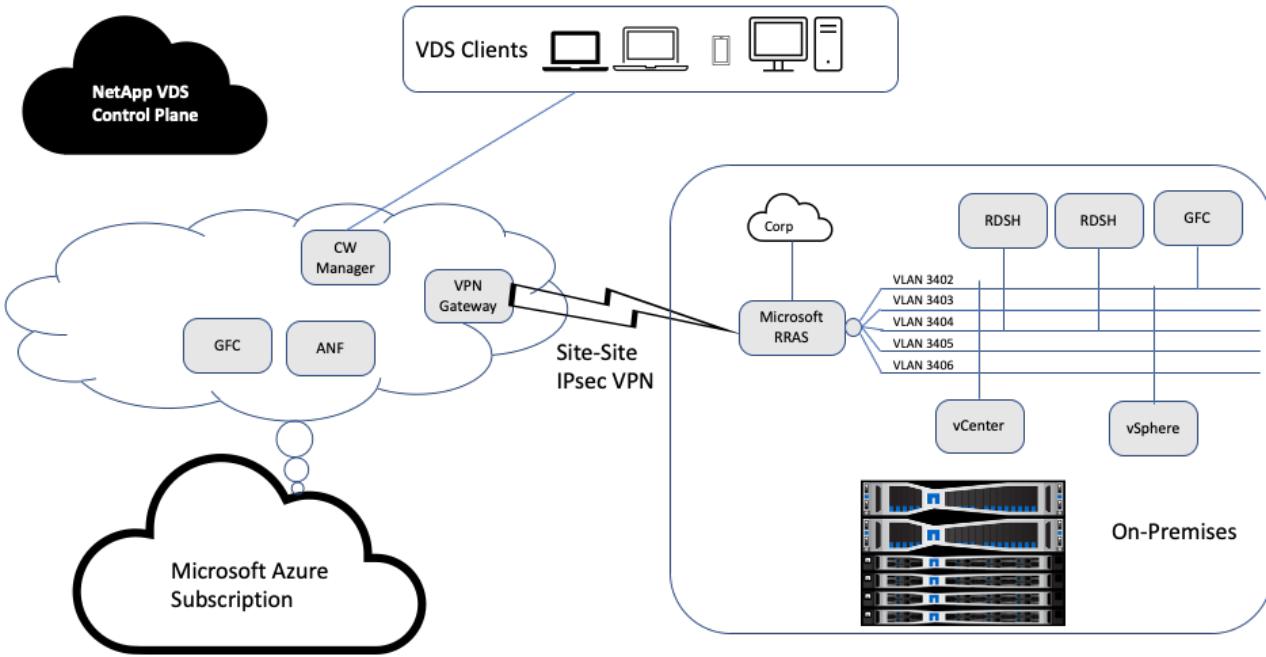
Toby vanRoojen  
Product Marketing Manager  
June, 2020

[Next: Hybrid Cloud Environment](#)

### Hybrid Cloud Environment

NetApp Virtual Desktop Service can be extended to on-premises when connectivity exists between on-premises resources and cloud resources. Enterprises can establish the link to Microsoft Azure using Express Route or a site-to-site IPsec VPN connection. You can also create links to other clouds in a similar way either using a dedicated link or with an IPsec VPN tunnel.

For the solution validation, we used the environment depicted in the following figure.



On-premises, we had multiple VLANs for management, remote-desktop-session hosts, and so on. They were on the 172.21.146-150.0/24 subnet and routed to the corporate network using the Microsoft Remote Routing Access Service. We also performed the following tasks:

1. We noted the public IP of the Microsoft Routing and Remote Access Server (RRAS; identified with IPchicken.com).
2. We created a Virtual Network Gateway resource (route-based VPN) on Azure Subscription.
3. We created the connection providing the local network gateway address for the public IP of the Microsoft RRAS server.
4. We completed VPN configuration on RRAS to create a virtual interface using pre-shared authentication that was provided while creating the VPN gateway. If configured correctly, the VPN should be in the connected state. Instead of Microsoft RRAS, you can also use pfSense or other relevant tools to create the site-to-site IPsec VPN tunnel. Since it is route-based, the tunnel redirects traffic based on the specific subnets configured.

Microsoft Azure Active Directory provides identity authentication based on oAuth. Enterprise client authentications typically require NTLM or Kerberos-based authentication. Microsoft Azure Active Directory Domain Services perform password hash sync between Azure Active Directory and on-prem domain controllers using ADConnect.

For this Hybrid VDS solution validation, we initially deployed to Microsoft Azure and added an additional site with vSphere. The advantage with this approach is that platform services were deployed to Microsoft Azure and were then readily backed up using the portal. Services can then be easily accessed from anywhere, even if the site-site VPN link is down.

To add another site, we used a tool called DCConfig. The shortcut to that application is available on the desktop of the cloud workspace manager (CWMgr) VM. After this application is launched, navigate to the DataCenter Sites tab, add the new datacenter site, and fill in the required info as shown below. The URL points to the vCenter IP. Make sure that the CWMgr VM can communicate with vCenter before adding the

configuration.



Make sure that vSphere PowerCLI 5.1 on CloudWorkspace manager is installed to enable communication with VMware vSphere environment.

The following figure depicts on-premises datacenter site configuration.

The screenshot shows the 'DataCenter' tab selected in the navigation bar. A table lists two sites: 'Site 1' (AzureRM) and 'Site 2' (vSphere). 'Site 2' is marked as primary. The 'Edit' button for Site 2 is highlighted. The right panel is titled 'DataCenter Site' and contains fields for 'DataCenter Site' (Site 2), 'Hypervisor' (vSphere), and buttons for 'Cancel Edit', 'Save', 'Load Hypervisor', and 'Test'. It also includes sections for 'General Settings' (Local VM Account, URL https://172.21.146.150/sdk, Vm Name Prefix, Max Concurrent Create Server, Subnet Mask, Default Gateway), 'DNS' (Primary DNS: 10.67.78.11, Secondary DNS, Set DNS Address: Yes), and 'VSphere' (Data Center: NetApp-HCI-Datacenter, Cluster, Resource Pool, Host Name, VM Folder: VDS, Max VMs In Datastore, Min HD Free Space In Datastore GB, Min Ram Free GB). At the bottom are 'Exclude VSphere DataStore' and 'Exclude VSphere ResourcePools' buttons.

Note that there are filtering options available for compute resource based on the specific cluster, host name, or free RAM space. Filtering options for storage resource includes the minimum free space on datastores or the maximum VMs per datastore. Datastores can be excluded using regular expressions. Click Save button to save the configuration.

To validate the configuration, click the Test button or click Load Hypervisor and check any dropdown under the vSphere section. It should be populated with appropriate values. It is a best practice to keep the primary hypervisor set to yes for the default provisioning site.

The VM templates created on VMware vSphere are consumed as provisioning collections on VDS. Provisioning collections come in two forms: shared and VDI. The shared provisioning collection type is used for remote desktop services for which a single resource policy is applied to all servers. The VDI type is used for WVD instances for which the resource policy is individually assigned. The servers in a provisioning collection can be assigned one of the following three roles:

- **TSDATA.** Combination of Terminal Services and Data server role.
- **TS.** Terminal Services (Session Host).
- **DATA.** File Server or Database Server. When you define the server role, you must pick the VM template and storage (datastore). The datastore chosen can be restricted to a specific datastore or you can use the least-used option in which the datastore is chosen based on data usage.

Each deployment has VM resource defaults for the cloud resource allocation based on Active Users, Fixed, Server Load, or User Count.

[Next: Single Server Load Test with Login VSI](#)

### Single server load test with Login VSI

The NetApp Virtual Desktop Service uses the Microsoft Remote Desktop Protocol to access virtual desktop sessions and applications, and the Login VSI tool determines the maximum number of users that can be hosted on a specific server model. Login VSI simulates user login at specific intervals and performs user operations like opening documents, reading and composing mails, working with Excel and PowerPoint, printing documents, compressing files, and taking random breaks. It then measures response times. User response time is low when server utilization is low and increases when more user sessions are added. Login VSI determines the baseline based on initial user login sessions and it reports the maximum user session when the user response exceeds 2 seconds from the baseline.

NetApp Virtual Desktop Service utilizes Microsoft Remote Desktop Protocol to access the Virtual Desktop session and Applications. To determine the maximum number of users that can be hosted on a specific server model, we used the Login VSI tool. Login VSI simulates user login at specific intervals and performs user operations like opening documents, reading and composing mails, working with Excel and PowerPoint, printing documents, compressing files, taking random breaks, and so on. It also measures response times. User response time is low when server utilization is low and increases when more user sessions are added. Login VSI determines the baseline based on the initial user login sessions and it reports maximum user sessions when the user response exceeds 2sec from the baseline.

The following table contains the hardware used for this validation.

Model	Count	Description
NetApp HCI H610C	4	Three in a cluster for launchers, AD, DHCP, and so on. One server for load testing.
NetApp HCI H615C	1	2x24C Intel Xeon Gold 6282 @2.1GHz. 1.5TB RAM.

The following table contains the software used for this validation.

product	Description
NetApp VDS 5.4	Orchestration
VM Template Windows 2019 1809	Server OS for RDSH
Login VSI	4.1.32.1
VMware vSphere 6.7 Update 3	Hypervisor
VMware vCenter 6.7 Update 3f	VMware management tool

The Login VSI test results are as follows:

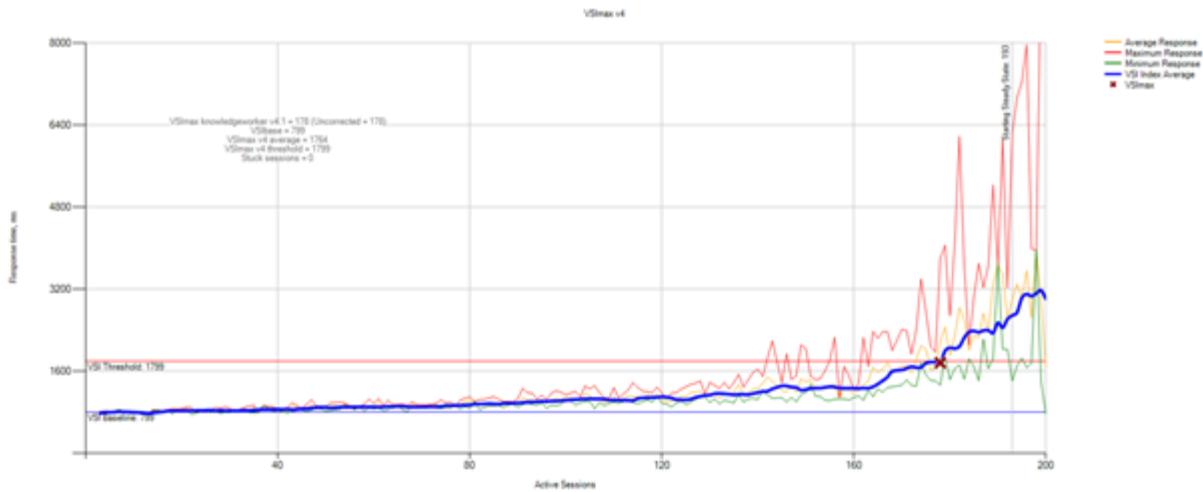
Model	VM configuration	Login VSI baseline	Login VSI Max
H610C	8 vCPU, 48GB RAM, 75GB disk, 8Q vGPU profile	799	178
H615C	12 vCPU, 128GB RAM, 75GB disk	763	272

Considering sub-NUMA boundaries and hyperthreading, the eight VMs chosen for VM testing and configuration depended on the cores available on the host.

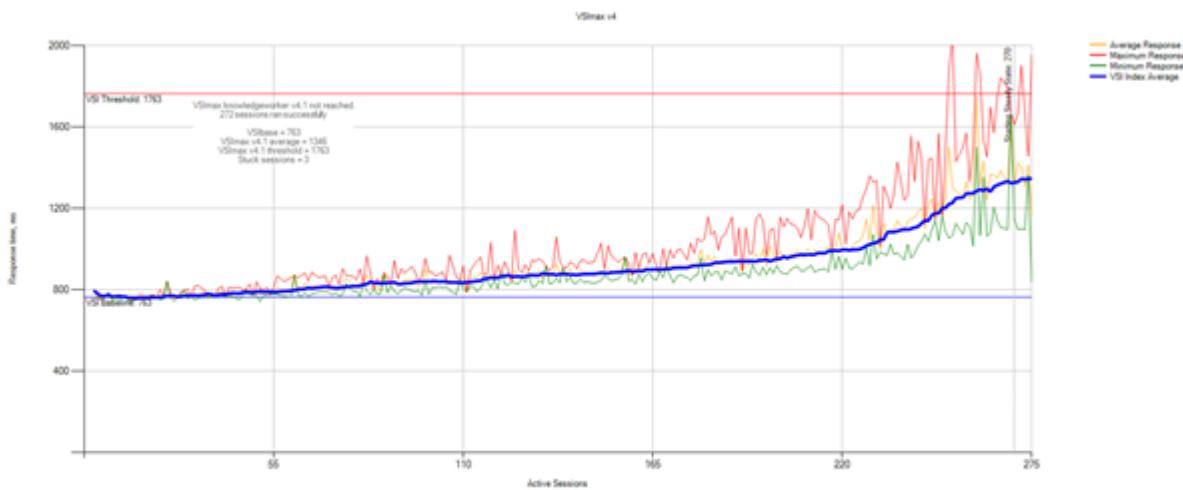
We used 10 launcher VMs on the H610C, which used the RDP protocol to connect to the user session. The following figure depicts the Login VSI connection information.

The screenshot shows the 'Connection' page of the Login VSI Management Console. The left sidebar has a 'Test Setup' section selected, containing 'Scenario', 'Connection' (which is highlighted), and 'Start Test'. The main area shows a 'CURRENT CONNECTION BASED ON Microsoft RDP Connection' section with a 'start connection wizard' button. Below it is a 'CONNECTION CONFIGURATION' section with a 'Command line' field containing a PowerShell command to run 'RDPConnect.exe'. There's also a 'CSV file' section with a 'use CSV file' checkbox and a browse button. The 'CONNECTION DETAILS' section includes fields for 'Server' (m6x.vly.cloudworkspace.app), 'Username' (LVS1-VDS\count/4), 'Password' (redacted), and 'Domain' (DemoVDS.com). At the bottom is a yellow navigation bar with icons for Analyzer, Settings, Help, Enter Benchmark Mode, Save Profile, Load Profile, and Exit.

The following figure displays the Login VSI response time versus the active sessions for the H610C.



The following figure displays the Login VSI response time versus active sessions for the H615C.



The performance metrics from Cloud Insights during H615C Login VSI testing for the vSphere host and VMs are shown in the following figure.



[Next: Management Portal](#)

## Management Portal

NetApp VDS Cloud Workspace Management Suite portal is available [here](#) and the upcoming version is available [here](#).

The portal allows centralized management for various VDS deployments including one that has sites defined for on-premises, administrative users, the application catalog, and scripted events. The portal is also used by administrative users for the manual provisioning of applications if required and to connect to any machines for troubleshooting.

Service providers can use this portal to add their own channel partners and allow them to manage their own clients.

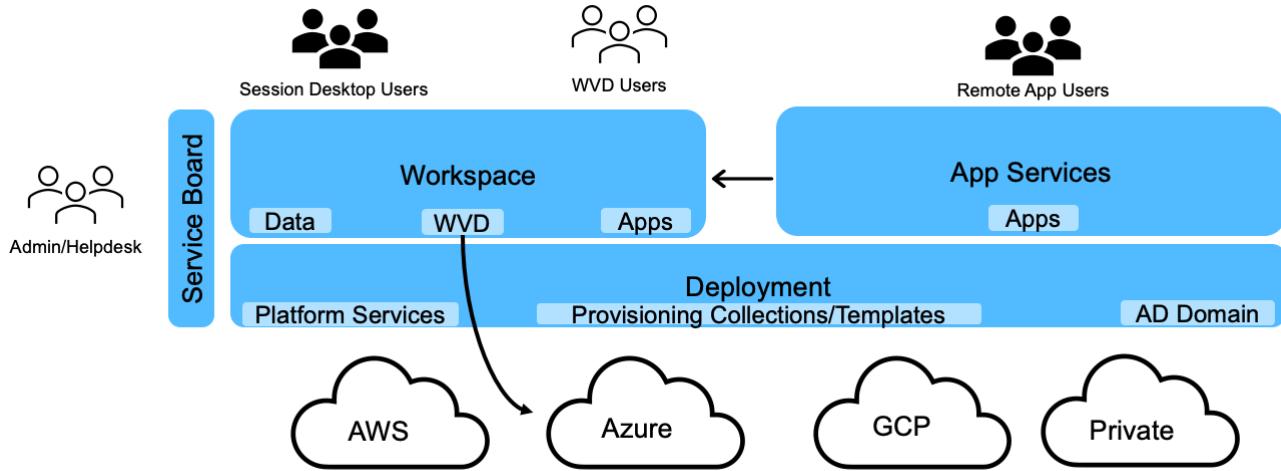
[Next: User Management](#)

## User Management

NetApp VDS uses Azure Active Directory for identity authentication and Azure Active Directory Domain Services for NTLM/Kerberos authentication. The ADConnect tool can be used to sync an on-prem Active Directory domain with Azure Active Directory.

New users can be added from the portal, or you can enable cloud workspace for existing users. Permissions for workspaces and application services can be controlled by individual users or by groups. From the management portal, administrative users can be defined to control permissions for the portal, workspaces, and so on.

The following figure depicts user management in NetApp VDS.



Each workspace resides in its own Active Directory organization unit (OU) under the Cloud Workspace OU as shown in the following figure.

Active Directory Users and Computers

Name	Type	Description
87499	Security Group...	Microsoft Access
87500	Security Group...	Microsoft Excel
87501	Security Group...	Google Chrome
87502	Security Group...	Microsoft PowerPoint
87503	Security Group...	Microsoft Word
87517	Security Group...	PuTTy
ych-all users	Security Group...	Company All Users

File Action View Help

Active Directory Users and Computers [cwmgr1.vds]

- > Saved Queries
- < vds.demo
- < < Builtin
- < < Cloud Workspace
  - < < Cloud Workspace Companies
    - < < hpyh
      - < < hpyh-groups
    - < < ych
      - < < ych-desktop users
      - < < ych-groups
  - < < Cloud Workspace Servers
  - < < Cloud Workspace Service Accounts
    - < Client Service Accounts
    - < Infrastructure Service Accounts
  - < < Cloud Workspace Tech Users
    - < Groups
    - < Level3 Technicians
  - < < Computers
  - < < Domain Controllers
  - < < ForeignSecurityPrincipals
  - < < Managed Service Accounts
  - < < Users

For more info, see [this video](#) on user permissions and user management in NetApp VDS.

When an Active Directory group is defined as a CRAUserGroup using an API call for the datacenter, all the users in that group are imported into the CloudWorkspace for management using the UI. As the cloud workspace is enabled for the user, VDS creates user home folders, settings permissions, user properties updates, and so on.

If VDI User Enabled is checked, VDS creates a single-session RDS machine dedicated to that user. It prompts for the template and the datastore to provision.

The screenshot shows the 'Security Settings' configuration page. At the top left is a 'Security Settings' button. Below it are two checkboxes: 'VDI User Enabled' (checked) and 'Mobile Drive Enabled' (unchecked). Under 'Hypervisor Template', there is a dropdown menu set to 'Windows20192899ver1'. A 'Storage Type' dropdown menu is set to 'DS02'. On the right side, there are four checkboxes: 'Account Expiration Enabled' (unchecked), 'Local Drive Access Enabled' (checked), 'Force Password Reset at Next Login' (unchecked), 'Wake On Demand Enabled' (unchecked), and 'Multi-factor Auth Enabled' (unchecked). At the bottom left is a grey 'Update' button.

[Next: Workspace Management](#)

## Workspace Management

A workspace consists of a desktop environment; this can be shared remote desktop sessions hosted on-premises or on any supported cloud environment. With Microsoft Azure, the desktop environment can be persistent with Windows Virtual Desktops. Each workspace is associated with a specific organization or client. Options available when creating a new workspace can be seen in the following figure.

## New Workspace

Client & Settings
Choose Applications
Add Users
Review & Provision

---

Select a Client [Add](#)

No Clients Added.

**Workspace Settings**

Company Name

Primary Notification Email

**Application Settings**

- Enable Remote App
- Enable App Locker
- Enable Application Usage Tracking

**Device Settings**

- Disable Printing Access
- Enable Workspace User Data Storage

**Security Settings**

- Require Complex User Password
- Enable MFA for All Users
- Permit Access To Task Manager

Cancel
Continue



Each workspace is associated with specific deployment.

Workspaces contain associated apps and app services, shared data folders, servers, and a WVD instance. Each workspace can control security options like enforcing password complexity, multifactor authentication, file audits, and so on.

Workspaces can control the workload schedule to power on extra servers, limit the number of users per server, or set the schedule for the resources available for given period (always on/off). Resources can also be configured to wake up on demand.

The workspace can override the deployment VM resource defaults if required. For WVD, WVD host pools (which contains session hosts and app groups) and WVD workspaces can also be managed from the cloud workspace management suite portal. For more info on the WVD host pool, see this [video](#).

[Next: Application Management](#)

### Application Management

Task workers can quickly launch an application from the list of applications made available to them. App services publish applications from the Remote Desktop Services session hosts. With WVD, App Groups provide similar functionality from multi-session Windows 10 host pools.

For office workers to power users, the applications that they require can be provisioned manually using a

service board, or they can be auto-provisioned using the scripted events feature in NetApp VDS.

For more information, see the [NetApp Application Entitlement page](#).

Next: [ONTAP features for Virtual Desktop Service](#)

## ONTAP features for Virtual Desktop Service

The following ONTAP features make it attractive choice for use with a virtual desktop service.

- **Scale-out filesystem.** ONTAP FlexGroup volumes can grow to more than 20PB in size and can contain more than 400 billion files within a single namespace. The cluster can contain up to 24 storage nodes, each with a flexible the number of network interface cards depending on the model used.

User's virtual desktops, home folders, user profile containers, shared data, and so on can grow based on demand with no concern for filesystem limitations.

- **File system analytics.** You can use the XCP tool to gain insights into shared data. With ONTAP 9.8+ and ActiveIQ Unified Manager, you can easily query and retrieve file metadata information and identify cold data.
- **Cloud tiering.** You can migrate cold data to an object store in the cloud or to any S3-compatible storage in your datacenter.
- **File versions.** Users can recover files protected by NetApp ONTAP Snapshot copies. ONTAP Snapshot copies are very space efficient because they only record changed blocks.
- **Global namespace.** ONTAP FlexCache technology allows remote caching of file storage making it easier to manage shared data across locations containing ONTAP storage systems.
- **Secure multi-tenancy support.** A single physical storage cluster can be presented as multiple virtual storage arrays each with its own volumes, storage protocols, logical network interfaces, identity and authentication domain, management users, and so on. Therefore, you can share the storage array across multiple business units or environments, such as test, development, and production.

To guarantee performance, you can use adaptive QoS to set performance levels based on used or allocated space, and you can control storage capacity by using quotas.

- **VMware integration.** ONTAP tools for VMware vSphere provides a vCenter plug-in to provision datastores, implement vSphere host best practices, and monitor ONTAP resources.

ONTAP supports vStorage APIs for Array Integration (VAAI) for offloading SCSI/file operations to the storage array. ONTAP also supports vStorage APIs for Storage Awareness (VASA) and Virtual Volumes support for both block and file protocols.

The Snapcenter Plug-in for VMware vSphere provides an easy way to back up and restore virtual machines using the Snapshot feature on a storage array.

ActiveIQ Unified Manager provides end-to-end storage network visibility in a vSphere environment. Administrators can easily identify any latency issues that might occur on virtual desktop environments hosted on ONTAP.

- **Security compliance.** With ActiveIQ Unified Manager, you can monitor multiple ONTAP systems with alerts for any policy violations.
- **Multi-protocol support.** ONTAP supports block (iSCSI, FC, FCoE, and NVMe/FC), file (NFSv3, NFSv4.1, SMB2.x, and SMB3.x), and object (S3) storage protocols.

- **Automation support.** ONTAP provides REST API, Ansible, and PowerShell modules to automate tasks with the VDS Management Portal.

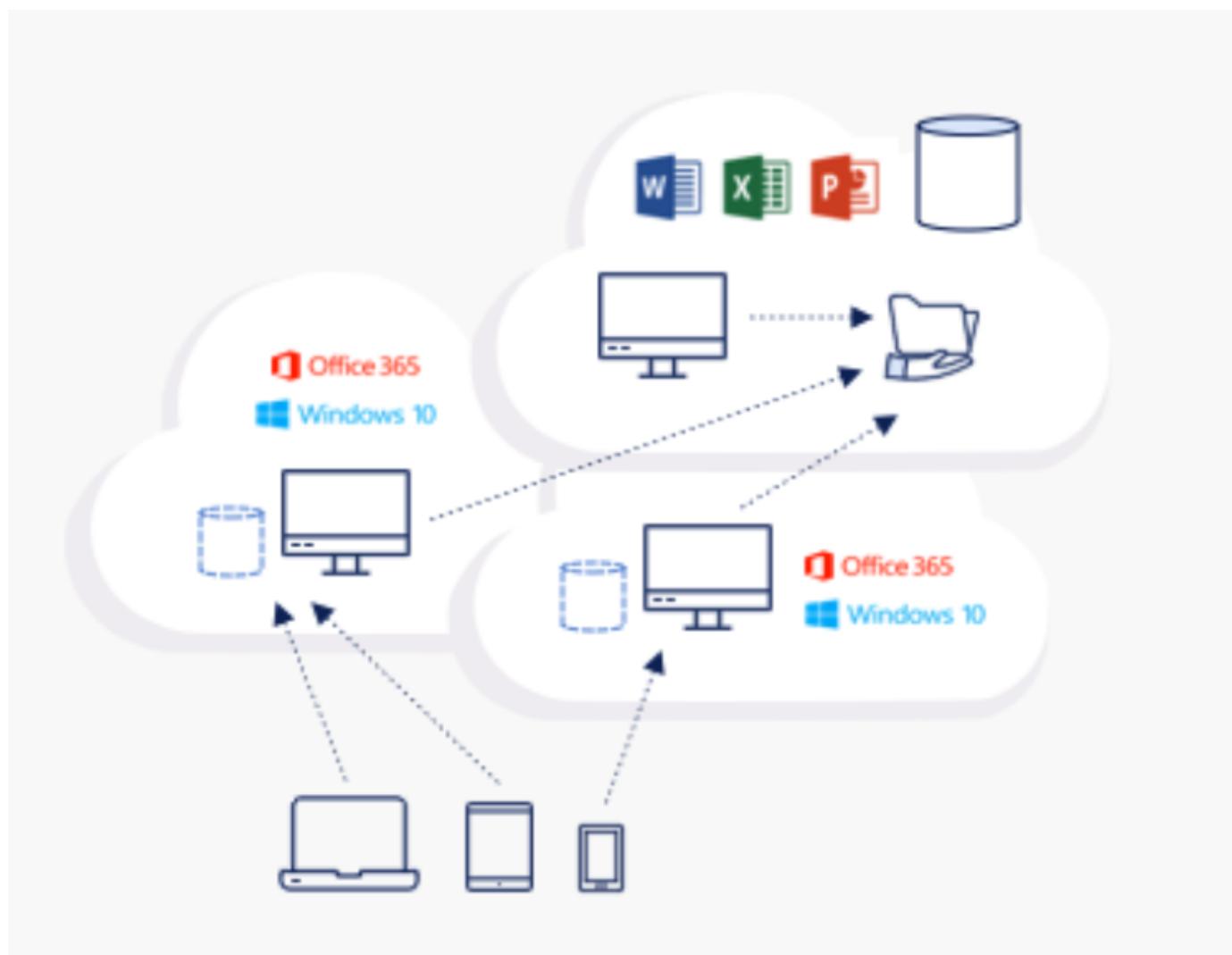
Next: Data Management

## Data Management

As a part of deployment, you can choose the file-services method to host the user profile, shared data, and the home drive folder. The available options are File Server, Azure Files, or Azure NetApp Files. However, after deployment, you can modify this choice with the Command Center tool to point to any SMB share. [There are various advantages to hosting with NetApp ONTAP](#). To learn how to change the SMB share, see [Change Data Layer](#).

### Global File Cache

When users are spread across multiple sites within a global namespace, Global File Cache can help reduce latency for frequently accessed data. Global File Cache deployment can be automated using a provisioning collection and scripted events. Global File Cache handles the read and write caches locally and maintains file locks across locations. Global File Cache can work with any SMB file servers, including Azure NetApp Files.



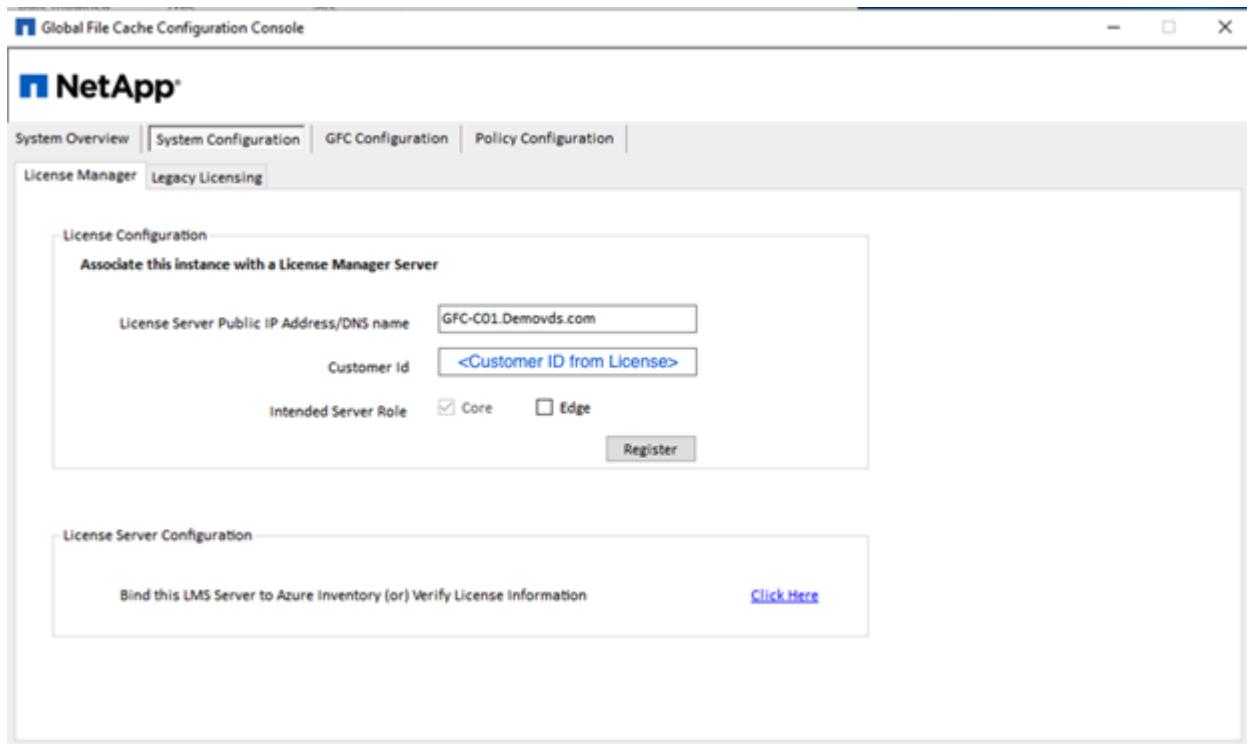
Global File Cache requires the following:

- Management server (License Management Server)
- Core
- Edge with enough disk capacity to cache the data

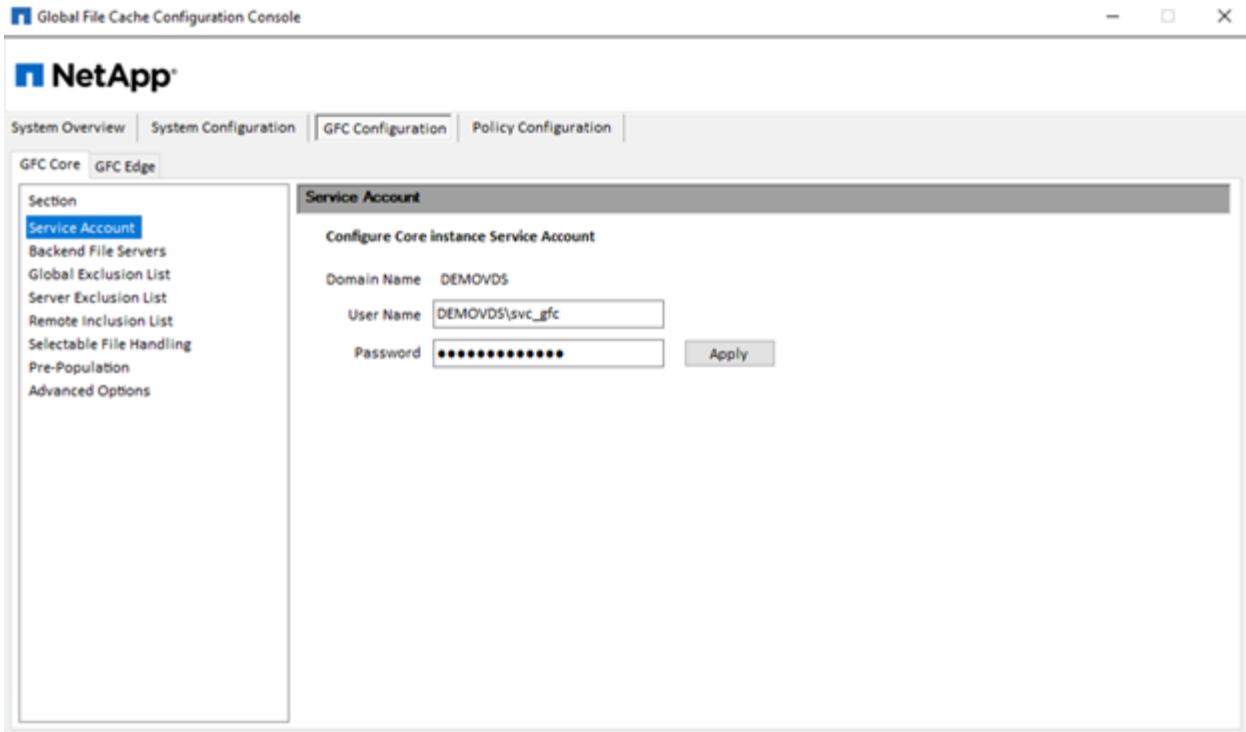
To download the software and to calculate the disk cache capacity for Edge, see the [GFC documentation](#).

For our validation, we deployed the core and management resources on the same VM at Azure and edge resources on NetApp HCI. Please note that the core is where high-volume data access is required and the edge is a subset of the core. After the software is installed, you must activate the license activated before use. To do so, complete the following steps:

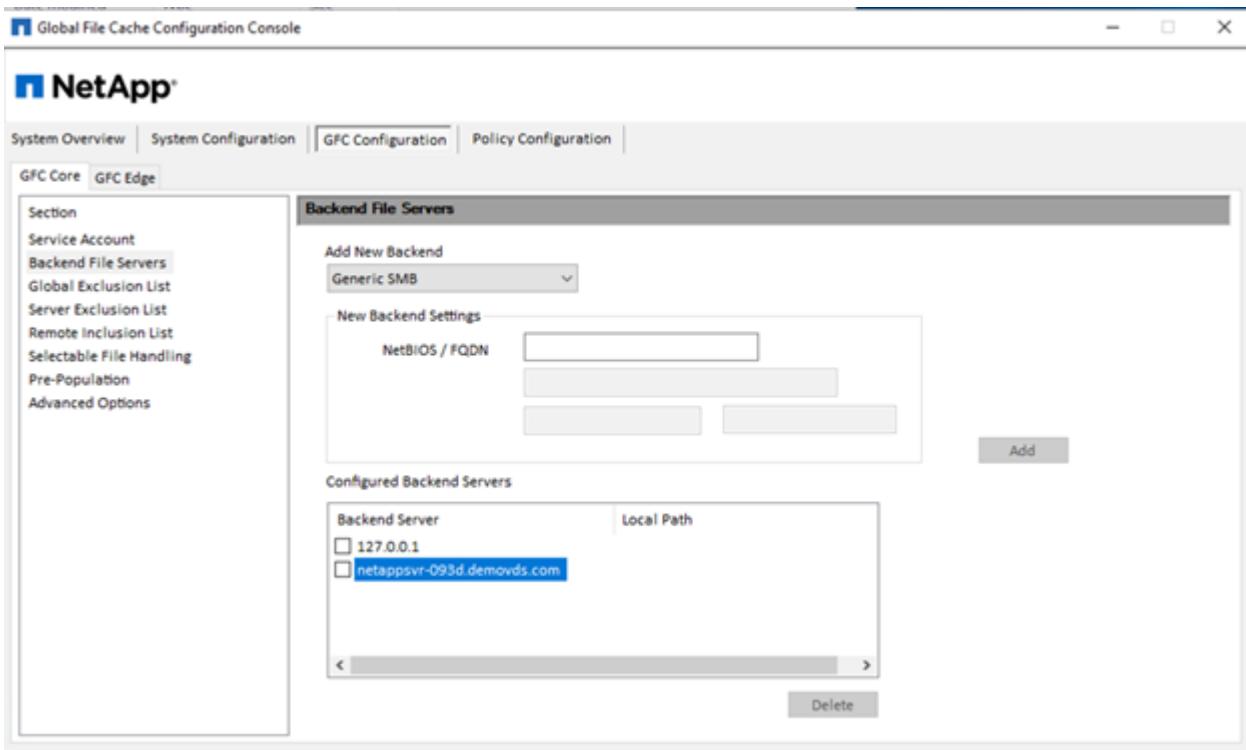
1. Under the License Configuration section, use the link [Click Here](#) to complete the license activation. Then register the core.



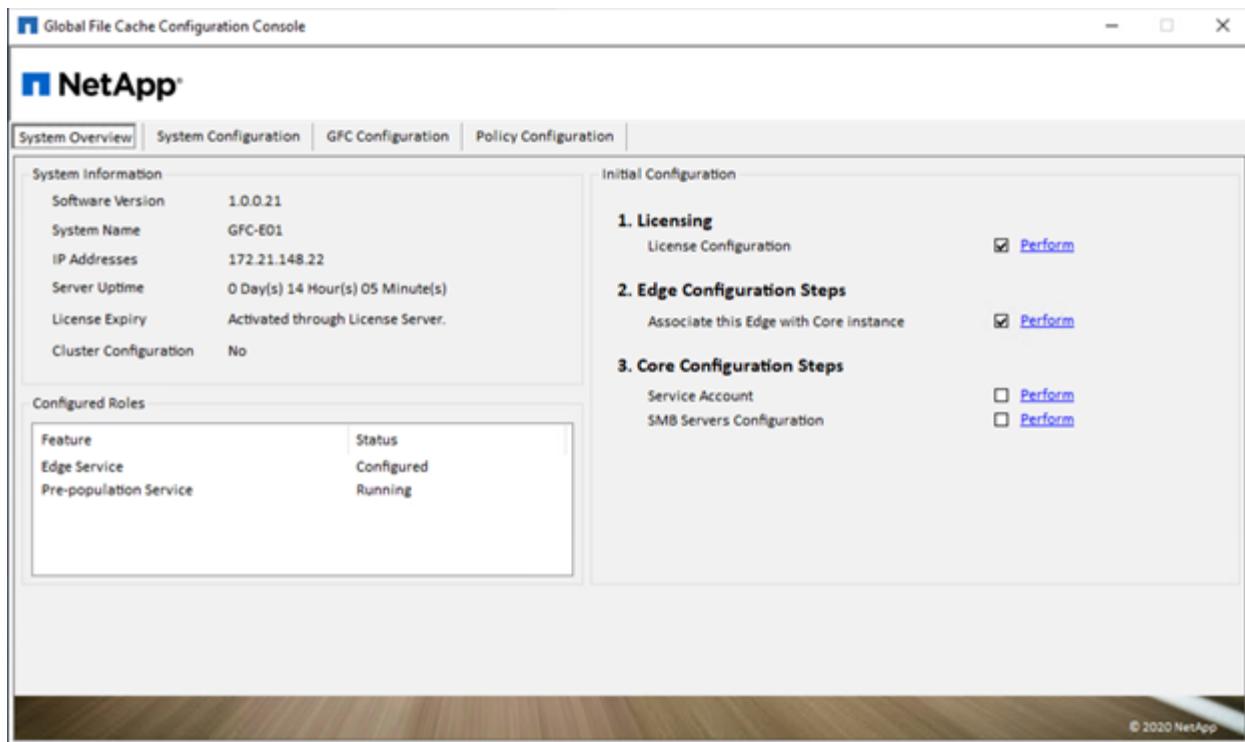
2. Provide the service account to be used for the Global File Cache. For the required permissions for this account, see the [GFC documentation](#).



3. Add a new backend file server and provide the file server name or IP.



4. On the edge, the cache drive must have the drive letter D. If it does not, use diskpart.exe to select the volume and change drive letter. Register with the license server as edge.



If core auto-configuration is enabled, core information is retrieved from the license management server automatically.

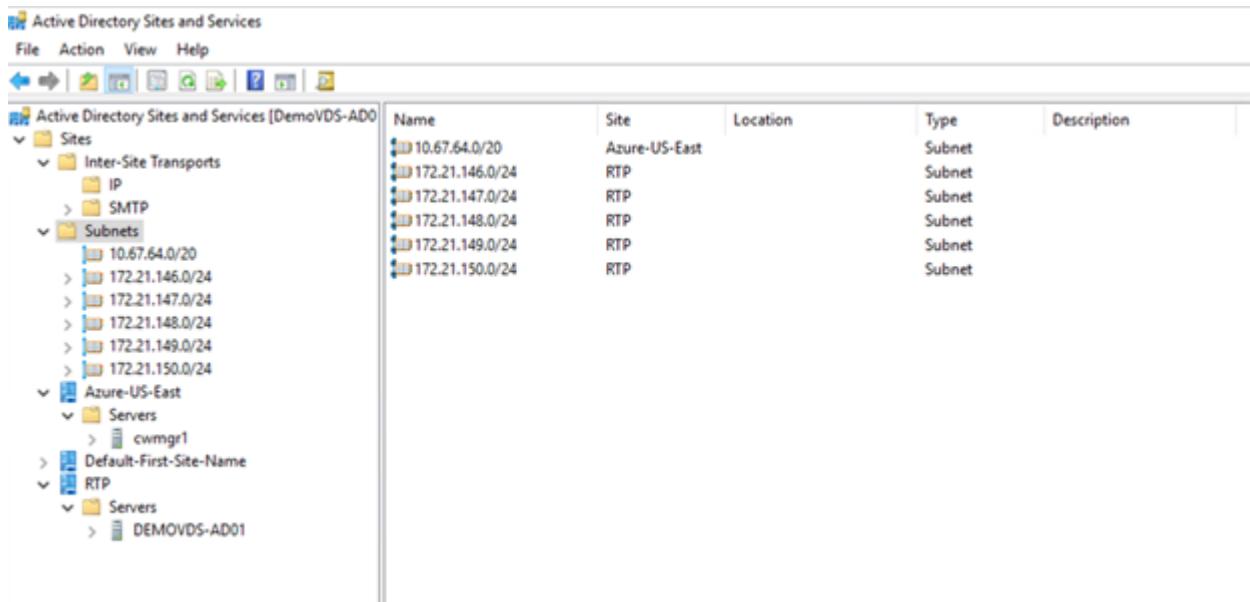
From any client machine, the administrators that used to access the share on the file server can access it with GFC edge using UNC Path \\<edge server name>\FASTDATA\<core server name>\<backend file server name>\<share name>. Administrators can include this path in user logonscript or GPO for users drive mapping at the edge location.

To provide transparent access for users across the globe, an administrator can setup the Microsoft Distributed

## Filesystem (DFS) with links pointing to file server shares and to edge locations.



When users log in with Active Directory credentials based on the subnets associated with the site, the appropriate link is utilized by the DFS client to access the data.

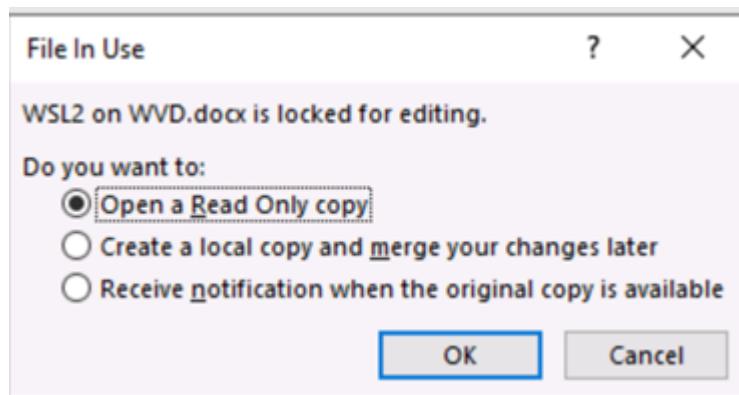


File icons change depending on whether a file is cached; files that are not cached have a grey X on the lower left corner of the icon. After a user in an edge location accesses a file, that file is cached, and the icon changes.

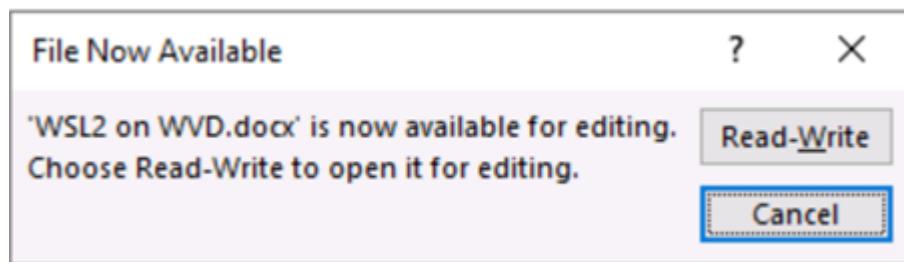
A screenshot of a Windows File Explorer window. The address bar shows the path: Network > demovds.com > DFS > CW-M6TX > Data. The left sidebar has sections for Quick access, This PC, and Network. The main area displays a list of files and folders with columns for Name, Date modified, Type, and Size. The list includes various file types like Department (File folder), Outlook (File folder), and several documents (PDF, XLSX, DOC, MM, PPT).

	Name	Date modified	Type	Size
★ Quick access	Department	10/1/2020 5:28 PM	File folder	
Desktop	Outlook	10/12/2020 3:05 PM	File folder	
Downloads	Outlook Files	10/12/2020 6:07 PM	File folder	
Documents	Output	10/12/2020 3:12 PM	File folder	
Pictures	WindowsPowerShell	10/11/2020 6:24 PM	File folder	
This PC	FSLogix	10/11/2020 9:11 PM	Registration Entries	2 KB
Network	GFC-1-0-0-21-Release	10/11/2020 10:05 ...	Application	26,869 KB
	PDF1.pdf	6/22/2016 9:31 PM	PDF File	1,101 KB
	PDF2.pdf	6/22/2016 9:31 PM	PDF File	1,066 KB
	Spreadsheet.xlsx	6/22/2016 9:31 PM	XLSX File	298 KB
	UserEdit.doc	6/22/2016 9:31 PM	DOC File	1,061 KB
	UserEdit1.doc	10/12/2020 3:13 PM	DOC File	1,061 KB
	UserEdit2.doc	10/12/2020 3:01 PM	DOC File	1,063 KB
	UserMindmap.mm	6/22/2016 9:31 PM	MM File	86 KB
	UserPresentation.ppt	6/22/2016 9:31 PM	PPT File	3,071 KB

When a file is open and another user is trying to open the same file from an edge location, the user is prompted with the following selection:



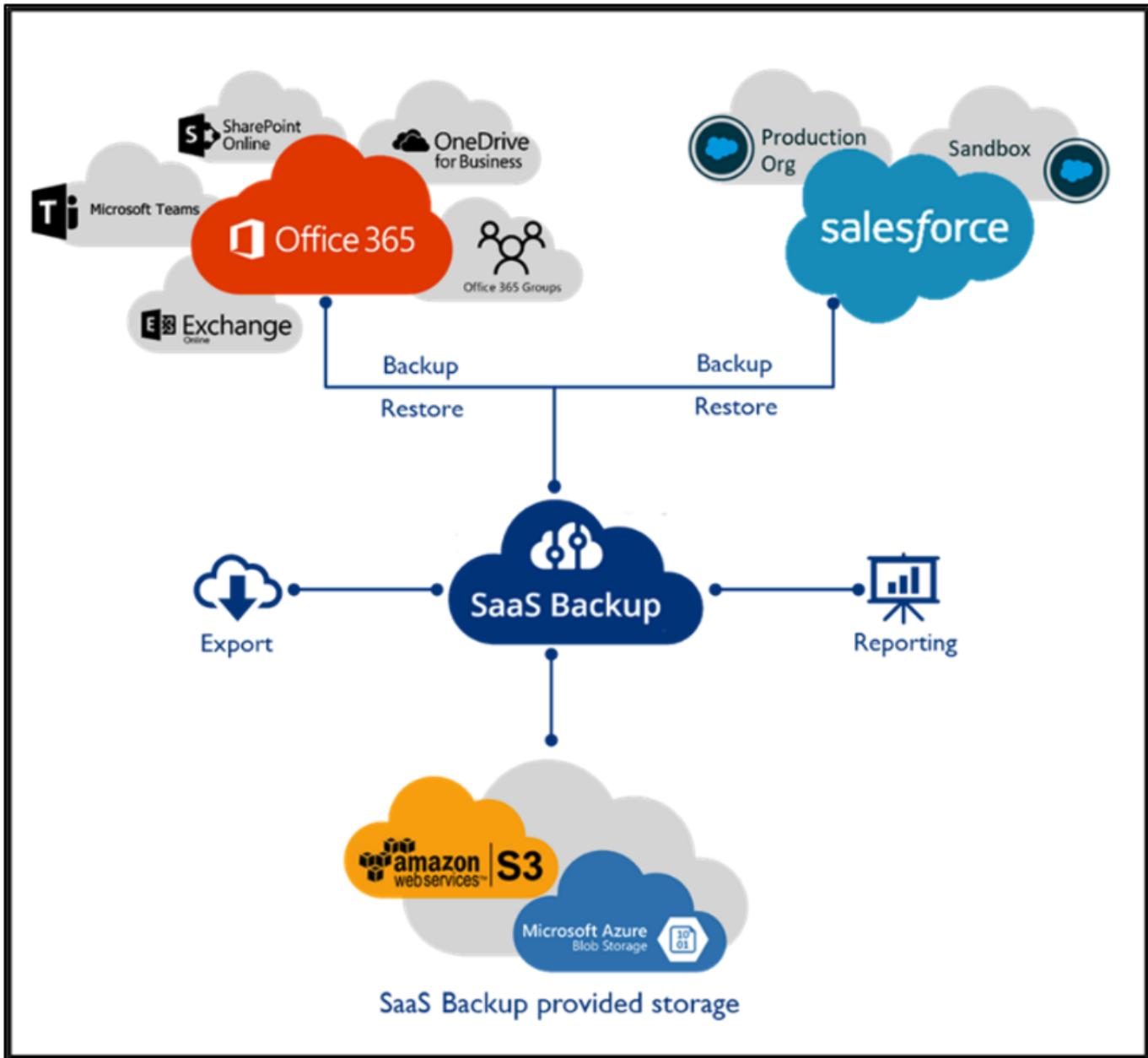
If the user selects the option to receive a notification when the original copy is available, the user is notified as follows:



For more information, see this [video on Talon and Azure NetApp Files Deployment](#).

## SaaS Backup

NetApp VDS provides data protection for Salesforce and Microsoft Office 365, including Exchange, SharePoint, and Microsoft OneDrive. The following figure shows how NetApp VDS provides SaaS Backup for these data services.



For a demonstration of Microsoft Office 365 data protection, see [this video](#).

For a demonstration of Salesforce data protection, see [this video](#).

**Next: Operation Management**

## Operation management

With NetApp VDS, administrators can delegate tasks to others. They can connect to deployed servers to troubleshoot, view logs, and run audit reports. While assisting customers, helpdesk or level-3 technicians can shadow user sessions, view process lists, and kill processes if required.

For information on VDS logfiles, see the [Troubleshooting Failed VDA Actions page](#).

For more information on the required minimum permissions, see the [VDA Components and Permissions page](#).

If you would like to manually clone a server, see the [Cloning Virtual Machines page](#).

To automatically increase the VM disk size, see the [Auto-Increase Disk Space Feature page](#).

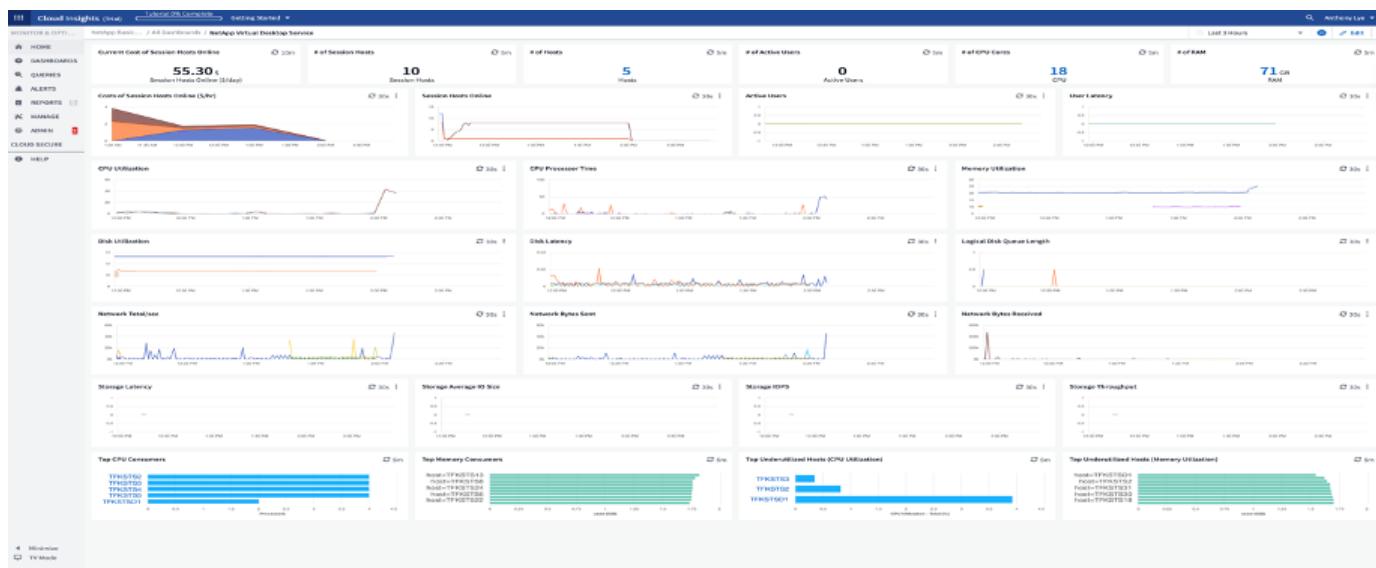
To identify the gateway address to manually configure the client, see the [End User Requirements page](#).

## Cloud Insights

NetApp Cloud Insights is a web-based monitoring tool that gives you complete visibility into infrastructure and applications running on NetApp and other third-party infrastructure components. Cloud Insights supports both private cloud and public clouds for monitoring, troubleshooting, and optimizing resources.

Only the acquisition unit VM (can be Windows or Linux) must be installed on a private cloud to collect metrics from data collectors without the need for agents. Agent-based data collectors allow you to pull custom metrics from Windows Performance Monitor or any input agents that Telegraf supports.

The following figure depicts the Cloud Insights VDS dashboard.



For more info on NetApp Cloud Insights, see [this video](#).

Next: [Tools and logs](#)

## Tools and Logs

### DCCconfig Tool

The DCCconfig tool supports the following hypervisor options for adding a site:

DataCenter Site

DataCenter Site	Site 3	Cancel New	Save
Hypervisor	Select Hypervisor ▾	Load Hypervisor	Test
<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <b>Select Hypervisor</b>            Aws            AzureClassic            AzureRM            ComputeEngine            HyperV            ProfitBricks            vCloud            vCloudRest            vSphere            XenServer         </div>			

Configuration

DataCenter	Accounts	Email	DatabaseConnection	Exclude	DataCenter Sites	Product Keys	Static IpAddress	Drive Mapping	
------------	----------	-------	--------------------	---------	------------------	--------------	------------------	---------------	--

		Save
	Description	DriveLetter
	Shared Data	P
	FTP	F
▶	User Home	H

Workspace-specific drive-letter mapping for shared data can be handled using GPO. Professional Services or the support team can use the advanced tab to customize settings like Active Directory OU names, the option to enable or disable deployment of FSLogix, various timeout values, and so on.

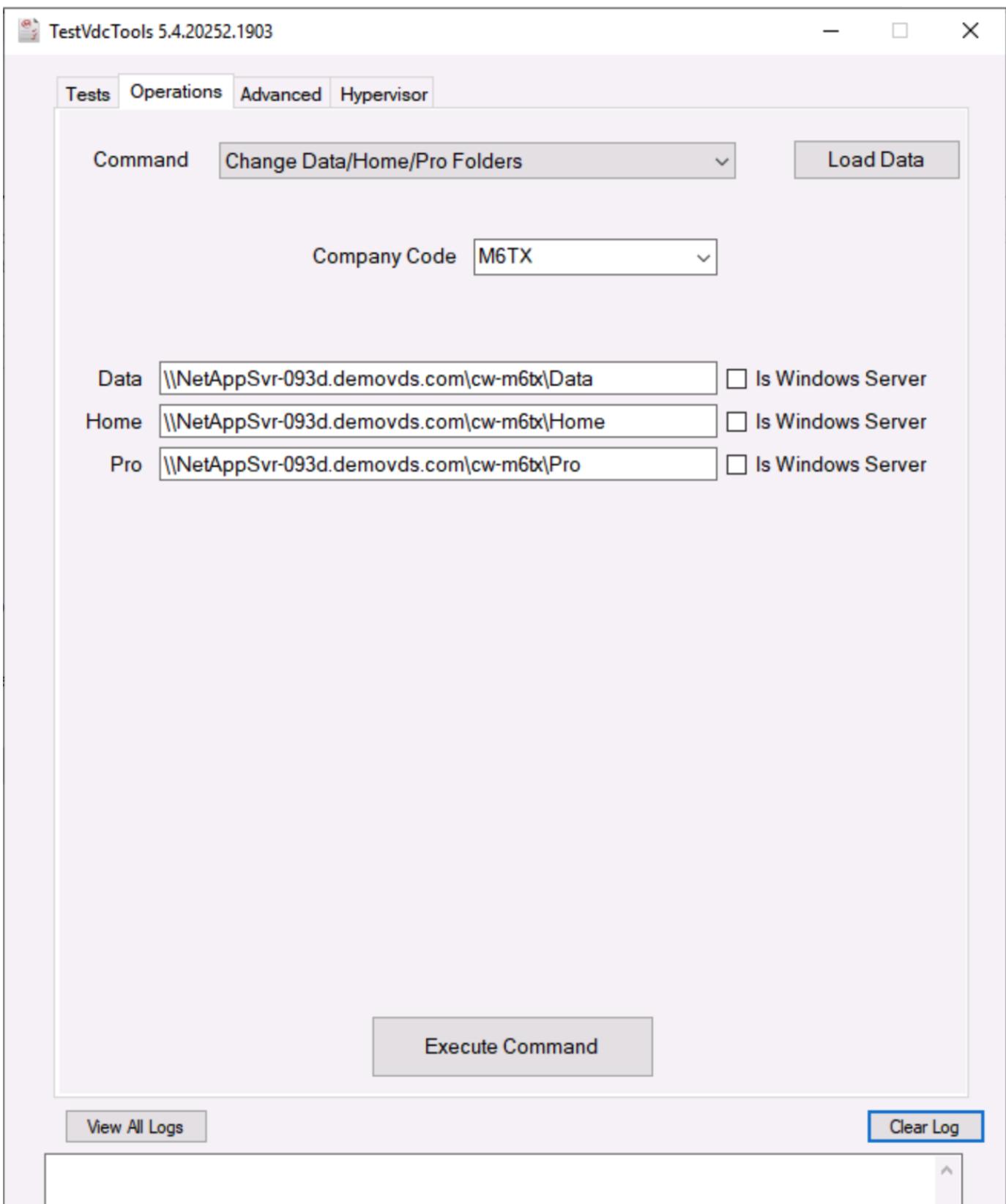
GroupName	FriendlyName	Value
Server Creation	UpdateVMNameWhenRemovedFromCache	<input type="checkbox"/>
Server Creation	UpdateVmIrevalRules	<input checked="" type="checkbox"/>
Server Creation	WaitAfterRebootMin	6
Server Creation	WaitAfterHypervisorCreateMin	1
Server Creation	WaitAfterSysPrepMin	10
Server Creation	WaitAfterSysPrepOr2008ServersMin	30
Server Creation	GFI Agent Path	
Server Creation	Automated Cloning Enabled	<input checked="" type="checkbox"/>
Server Creation	CompaniesOU	Cloud Workspace Companies
Server Creation	Install ThinPrint v11	<input checked="" type="checkbox"/>
Server Creation	ServersOU	Cloud Workspace Servers
Server Creation	Install FLogix	<input checked="" type="checkbox"/>
Server Creation	Use Default OUs	<input checked="" type="checkbox"/>
Server Creation	Max Threads	50
Server Creation	Wait for DNS to Update Minutes	15
Check Vdc Tools Version	Run Every X Minutes	5
Daily Actions	Enabled	<input checked="" type="checkbox"/>
Daily Actions	Run at startup	<input checked="" type="checkbox"/>
Generate Reports	Time Of Day	06:00
Daily Maintenance	Enabled	<input checked="" type="checkbox"/>
Daily Maintenance	Time Of Day	08:01
Weekly Maintenance	Enabled	<input checked="" type="checkbox"/>
Weekly Maintenance	Time Of Day	00:01
Automatic Resource Allocation	Day	Sunday
Resource Allocation	Enabled	<input checked="" type="checkbox"/>
EmailReports	Use Data Center Defaults	<input checked="" type="checkbox"/>
Server Heartbeat	IncludeEmailAttachment	<input type="checkbox"/>
Server Heartbeat	Interval Minutes	15

## Command Center (Previously known as TestVdc Tools)

To launch Command Center and the required role, see the [Command Center Overview](#).

You can perform the following operations:

- Change the SMB Path for a workspace.



- Change the site for provisioning collection.

TestVdcTools 5.4.20252.1903

Tests Operations Advanced Hypervisor

Command Edit Provisioning Collection

Provisioning Collection Windows2019

Description On vSphere Site 2

Share Drive P

Minimum Cache Level 1

Operating System Windows Server 2019

Collection Type Shared

	Data Center Site	Role	Template	Storage
▶	Site 2	TSData	Windows2019	DS01
*				

< >

Execute Command

Log Files

Name	Date modified	Type	Size
CwAgent	9/19/2020 12:35 PM	File folder	
CWAutomationService	9/19/2020 12:34 PM	File folder	
CWManagerX	9/19/2020 12:53 PM	File folder	
CwVmAutomationService	9/19/2020 12:34 PM	File folder	
TestVdcTools	9/22/2020 8:20 PM	File folder	
report	9/19/2020 12:18 PM	Executable Jar File	705 KB

Check [automation logs](#) for more info.

Next: Conclusion

## GPU considerations

GPUs are typically used for graphic visualization (rendering) by performing repetitive arithmetic calculations. This repetitive compute capability is often used for AI and deep learning use cases.

For graphic intensive applications, Microsoft Azure offers the NV series based on the NVIDIA Tesla M60 card with one to four GPUs per VM. Each NVIDIA Tesla M60 card includes two Maxwell-based GPUs, each with 8GB of GDDR5 memory for a total of 16GB.



An NVIDIA license is included with the NV series.

## TechPowerUp GPU-Z 2.36.0

- X

Graphics Card

Sensors

Advanced

Validation



Name

NVIDIA Tesla M60

Lookup

GPU

GM204

Revision

FF

Technology

28 nm

Die Size

398 mm<sup>2</sup>

Release Date

Aug 30, 2015

Transistors

5200M

BIOS Version

84.04.85.00.03



UEFI

Subvendor

NVIDIA

Device ID

10DE 13F2 - 10DE 115E

ROPs/TMUs

64 / 128

Bus Interface

PCI

?

Shaders

2048 Unified

DirectX Support

12 (12\_1)

Pixel Fillrate

75.4 GPixel/s

Texture Fillrate

150.8 GTexel/s

Memory Type

GDDR5 (Hynix)

Bus Width

256 bit

Memory Size

8192 MB

Bandwidth

160.4 GB/s

Driver Version

27.21.14.5257 (NVIDIA 452.57) / 2016

Driver Date

Oct 22, 2020

Digital Signature

WHQL

GPU Clock

557 MHz

Memory

1253 MHz

Boost

1178 MHz

Default Clock

557 MHz

Memory

1253 MHz

Boost

1178 MHz

NVIDIA SLI

Disabled

Computing

 OpenCL CUDA DirectCompute DirectML

Technologies

 Vulkan Ray Tracing PhysX OpenGL 4.6

NVIDIA Tesla M60

Close

With NetApp HCI, the H615C GPU contains three NVIDIA Tesla T4 cards. Each NVIDIA Tesla T4 card has a Touring-based GPU with 16GB of GDDR6 memory. When used in a VMware vSphere environment, virtual machines are able to share the GPU, with each VM having dedicated frame buffer memory. Ray tracing is available with the GPUs on the NetApp HCI H615C to produce realistic images including light reflections. Please note that you need to have an NVIDIA license server with a license for GPU features.

## TechPowerUp GPU-Z 2.36.0

— □ ×

Graphics Card

Sensors

Advanced

Validation



Name

NVIDIA GRID T4-8Q

Lookup

GPU

TU104

Revision

A1



Technology

12 nm

Die Size

545 mm<sup>2</sup>

Release Date

Sep 13, 2018

Transistors

13600M

BIOS Version

0.00.00.00.00



UEFI

Subvendor

NVIDIA

Device ID

10DE 1EB8 - 10DE 130F

ROPs/TMUs

8 / 160

Bus Interface

PCI

?

Shaders

2560 Unified

DirectX Support

12 (12\_2)

Pixel Fillrate

4.7 GPixel/s

Texture Fillrate

93.6 GTexel/s

Memory Type

GDDR6

Bus Width

256 bit

Memory Size

8192 MB

Bandwidth

Unknown

Driver Version

27.21.14.5257 (NVIDIA 452.57) / 2016

Driver Date

Oct 22, 2020

Digital Signature

WHQL

GPU Clock

585 MHz

Memory

0 MHz

Shader

N/A

Default Clock

585 MHz

Memory

0 MHz

Shader

N/A

NVIDIA SLI

Disabled

Computing

 OpenCL CUDA DirectCompute DirectML

Technologies

 Vulkan Ray Tracing PhysX OpenGL 4.6

NVIDIA GRID T4-8Q

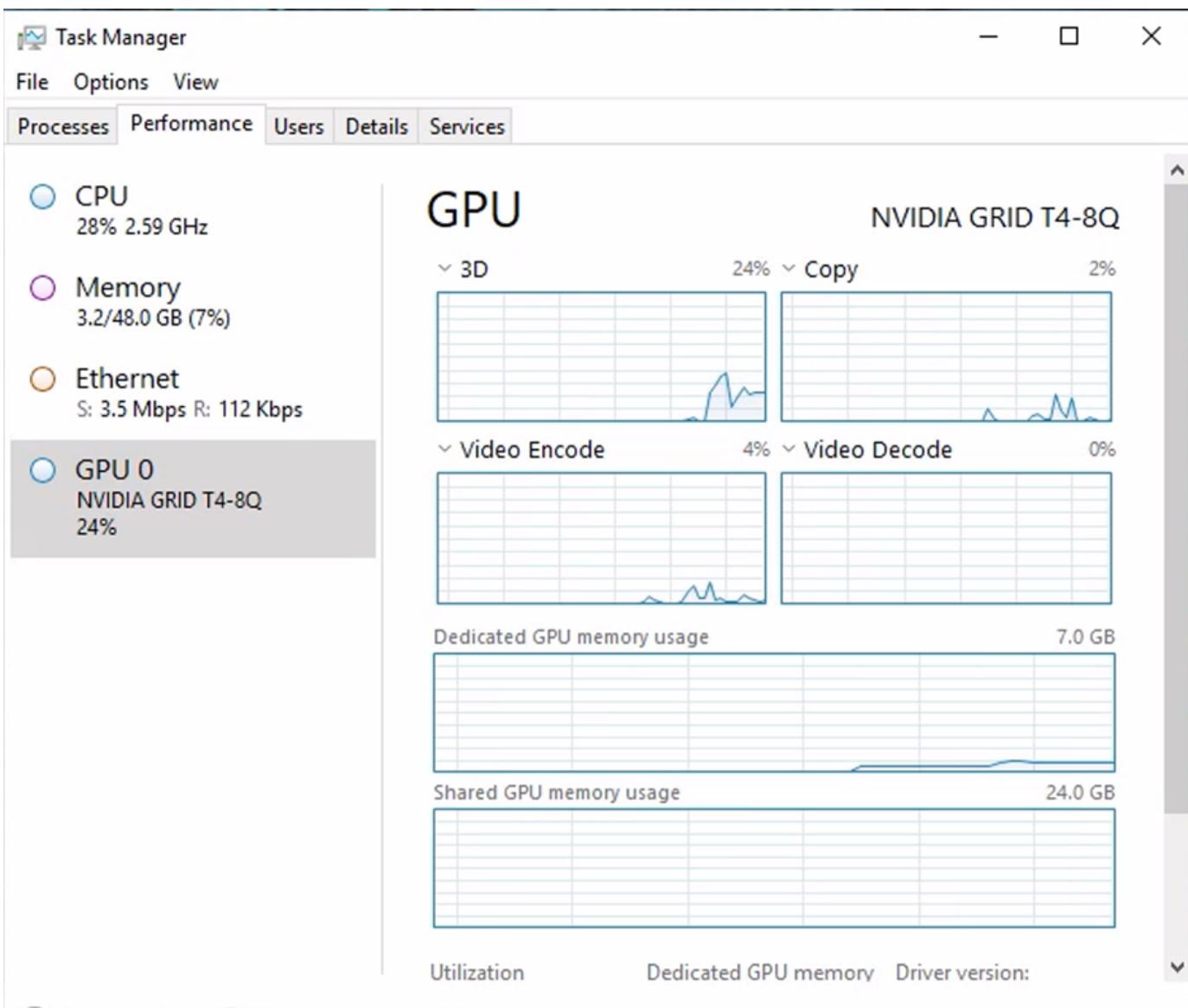
Close

To use the GPU, you must install the appropriate driver, which can be downloaded from the NVIDIA license portal. In an Azure environment, the NVIDIA driver is available as GPU driver extension. Next, the group policies in the following screenshot must be updated to use GPU hardware for remote desktop service sessions. You should prioritize H.264 graphics mode and enable encoder functionality.

The screenshot shows the Local Group Policy Editor window. The left pane displays a tree view of policy settings under 'Microsoft account' and 'Remote Desktop Services'. The right pane lists specific policy settings with their current state and comments:

Setting	State	Comment
RemoteFX for Windows Server 2008 R2	Not configured	No
Limit maximum color depth	Not configured	No
Enforce Removal of Remote Desktop Wallpaper	Enabled	No
Use hardware graphics adapters for all Remote Desktop Services sessions	Not configured	No
Limit maximum display resolution	Not configured	No
Limit number of monitors	Not configured	No
Remove "Disconnect" option from Shut Down dialog	Not configured	No
Remove Windows Security item from Start menu	Not configured	No
Use advanced RemoteFX graphics for RemoteApp	Not configured	No
Prioritize H.264/AVC 444 graphics mode for Remote Desktop Connections	Enabled	No
Configure H.264/AVC hardware encoding for Remote Desktop Connections	Enabled	No
Configure compression for RemoteFX data	Not configured	No
Configure image quality for RemoteFX Adaptive Graphics	Not configured	No
Enable RemoteFX encoding for RemoteFX clients designed for Windows Server 2008 R2 SP1	Not configured	No
Configure RemoteFX Adaptive Graphics	Not configured	No
Start a program on connection	Not configured	No
Always show desktop on connection	Not configured	No
Allow desktop composition for remote desktop sessions	Not configured	No
Do not allow font smoothing	Not configured	No

Validate GPU performance monitoring with Task Manager or by using the nvidia-smi CLI when running WebGL samples. Make sure that GPU, memory, and encoder resources are being consumed.



To make sure that the virtual machine is deployed to the NetApp HCI H615C with Virtual Desktop Service, define a site with the vCenter cluster resource that has H615C hosts. The VM template must have the required vGPU profile attached.

For shared multi-session environments, consider allocating multiple homogenous vGPU profiles. However, for high end professional graphics application, it is better to have each VM dedicated to a user to keep VMs isolated.

The GPU processor can be controlled by a QoS policy, and each vGPU profile can have dedicated frame buffers. However, the encoder and decoder are shared for each card. The placement of a vGPU profile on a GPU card is controlled by the vSphere host GPU assignment policy, which can emphasize performance (spread VMs) or consolidation (group VMs).

[Next: Solutions for industry.](#)

## Solutions for Industry

Graphics workstations are typically used in industries such as manufacturing, healthcare, energy, media and entertainment, education, architecture, and so on. Mobility is often limited for graphics-intensive applications.

To address the issue of mobility, Virtual Desktop Services provide a desktop environment for all types of workers, from task workers to expert users, using hardware resources in the cloud or with NetApp HCI, including options for flexible GPU configurations. VDS enables users to access their work environment from anywhere with laptops, tablets, and other mobile devices.

To run manufacturing workloads with software like ANSYS Fluent, ANSYS Mechanical, Autodesk AutoCAD, Autodesk Inventor, Autodesk 3ds Max, Dassault Systèmes SOLIDWORKS, Dassault Systèmes CATIA, PTC Creo, Siemens PLM NX, and so on, the GPUs available on various clouds (as of Jan 2021) are listed in the following table.

GPU Model	Microsoft Azure	Google Compute (GCP)	Amazon Web Services (AWS)	On-Premises (NetApp HCI)
NVIDIA M60	Yes	Yes	Yes	No
NVIDIA T4	No	Yes	Yes	Yes
NVIDIA P100	No	Yes	No	No
NVIDIA P4	No	Yes	No	No

Shared desktop sessions with other users and dedicated personal desktops are also available. Virtual desktops can have one to four GPUs or can utilize partial GPUs with NetApp HCI. The NVIDIA T4 is a versatile GPU card that can address the demands of a wide spectrum of user workloads.

Each GPU card on NetApp HCI H615C has 16GB of frame buffer memory and three cards per server. The number of users that can be hosted on single H615C server depends on the user workload.

Users/Server	Light (4GB)	Medium (8GB)	Heavy (16GB)
H615C	12	6	3

To determine the user type, run the GPU profiler tool while users are working with applications performing typical tasks. The GPU profiler captures memory demands, the number of displays, and the resolution that users require. You can then pick the vGPU profile that satisfies your requirements.

Virtual desktops with GPUs can support a display resolution of up to 8K, and the utility nView can split a single monitor into regions to work with different datasets.

With ONTAP file storage, you can realize the following benefits:

- A single namespace that can grow up to 20PB of storage with 400 billion of files, without much administrative input
- A namespace that can span the globe with a Global File Cache
- Secure multitenancy with managed NetApp storage
- The migration of cold data to object stores using NetApp FabricPool
- Quick file statistics with file system analytics
- Scaling a storage cluster up to 24 nodes increasing capacity and performance
- The ability to control storage space using quotas and guaranteed performance with QoS limits
- Securing data with encryption
- Meeting broad requirements for data protection and compliance
- Delivering flexible business continuity options

[Next: Conclusion](#)

## Conclusion

The NetApp Virtual Desktop Service provides an easy-to-consume virtual desktop and application environment with a sharp focus on business challenges. By extending VDS with the on-premises ONTAP environment, you can use powerful NetApp features in a VDS environment, including rapid clone, in-line deduplication, compaction, thin provisioning, and compression. These features save storage costs and improve performance with all-flash storage. With VMware vSphere hypervisor, which minimizes server-provisioning time by using Virtual Volumes and vSphere API for Array integration. Using the hybrid cloud, customers can pick the right environment for their demanding workloads and save money. The desktop session running on-premises can access cloud resources based on policy.

[Next: Where to Find Additional Information](#)

## Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- [NetApp Cloud](#)
- [NetApp VDS Product Documentation](#)
- [Connect your on-premises network to Azure with VPN Gateway](#)
- [Azure Portal](#)
- [Microsoft Windows Virtual Desktop](#)
- [Azure NetApp Files Registration](#)

## VMware Horizon

## Citrix Virtual Apps and Desktops

### TR-4854: NetApp HCI for Citrix Virtual Apps and Desktops with Citrix Hypervisor

Suresh Thoppay, NetApp

NetApp HCI infrastructure allows you to start small and build in small increments to meet the demands of virtual desktop users. Compute or storage nodes can be added or removed to address changing business requirements.

Citrix Virtual Apps and Desktops provides a feature-rich platform for end-user computing that addresses various deployment needs, including support for multiple hypervisors. The premium edition of this software includes tools to manage images and user policies.

Citrix Hypervisor (formerly known as Citrix Xen Hypervisor) provides additional features to Citrix Virtual Apps and Desktops compared to running on other hypervisor platforms. The following are key benefits of running on Citrix Hypervisor:

- A Citrix Hypervisor license is included with all versions of Citrix Virtual Apps and Desktops. This licensing helps to reduce the cost of running the Citrix Virtual Apps and Desktops platform.
- Features like PVS Accelerator and Storage Accelerator are only available with Citrix Hypervisor.

- For Citrix solutions, the Citrix Hypervisor is the preferred workload choice.
- Available in Long Term Service Release (LTSR; aligns with Citrix Virtual Apps and Desktops) and Current Release (CR) options.

## Abstract

This document reviews the solution architecture for Citrix Virtual Apps and Desktops with Citrix Hypervisor. It provides best practices and design guidelines for Citrix implementation on NetApp HCI. It also highlights multitenancy features, user profiles, and image management.

## Solution Overview

Service providers who deliver the Virtual Apps and Desktops service prefer to host it on Citrix Hypervisor to reduce cost and for better integration. The NetApp Deployment Engine (NDE), which performs automated installation of VMware vSphere on NetApp HCI, currently doesn't support deployment of Citrix Hypervisor. Citrix Hypervisor can be installed on NetApp HCI using PXE boot or installation media or other deployment methods supported by Citrix.

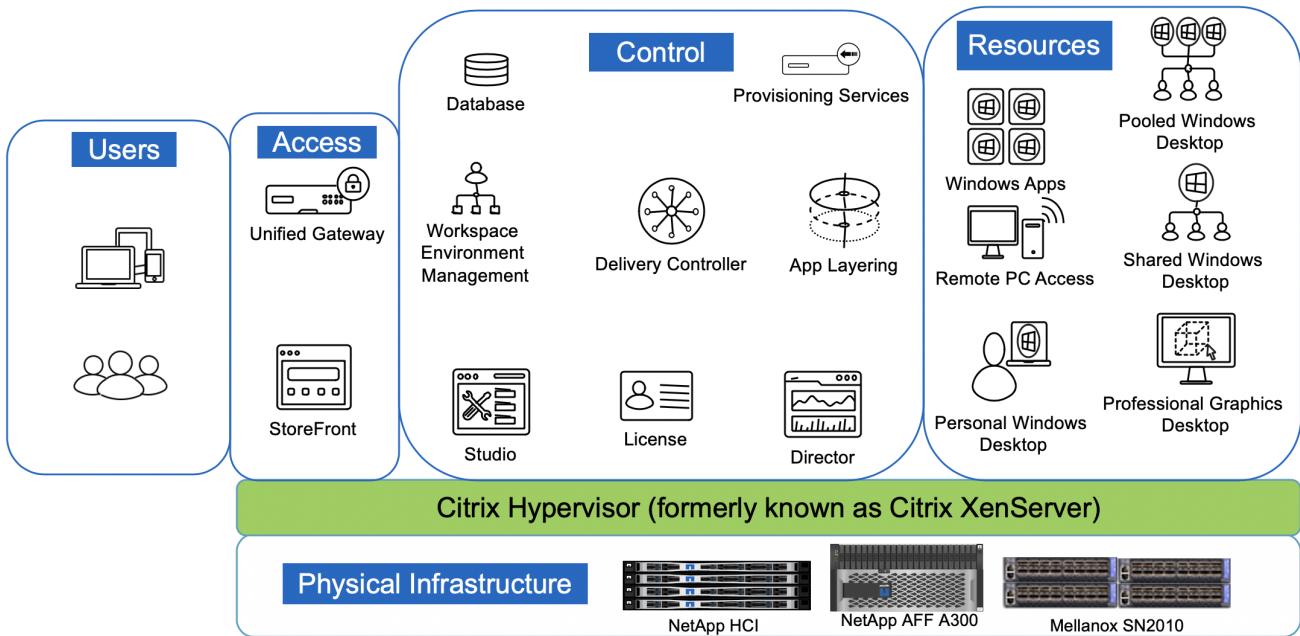
Citrix Virtual Apps and Desktops can automate the provisioning of desktops and session hosts either using Citrix Provisioning (network-based) or by Machine Creation Services (hypervisor storage-based). Both Microsoft Windows-based OSs and popular Linux flavors are supported. Existing physical workstations, desktop PCs, and VMs on other hypervisors that are not enabled for auto-provisioning can also be made available for remote access by installing the agents.

The Citrix Workspace Application, a client software used to access Virtual Apps and Desktops, is supported on various devices including tablets and mobile phones. Virtual Apps and Desktops can be accessed using a browser-based HTML5 interface internally or externally to the deployment location.

Based on your business needs, the solution can be extended to multiple sites. However, remember that NetApp HCI storage efficiencies operate on a per-cluster basis.

The following figure shows the high-level architecture of the solution. The access, control, and resource layers are deployed on top of Citrix Hypervisor as virtual machines. Citrix Hypervisor runs on NetApp HCI compute nodes. The virtual disk images are stored in the iSCSI storage repository on NetApp HCI storage nodes.

A NetApp AFF A300 is used in this solution for SMB file shares to store user profiles with FSLogix containers, Citrix profile management (for multisession write-back support), Elastic App Layering images, and so on. We also use SMB file share to mount ISO images on Citrix Hypervisor.



A Mellanox SN2010 switch is used for 10/25/100Gb Ethernet connectivity. Storage nodes use SFP28 transceivers for 25Gb connection, compute nodes use SFP/SFP+ transceivers for 10Gb connection, and interswitch links are QSFP28 transceivers for a 100Gb connection.

Storage ports are configured with multichassis link aggregation (MLAG) to provide total throughput of 50Gb and are configured as trunk ports. Compute node ports are configured as hybrid ports to create a VLAN for iSCSI, XenMotion, and workload VLANs.

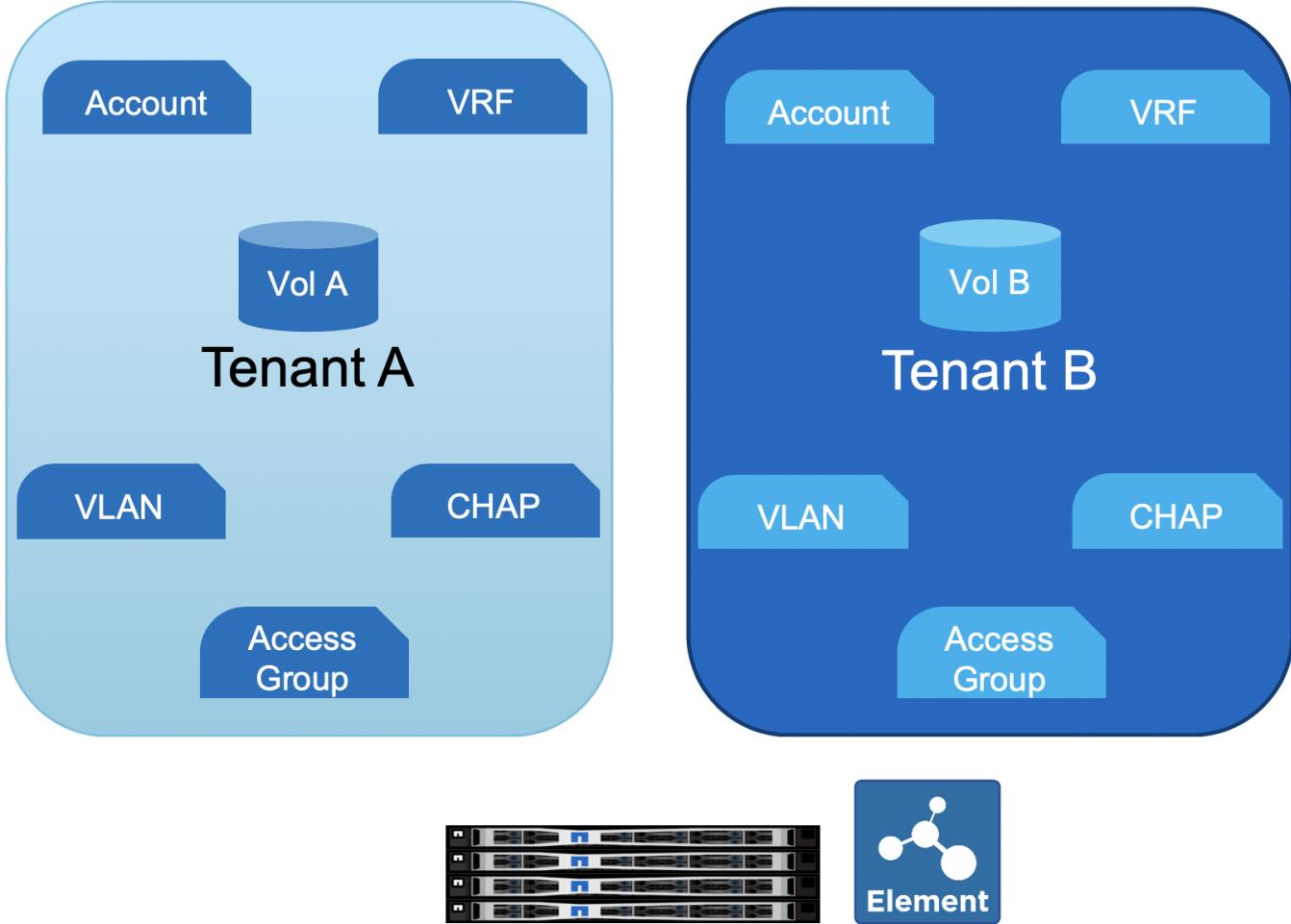
## Physical Infrastructure

### NetApp HCI

NetApp HCI is available as compute nodes or storage nodes. Depending on the storage node model, a minimum of two to four nodes is required to form a cluster. For the compute nodes, a minimum of two nodes are required to provide high availability. Based on demand, nodes can be added one at a time to increase compute or storage capacity.

A management node (mNode) deployed on a compute node runs as a virtual machine on supported hypervisors. The mNode is used for sending data to ActiveIQ (a SaaS-based management portal), to host a hybrid cloud control portal, as a reverse proxy for remote support of NetApp HCI, and so on.

NetApp HCI enables you to have nondistributive rolling upgrades. Even when one node is down, data is serviced from the other nodes. The following figure depicts NetApp HCI storage multitenancy features.

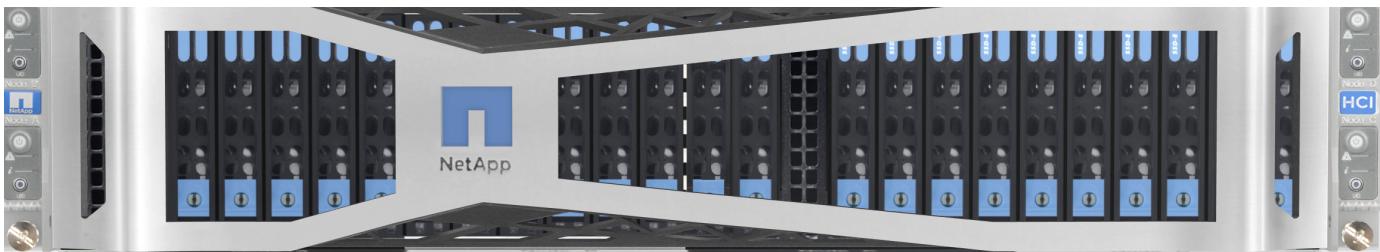


NetApp HCI Storage provides flash storage through iSCSI connection to compute nodes. iSCSI connections can be secured using CHAP credentials or a volume access group. A volume access group only allows authorized initiators to access the volumes. An account holds a collection of volumes, the CHAP credential, and the volume access group. To provide network-level separation between tenants, different VLANs can be used, and volume access groups also support virtual routing and forwarding (VRF) to ensure the tenants can have same or overlapping IP subnets.

A RESTful web interface is available for custom automation tasks. NetApp HCI has PowerShell and Ansible modules available for automation tasks. For more info, see [NetApp.IO](#).

## Storage Nodes

NetApp HCI supports two storage node models: the H410S and H610S. The H410 series comes in a 2U chassis containing four half-width nodes. Each node has six SSDs of sizes 480GB, 960GB, or 1.92TB with the option of drive encryption. The H410S can start with a minimum of two nodes. Each node delivers 50,000 to 100,000 IOPS with a 4K block size. The following figure presents a front and back view of an H410S storage node.



The H610S is a 1U storage node with 12 NVMe drives of sizes 960GB, 1.92TB, or 3.84TB with the option of drive encryption. A minimum of four H610S nodes are required to form a cluster. It delivers around 100,000 IOPS per node with a 4K block size. The following figure depicts a front and back view of an H610S storage node.

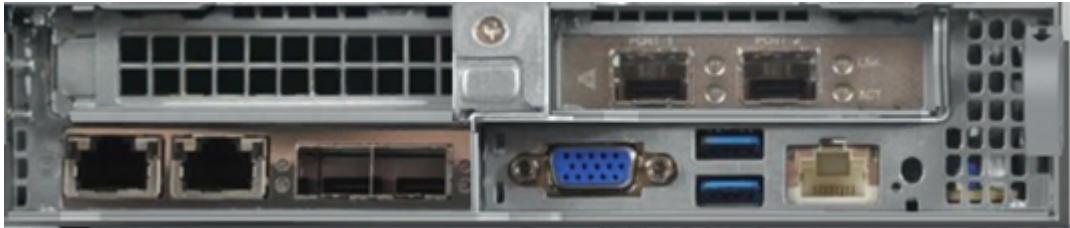


In a single cluster, there can be a mix of storage node models. The capacity of a single node can't exceed 1/3 of the total cluster size. The storage nodes come with two network ports for iSCSI (10/25GbE – SFP28) and two ports for management (1/10GbE – RJ45). A single out-of-band 1GbE RJ45 management port is also available.

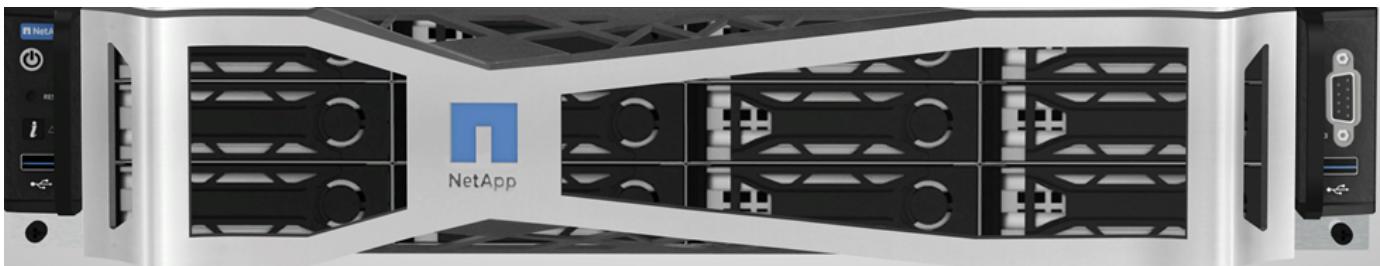
## Compute Nodes

NetApp HCI compute nodes are available in three models: H410C, H610C, and H615C. Compute nodes are all RedFish API-compatible and provide a BIOS option to enable Trusted Platform Module (TPM) and Intel Trusted eXecution Technology (TXT).

The H410C is a half-width node that can be placed in a 2U chassis. The chassis can have a mix of compute and storage nodes. The H410C comes with first-generation Intel Xeon Silver/Gold scalable processors with 4 to 20 cores in dual-socket configurations. The memory size ranges from 384GB to 1TB. There are four 10/25GbE (SFP28) ports and two 1GbE RJ45 ports, with one 1GbE RJ45 port available for out-of-band management. The following figure depicts a front and back view of an H410C compute node.



The H610C is 2RU and has a dual-socket first generation Intel Xeon Gold 6130 scalable processor with 16 cores of 2.1GHz, 512GB RAM and two NVIDIA Tesla M10 GPU cards. This server comes with two 10/25GbE SFP28 ports and two 1GbE RJ45 ports, with one 1GbE RJ45 port available for out-of-band management. The following figure depicts a front and back view of an H610C compute node.



The H610C has two Tesla M10 cards providing a total of 64GB frame buffer memory with a total of 8 GPUs. It can support up to 64 personal virtual desktops with GPU enabled. To host more sessions per server, a shared desktop delivery model is available.

The H615C is a 1RU server with a dual socket for second-generation Intel Xeon Silver/Gold scalable processors with 4 to 24 cores per socket. RAM ranges from 384GB to 1.5TB. One model contains three NVIDIA Tesla T4 cards. The server includes two 10/25GbE (SFP28) and one 1GbE (RJ45) for out-of-band management. The following figure depicts a front and back view of an H615C compute node.





The H615C includes three Tesla T4 cards providing a total of 48GB frame buffer and three GPUs. The T4 card is a general-purpose GPU card that can be used for AI inference workloads as well as for professional graphics. It includes ray tracing cores that can help simulate light reflections.

## Hybrid Cloud Control

The Hybrid Cloud Control portal is often used for scaling out NetApp HCI by adding storage or/and compute nodes. The portal provides an inventory of NetApp HCI compute and storage nodes and a link to the ActiveIQ management portal. See the following screenshot of Hybrid Cloud Control.

Upgrades

MANAGEMENT SERVICES STORAGE COMPUTE

Upgrade Storage Cluster

Element is the operating system of your storage cluster. It includes software, NetApp Deployment Engine and firmware.

Select a storage cluster to see the latest compatible upgrades packages.

Cluster	Nodes	Current Version	Upgrade Status	Health Check Only
Storage_Cluster_01	36	Element 11.5	Upgrades Available	
Storage_Cluster_02	6	Element 11.3	Upgrades Available	

Select a package to begin an upgrade, or upload a NetApp-approved upgrade package.

Element-Version-12.0  
 Element-Version-11.7  
 Element-Version-11.5  
 Element-Version-11.3

[See release notes](#)

[Begin Upgrade](#)

## NetApp AFF

NetApp AFF provides an all-flash, scale-out file storage system, which is used as a part of this solution. ONTAP is the storage software that runs on NetApp AFF. Some key benefits of using ONTAP for SMB file storage are as follows:

- Storage Virtual Machines (SVM) for secure multitenancy
- NetApp FlexGroup technology for a scalable, high-performance file system
- NetApp FabricPool technology for capacity tiering. With FabricPool, you can keep hot data local and transfer cold data to cloud storage).

- Adaptive QoS for guaranteed SLAs. You can adjust QoS settings based on allocated or used space.
- Automation features (RESTful APIs, PowerShell, and Ansible modules)
- Data protection and business continuity features including NetApp Snapshot, NetApp SnapMirror, and NetApp MetroCluster technologies

## Mellanox Switch

A Mellanox SN2010 switch is used in this solution. However, you can also use other compatible switches. The following Mellanox switches are frequently used with NetApp HCI.

Model	Rack Unit	SFP28 (10/25GbE) ports	QSFP (40/100GbE) ports	Aggregate Throughput (Tbps)
SN2010	Half-width	18	4	1.7
SN2100	Half-width	—	16	3.2
SN2700	Full-width	—	32	6.4



QSFP ports support 4x25GbE breakout cables.

Mellanox switches are open Ethernet switches that allow you to pick the network operating system. Choices include the Mellanox Onyx OS or various Linux OSs such as Cumulus-Linux, Linux Switch, and so on. Mellanox switches also support the switch software development kit, the switch abstraction interface (SAI; part of the Open Compute Project), and Software for Open Networking in the Cloud (SONIC).

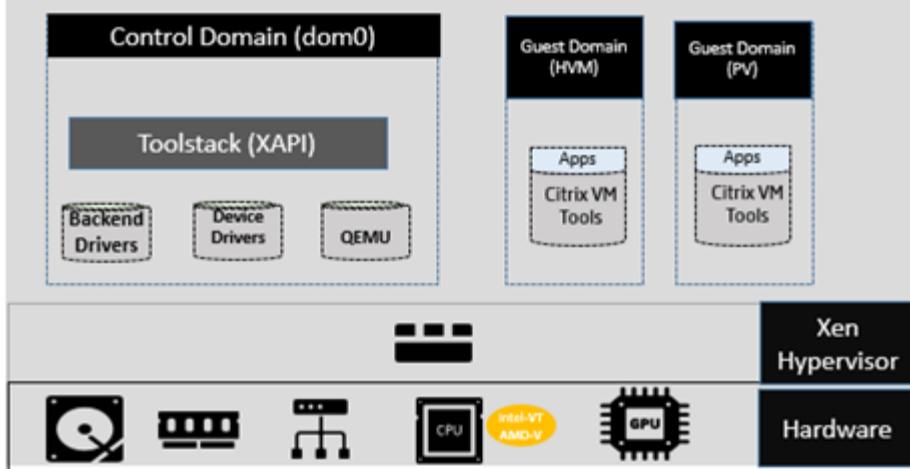
Mellanox switches provide low latency and support traditional data center protocols and tunneling protocols like VXLAN. VXLAN Hardware VTEP is available to function as an L2 gateway. These switches support various certified security standards like UC API, FIPS 140-2 (System Secure Mode), NIST 800-181A (SSH Server Strict Mode), and CoPP (IP Filter).

Mellanox switches support automation tools like Ansible, SALT Stack, Puppet, and so on. The Web Management Interface provides the option to execute multi-line CLI commands.

## Citrix Hypervisor

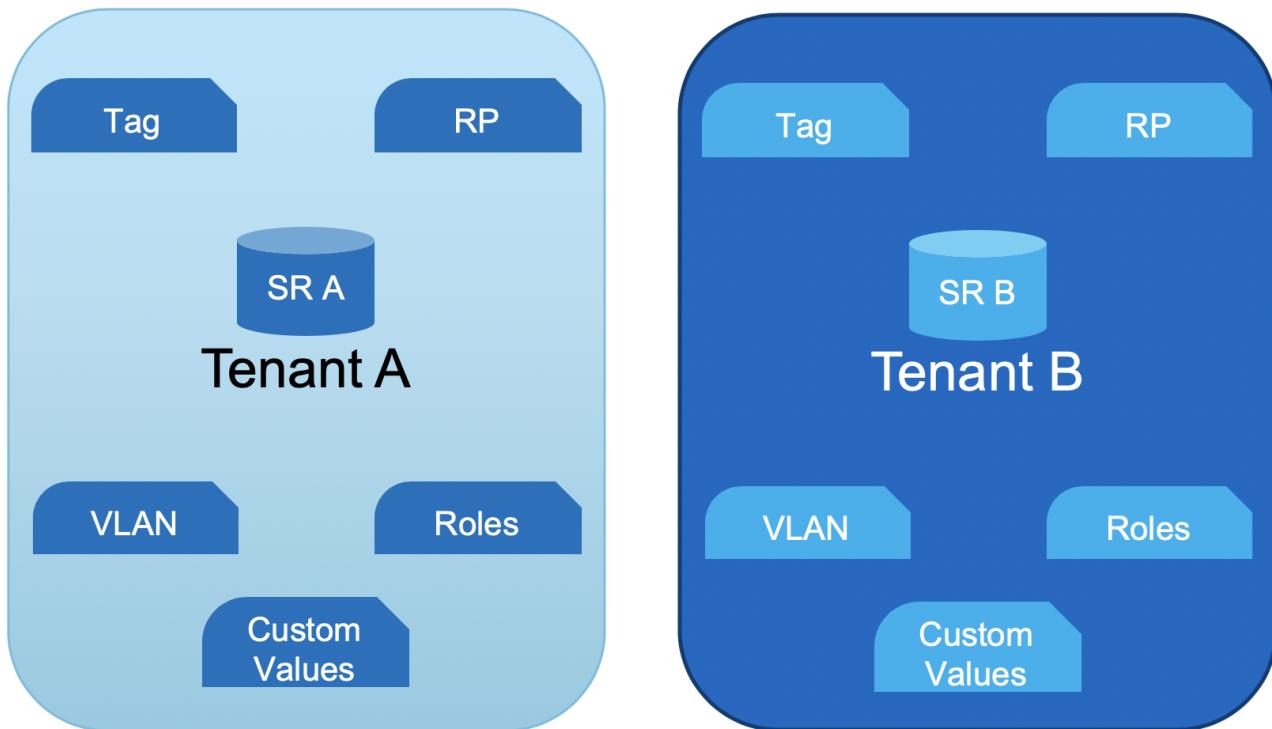
Citrix Hypervisor (formerly known as XenServer) is the industry-leading, cost-effective, open-source platform for desktop virtualization infrastructure. XenCenter is a light-weight graphical management interface for Citrix Hypervisor servers. The following figure presents an overview of the Citrix Hypervisor architecture.

## Architecture Overview



Citrix Hypervisor is a type-1 hypervisor. The control domain (also called Domain 0 or dom0) is a secure, privileged Linux VM that runs the Citrix Hypervisor management tool stack known as XAPI. This Linux VM is based on a CentOS 7.5 distribution. Besides providing Citrix Hypervisor management functions, dom0 also runs the physical device drivers for networking, storage, and so on. The control domain can talk to the hypervisor to instruct it to start or stop guest VMs.

Virtual desktops run in the guest domain, sometimes referred as the user domain or domU, and request resources from the control domain. Hardware-assisted virtualization uses CPU virtualization extensions like Intel VT. The OS kernel doesn't need to be aware that it is running on a virtual machine. Quick Emulator (QEMU) is used for virtualizing the BIOS, the IDE, the graphic adapter, USB, the network adapter, and so on. With paravirtualization (PV), the OS kernel and device drivers are optimized to boost performance in the virtual machine. The following figure presents multitenancy features of Citrix Hypervisor.



Resources from NetApp HCI makes up the hardware layer, which includes compute, storage, network, GPUs, and so on.

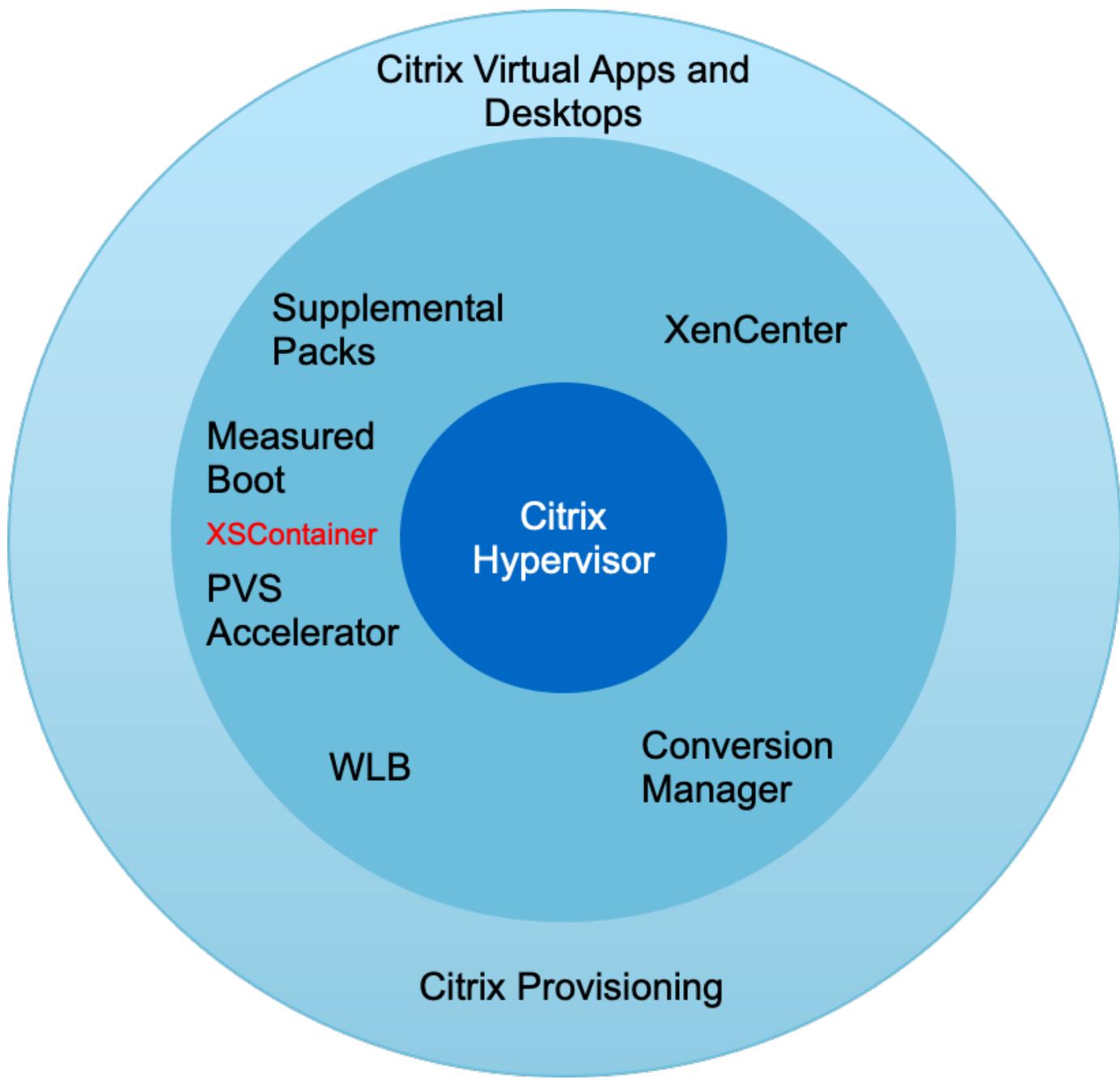
## Compute

The CPU and memory details of NetApp HCI are covered in the previous section. However, this section focuses on how the compute node is utilized in the Citrix Hypervisor environment.

Each NetApp HCI compute node with Citrix Hypervisor installed is referred as a server. A pool of servers is managed as a resource pool (RP). The resource pools are created with similar model compute nodes to provide similar performance when the workload is moved from one node to another. A resource pool always contains a node designated as master, which exposes the management interface (for XenCenter and the CLI) and which can be routed to other member servers as necessary. When high availability is enabled, master re-election takes place if the master node goes down.

A resource pool can have up to 64 servers (soft limit). However, when clustering is enabled with the GFS2 shared storage resource, the number of servers is restricted to 16.

The resource pool picks a server for hosting the workload and can be migrated to other server using the Live Migration feature. To load balance across the resource pool, the optional WLB management pack must be installed on Citrix Hypervisor.



Each tenant resource can be hosted on dedicated resource pools or can be differentiated with tags on the same resource pool. Custom values can be defined for operational and reporting purpose.

## Storage

NetApp HCI compute nodes have local storage that is not recommended for the storage of any persistent data. Such data should be stored on an iSCSI volume created with NetApp HCI storage or can be on NFS datastore on NetApp AFF.

To use NetApp HCI storage, iSCSI must be enabled on Citrix Hypervisor servers. Using the iQN, register the initiators and create access groups on the Element management portal. Create the volumes (remember to enable 512e block size support for LVM over iSCSI SR) and assign the account ID and access group.

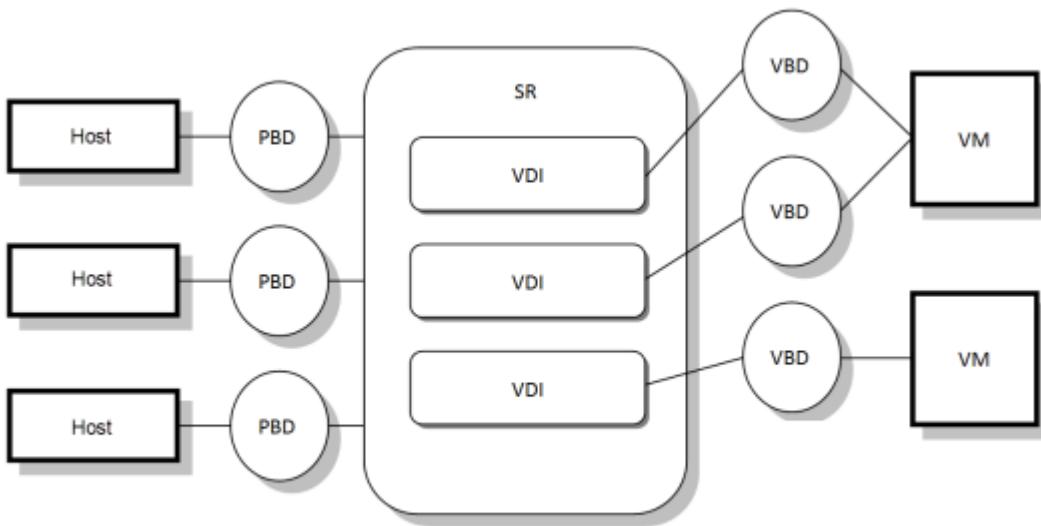


The iSCSI initiator can be customized using the following command on the CLI:

```
xe host-param-set uuid=valid_host_id other-
config:iscsi_iqn=new_initiator_iqn
```

Multipathing of iSCSI is supported when multiple iSCSI NICs are configured. iSCSI configuration is performed using XenCenter or by using CLI commands like `iscsiadm` and `multipath`. This configuration can also be performed with the various Citrix Hypervisor CLI tools. For iSCSI multipath for single target storage arrays, see [CTX138429](#).

A storage repository (SR) is the storage target in which virtual machine (VM) virtual disk images (VDIs) are stored. A VDI is a storage abstraction that represents a virtual hard disk drive (HDD). The following figure depicts various Citrix Hypervisor storage objects.



The relationship between the SR and host is handled by a physical block device (PBD), which stores the configuration information required to connect and interact with the given storage target. Similarly, a virtual block device (VBD) maintains the mapping between VDIs and a VM. Apart from that, a VBD is also used for fine tuning the quality of service (QoS) and statistics for a given VDI. The following screenshot presents Citrix Hypervisor storage repository types.

 Choose the type of new storage ?

Type	
Name	<b>Virtual disk storage</b>
Location	<input checked="" type="radio"/> iSCSI <input type="radio"/> Hardware HBA <input type="radio"/> Software FCoE
	<b>Block based storage</b> <i>iSCSI</i> iSCSI or Fibre Channel access to a shared LUN can be configured to host fully provisioned virtual disks using LVM or be formatted with the GFS2 cluster file system for hosting thinly provisioned virtual disks.
	<b>File based storage</b> <input type="radio"/> NFS <input type="radio"/> SMB/CIFS
	<b>ISO library</b> <input type="radio"/> Windows File Sharing (SMB/CIFS) <input type="radio"/> NFS ISO

**CITRIX**

[< Previous](#) [Next >](#) [Cancel](#)

With NetApp HCI, the following SR types can be created. The following table provides a comparison of features.

Feature	LVM over iSCSI	GFS2
Maximum virtual disk image size	2TiB	16TiB
Disk provisioning method	Thick Provisioned	Thin Provisioned
Read-caching support	No	Yes
Clustered pool support	No	Yes

Feature	LVM over iSCSI	GFS2
Known constraints	<ul style="list-style-type: none"> <li>• Read caching not supported</li> </ul>	<ul style="list-style-type: none"> <li>• VM migration with storage live migration is not supported for VMs whose VDIs are on a GFS2 SR. You also cannot migrate VDIs from another type of SR to a GFS2 SR.</li> <li>• Trim/unmap is not supported on GFS2 SRs.</li> <li>• Performance metrics are not available for GFS2 SRs and disks on these SRs.</li> <li>• Changed block tracking is not supported for VDIs stored on GFS2 SRs.</li> <li>• You cannot export VDIs that are greater than 2TiB as VHD or OVA/OVF. However, you can export VMs with VDIs larger than 2TiB in XVA format.</li> <li>• Clustered pools only support up to 16 hosts per pool.</li> </ul>

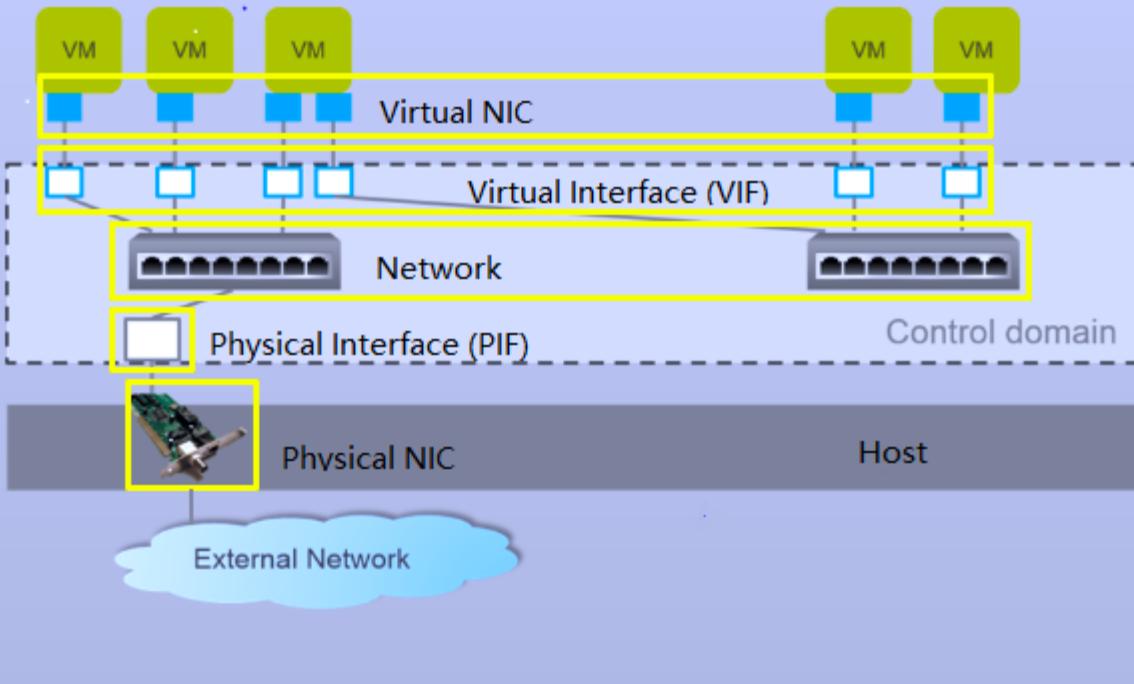
With the current features available in NetApp HCI, the Intellicache feature of Citrix Hypervisor is not of value to NetApp HCI customers. Intellicache improves performance for file-based storage systems by caching data in a local storage repository.

Read caching allows you to improve performance for certain storage repositories by caching data in server memory. GFS2 is the first iSCSI volume to support read caching.

## Network

Citrix Hypervisor networking is based on Open vSwitch with support for OpenFlow. It supports fine grain security policies to control the traffic sent and receive from a VM. It also provides detailed visibility about the behavior and performance of all traffic sent in the virtual network environment. The following figure presents an overview of Citrix Hypervisor networking.

## Networking Overview



The physical interface (PIF) is associated with a NIC on the server. With Network HCI, up to six NICs are available for use. With the model, which only has two NICs, SR-IOV can be used to add more PIFs. The PIF acts as an uplink port to the virtual switch network. The virtual interface (VIF) connects to a NIC on virtual machines.

Various network options are available:

- An external network with VLANs
- A single server private network with no external connectivity
- Bonded network (active/active – aggregate throughput)
- Bonded network (active/passive – fault tolerant)
- Bonded network (LACP – load balancing based on source and destination IP and port)
- Bonded network (LACP – load balancing based on source and destination mac address)
- Cross-server private network in which the network does not leave the resource pool
- SR-IOV

The network configuration created on the master server is replicated to other member servers. Therefore, when a new server is added to the resource pool, its network configuration is replicated from the master.



You can only assign one IP address per VLAN per NIC. For iSCSI multipath, you must have multiple PIFs to assign an IP on the same subnet. For H615C, you can consider SR-IOV for iSCSI.

New Network - NetApp-HCI-RP01

Choose the type of network to create

Select Type

Select the type of new network you would like to create:

**External Network**  
Create a network that passes traffic over one of your VLANs.

**Single-Server Private Network**  
Create a network that does not leave each server.  
This can be used as a private connection between VMs on the same host.

**Bonded Network**  
Create a network that bonds together two or more of your NICs.  
This will create a single higher performing channel.

**Cross-Server Private Network**  
Create a network that does not leave the pool.  
This can be used as a private connection between VMs in the pool.  
This type of network requires the vSwitch Controller to be running.

**SR-IOV Network**  
Enable SR-IOV on a NIC and create an SR-IOV network on that NIC.

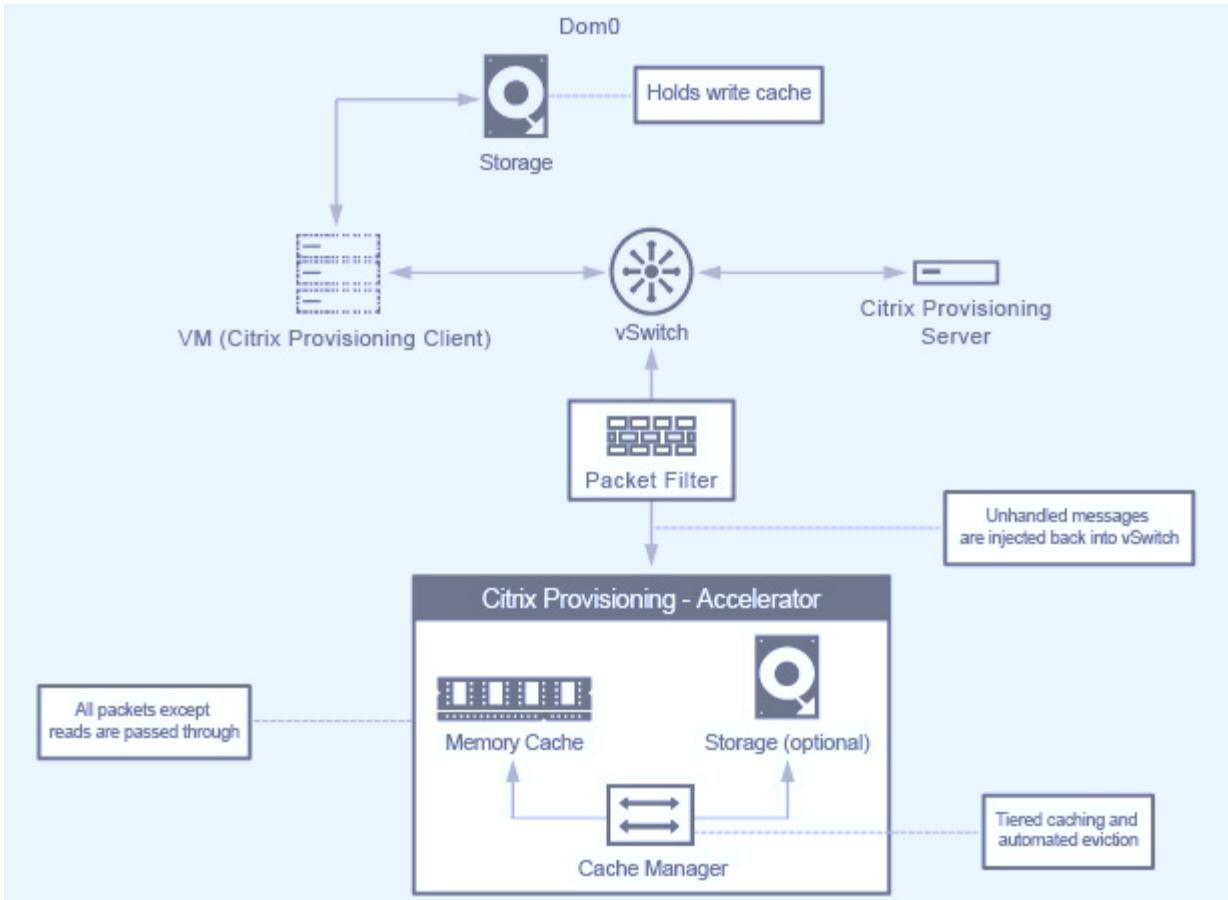
 **Cross-server private networks require the vSwitch Controller to be configured and running.**

< Previous    Next >    Cancel

**CITRIX**

Because the network on Citrix Hypervisor is based on Open vSwitch, you can manage it with ovs-vsctl and ovs-appctl commands. It also supports NVGRE/VXLAN as an overlay solution for large scale-out environments.

When used with Citrix Provisioning (PVS), PVS Accelerator improves performance by caching Domain 0 memory or by combining memory and a local storage repository.



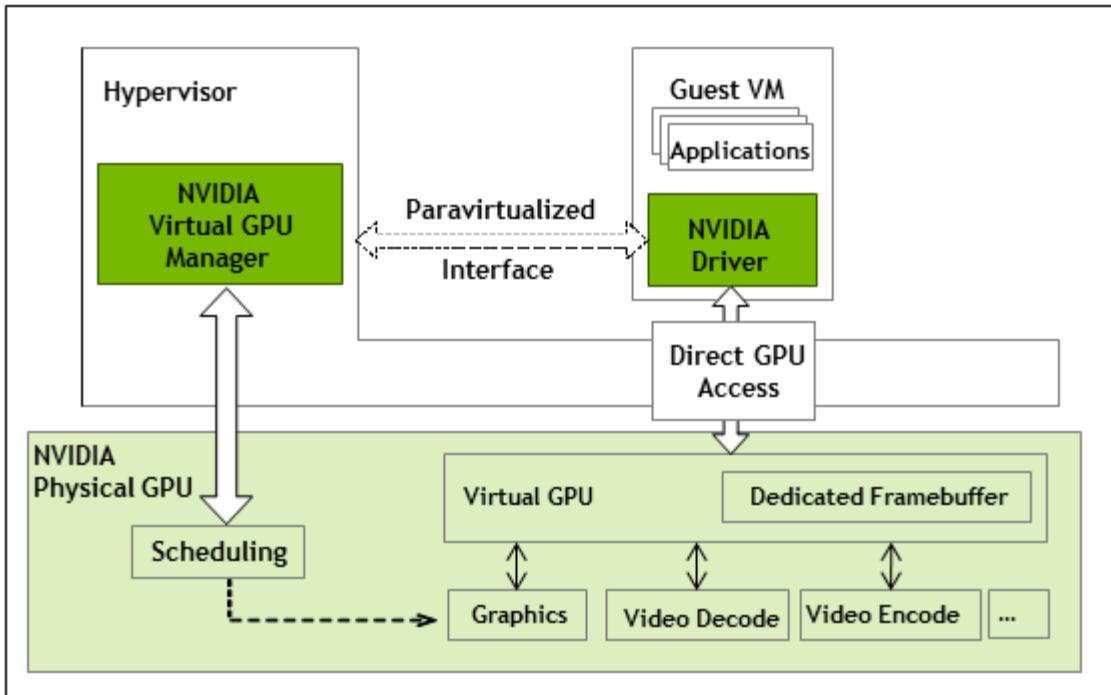
## GPU

Citrix Hypervisor was the first to deploy NVIDIA vGPUs, a virtualization platform for GPUs, enabling the sharing of GPU across multiple virtual machines. NetApp HCI H610C (with NVIDIA Tesla M10 cards) and H615C (with NVIDIA Tesla T4 cards) can provide GPU resources to virtual desktops, providing hardware acceleration to enhance the user experience.

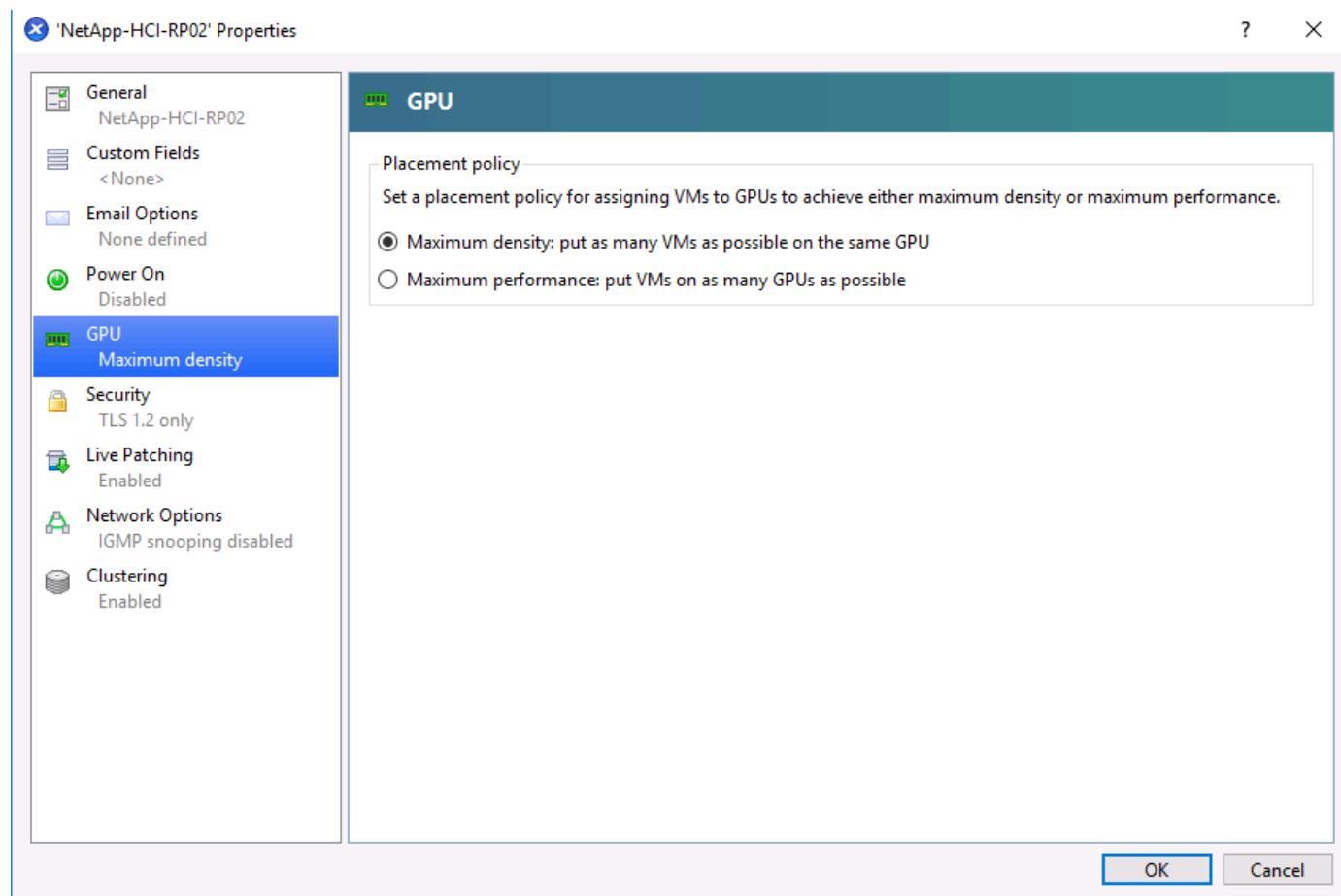
A NetApp HCI GPU can be consumed in a Citrix Hypervisor environment by using pass-through mode, where the whole GPU is presented to a single virtual machine, or it can be consumed using NVIDIA vGPU. Live migration of a VM with GPU pass through is not supported, and therefore NVIDIA vGPU is the preferred choice.

NVIDIA Virtual GPU Manager for Citrix Hypervisor can be deployed along with other management packs by using XenCenter or it can be installed using an SSH session with the server. The virtual GPU gets its own dedicated frame buffers, while sharing the streaming processors, encoder, decoder and so on. It can also be controlled using a scheduler.

The H610C has two Tesla M10 graphic cards, each with 4 GPUs per card. Each GPU has 8GB of frame buffer memory with a total of 8 GPUs and 64GB of memory per server. H615C has three Tesla T4 cards, each with its own GPU and 16GB frame buffer memory with a total of 3 GPUs and 48GB of graphic memory per server. The following figure presents an overview of the NVIDIA vGPU architecture.

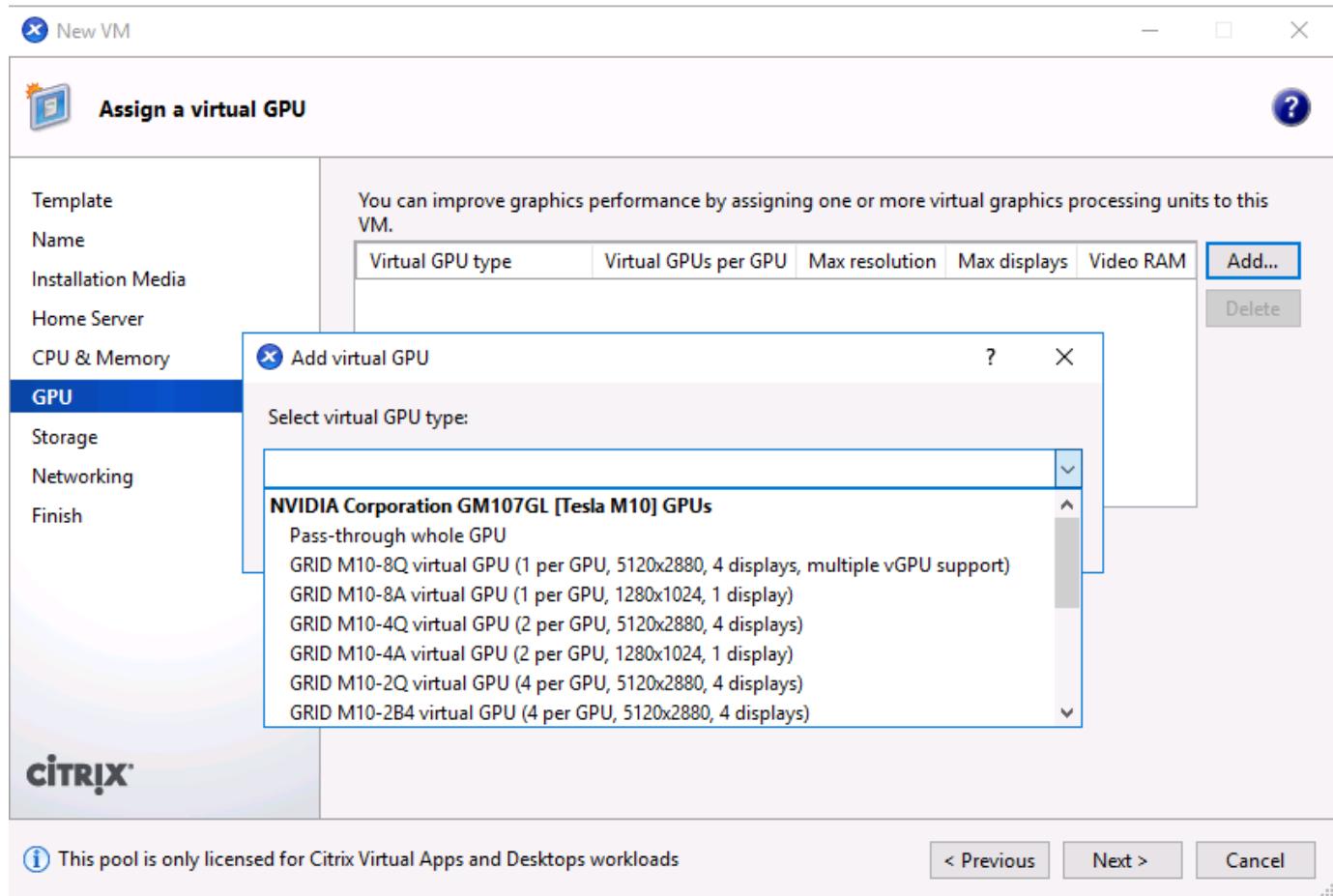


NVIDIA vGPU supports homogenous profiles for each GPU. The placement of virtual machines on a GPU is controlled by a policy that sets either maximum density or maximum performance in response to demand.



When creating a VM, you can set a virtual GPU profile. The vGPU profile you chose is based on the frame buffer memory level needed, the number of displays, and the resolution requirement. You can also set the

purpose of a virtual machine, whether it be virtual apps (A), virtual desktops (B), a professional Quadro virtual workstation (Q), or compute workloads (C) for AI inferencing applications.



Independently from XenCenter, the CLI utility on the Citrix Hypervisor nvidia-smi can be used to troubleshoot and for monitoring the performance.

The NVIDIA driver on a virtual machine is required to access the virtual GPU. Typically, the hypervisor driver version and the VM guest driver should have the same vGPU release version. But, starting with vGPU release 10, the hypervisor can have the latest version while the VM driver can be the n-1 version.

## Security

Citrix Hypervisor supports authentication, authorization, and audit controls. Authentication is controlled by local accounts as well as by Active Directory. Users and groups can be assigned to roles that control permission to resources. Events and logging can be stored remotely in addition to on the local server.

Citrix Hypervisor supports Transport Layer Security (TLS) 1.2 to encrypt the traffic using SSL certificates.

Because most configuration is stored locally in an XML database, some of the contents, like SMB passwords, are in clear text, so you must protect access to the hypervisor.

## Data Protection

Virtual machines can be exported as OVA files, which can be used to import them to other hypervisors. Virtual machines can also be exported in the native XVA format and imported to any other Citrix Hypervisor. For disaster recovery, this second option is also available along with storage-based replication handled by

SnapMirror or native Element OS synchronous or asynchronous replication. With NetApp, HCI storage can also be paired with ONTAP storage for replication.

Storage-based snapshot and cloning features are available to provide crash-consistent image backups. Hypervisor-based snapshots can be used to provide point-in-time snapshots and can also be used as templates to provision new virtual machines.

## Resource Layer

### Compute

To host virtual apps and desktop resources, a connection to a hypervisor and resource details should be configured in Citrix Studio or with PowerShell. In the case of Citrix Hypervisor, a resource pool master node DNS or IP address is required. For a secure connection, use HTTPS with SSL certificates installed on the server. Resources are defined with selection the of storage resources and networks.

[Error: Missing Graphic Image]

When additional compute capacity is required, a hypervisor server can be added to existing resource pool. Whenever you add a new resource pool and you need to make it available for hosting virtual apps and desktops, you must define a new connection.

A site is where the SQL database resides and is known as the primary zone. Additional zones are added to address users in different geographic locations to provide better response time by hosting on local resources. A satellite zone is a remote zone that only has hypervisor components to host virtual apps or desktops with optional delivery controllers.

Citrix Provisioning also uses the connection and resources information when using the Citrix Virtual Desktops Setup Wizard.

[Error: Missing Graphic Image]

### Storage

The storage repository for Virtual Apps and Desktops is controlled using the connection and resources covered in the section [Compute](#). When you define the resource, you have the option to pick the shared storage and enable Intellicache with Citrix Hypervisor.

[Error: Missing Graphic Image]

There is also an option to pick resources for the OS, the personal vDisk, and temporary data. When multiple resources are selected, Cltrix Virtual Apps and Desktops automatically spreads the load. In a multitenant environment, a dedicated resource selection can be made for each tenant resource.

[Error: Missing Graphic Image]

Citrix Provisioning requires an SMB file share to host the vDisks for the devices. We recommend hosting this SMB share on a FlexGroup volume to improve availability, performance, and capacity scaling.

[Error: Missing Graphic Image]

### FSLogix

FSLogix allows users to have a persistent experience even in non-persistent environments like pooled desktop deployment scenarios. It optimizes file I/O between the virtual desktops and the SMB file store and reduces

login time. A native (local) profile experience minimizes the tasks required on the master image to set up user profiles.

[Error: Missing Graphic Image]

FSLogix keeps user settings and personal data in its own container (VHD file). The SMB file share to store the FSLogix user profile container is configured on a registry that is controlled by group policy object. Citrix User Profile Management can be used along with FSLogix to support concurrent sessions with virtual desktops at the same time on virtual apps.

[Error: Missing Graphic Image]

This figure shows the content of the FSLogix SMB location. Note that we switched the directory name to show the username before the security identifier (sid).

## Network

Virtual Apps and Desktops require a connection and resources to host, as covered in the section [Compute](#). When defining the resource, pick the VLANs that must be associated with the resource. During machine catalog deployment, you are prompted to associate the VM NIC to the corresponding network.

[Error: Missing Graphic Image]

## GPU

As indicated in the previous section, when you determine whether the hypervisor server has a GPU resource, you are prompted to enable graphics virtualization and pick the vGPU profile.

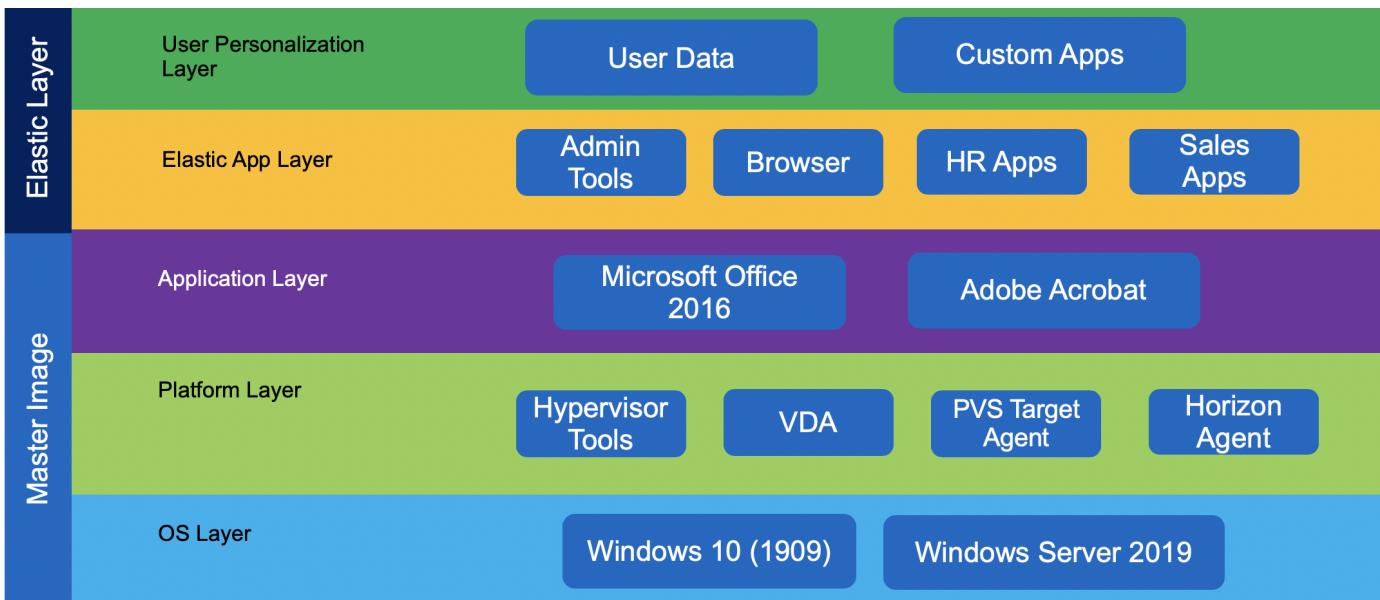
## Control Layer

### App Layering

Layering is a technology to separate the OS, applications, and user settings and data, each hosted on its own virtual disks or group of virtual disks. These components are then merged with the OS as if they were all on same machine image. Users can continue with their work without any additional training. Layers make it easy to assign, patch, and update. A layer is simply a container for file system and registry entries unique to that layer.

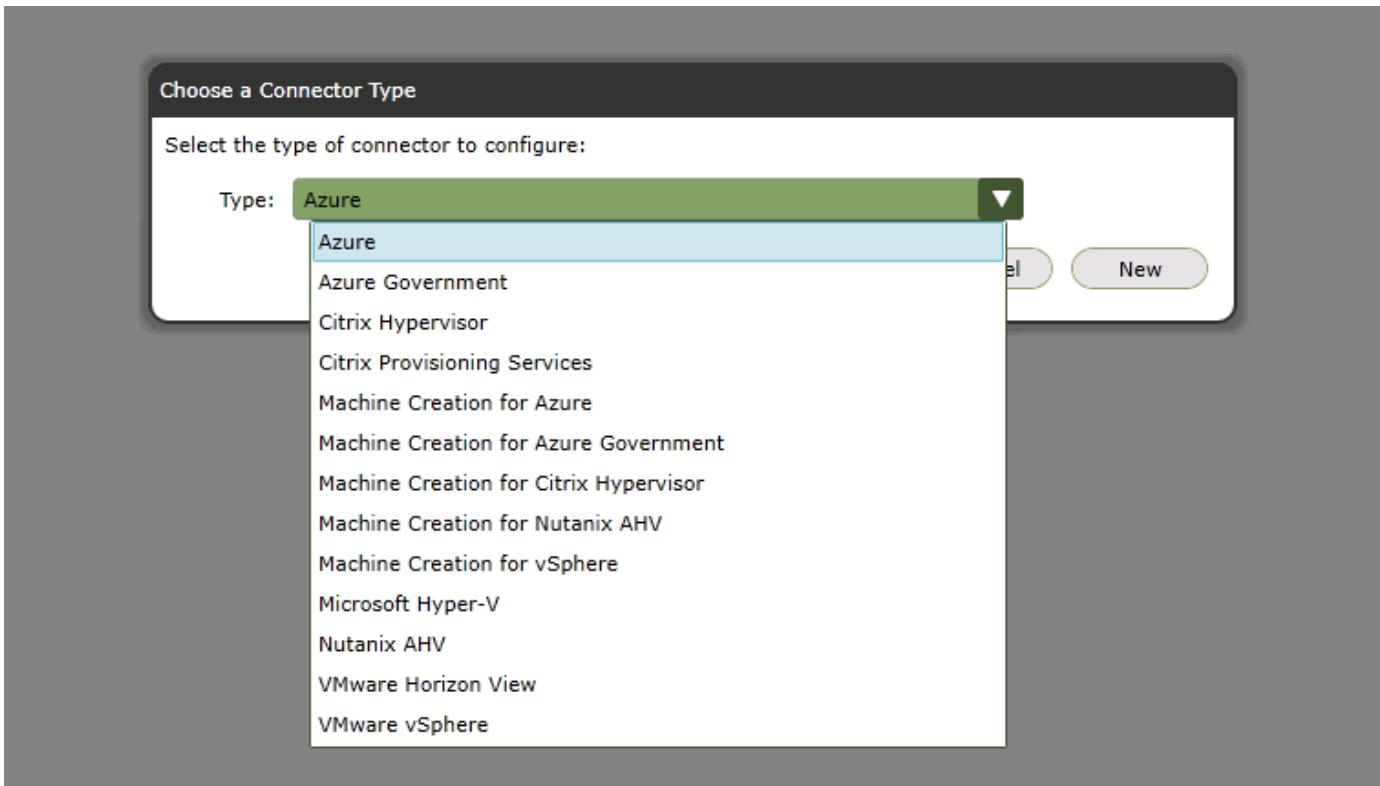
Citrix App Layering allows you to manage master images for Citrix Virtual Apps and Desktops as well as for the VMware Horizon environment. App layering also allows you to provision applications to users on demand; these apps are attached while logging in. The user personalization layer allows users to install custom apps and store the data on their dedicated layer. Therefore, you can have a personal desktop experience even when you are using a shared desktop model.

Citrix App Layering creates merged layers to create the master image and does not have any additional performance penalty. With Elastic Layers, the user login time increases.

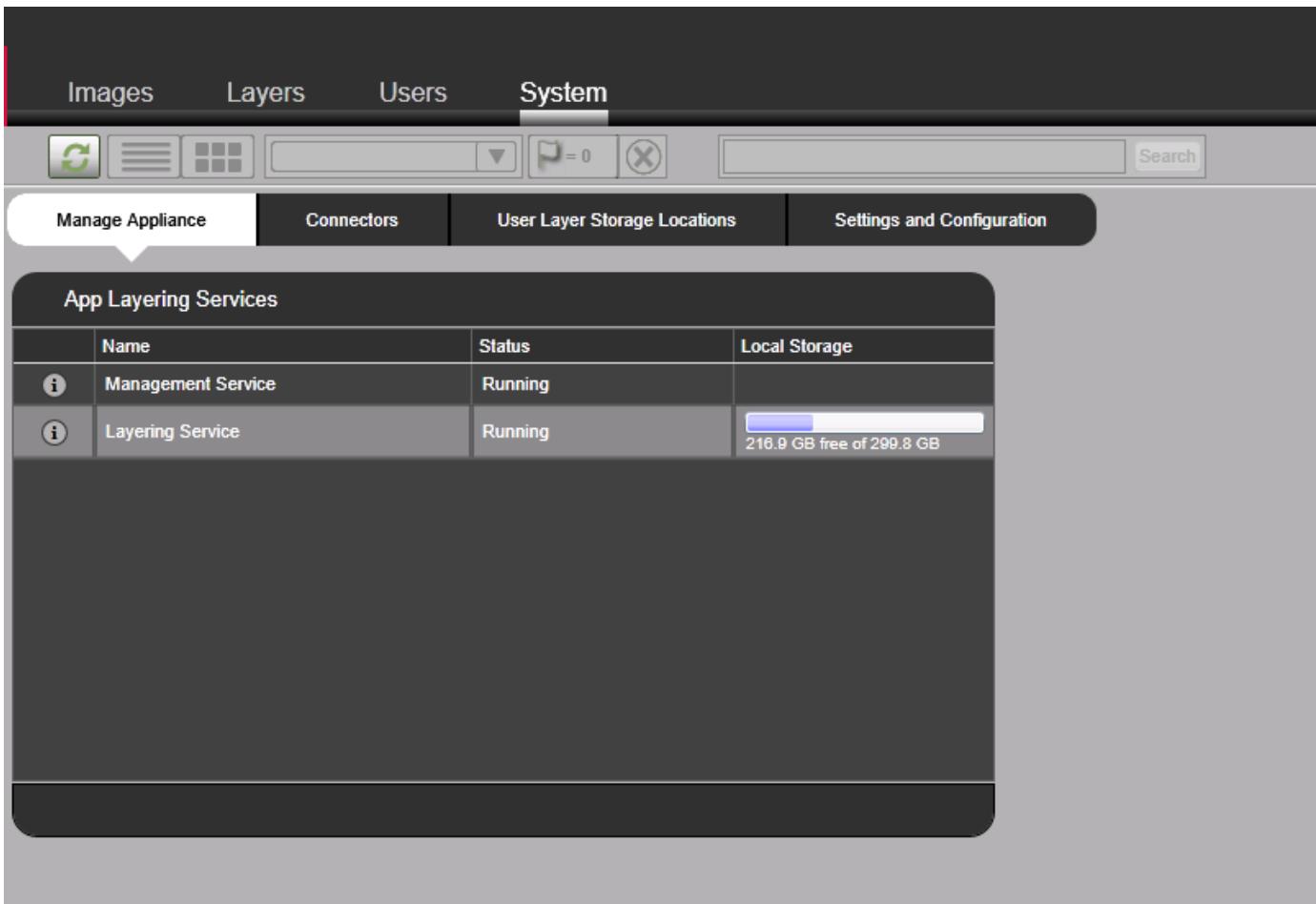


Citrix App Layering uses a single virtual appliance to manage the layers and hands off using the image and application delivery to another platform. The Citrix Enterprise Layer Manager (ELM) portal must be accessed from web browsers that supports Microsoft Silverlight 4.0. A cloud- based management portal is also available if local management interface requirements cannot be met.

Initial configuration includes the creation of platform connectors of two types; the first is a platform connector for layer creation, and the other is a platform connector for image publishing.



A layer repository is an SMB file share configured with ELM where Elastic Layers are stored. A layer work disk is where all the layers created by ELM are stored. The disk is attached to the appliance and is consumed as a block device on which a local Linux file system is used. The layer work disk is used as scratch area where the layer images are put together. After the master image is created, it is pushed to the provisioning platform.



When there are common or shared files on multiple layers, by default the high priority layer ID wins. Layer ID is incremented whenever a new layer is created. If you would like to control layer priority, use the support utility on the [Citrix LayerPriority Utility page](#).

ELM also supports authentication and role-based access control with integration with Active Directory and LDAP.

## Delivery Controller

The delivery controller is responsible for user access, brokering, and optimizing connections. It also provides Machine Creation Services (MCS) for provisioning virtual machines in an effective manner. At least one delivery controller is required per site, and typically additional controllers are added for redundancy and scalability.

Virtual desktop agents (VDA) must register with the delivery controller to make it available to users. During VDA deployment, the initial registration options can be provided manually through GPO based on the Active Directory OU. This process can also be handled with MCS.

Delivery controllers keep a local host cache in case a controller loses its connectivity to database server.

## Database

A SQL Server database is used for site configuration data, logging, and monitoring. There should be at least one database per site. To provide high availability, use Microsoft SQL Server features like AlwaysOn availability groups, database mirroring, or SQL clustering. At a minimum, consider using the hypervisor high-availability feature for a SQL VM.

Even though the controller has a local host cache, it doesn't affect any existing connections. However, for new connections, NetApp recommends database connectivity.

## Director

Citrix Director provides a monitoring solution for Citrix Virtual Apps and Desktops. Help Desk users can search for a specific user session and get a complete picture for troubleshooting. When Citrix Virtual Apps and Desktop Resources are hosted on Citrix Hypervisor, Help Desk users have the option to launch a console session from the Director portal.

The screenshot shows the Citrix Director web interface. In the top navigation bar, 'Director' and 'Durham' are selected. The main dashboard displays 'Admin Connected' and 'HR (HCIEUC-HR-04)'. Below this, the 'Activity Manager' section shows an application named 'Microsoft Edge' with a status of 'Running'. The 'Machine Details' section on the right provides detailed information about a machine named 'HCIEUCVR-04' in the 'HR' delivery group, located in the 'Durham' site. It includes fields like 'Display name', 'Delivery Group', 'Machine Catalog', 'Machine IP', 'OS type', 'Allocation Type', 'Machine IP', 'Organizational unit', 'VDA version', 'Hosting Connection Name', 'Host Name', 'VM name', 'vCPU', 'Memory', 'Hard disk', 'Avg. disk sec/transfer', 'Current disk queue length', and 'VDA software'. The 'Session Details' section shows a session for 'E13U34-HCIEUC-Demo' with details such as 'Session State', 'Application State', 'Anonymous', 'Time in state', 'Endpoint name', 'Endpoint IP', 'Connection type', 'Protocol', 'Citrix Workspace App Version', 'ICA RTT', 'ICA Latency', 'Launched via', 'Connected via', and 'Policies'.

## License

The Citrix license server manages the repository of all Citrix licenses so that licenses can be easily consumed by applications. The license server provides a management portal for advanced troubleshooting. For regular operations, Citrix Studio can also be used.

## Provisioning Services

Provisioning services enable the provisioning of desktop images even to bare metal workstations by using PXE boot. An ISO or CDROM-based boot option is also available to support environments in which network changes aren't allowed for PXE boot. The DHCP server options that we used in our lab is provided in the following figure. CP.HCIEUC.Demo and PVS.HCIEUC.Demo are the load balancer virtual IPs that point to two provisioning servers. When option 011 and 017 are available, options 066 and 067 are ignored.

The screenshot shows the Citrix DHCP configuration interface. The left sidebar shows a tree structure with 'DHCP' selected, followed by 'jumphost01.h', 'IPv4', 'Scope', and 'Scope'. The main pane displays a table of DHCP options:

Option Name	Vendor	Value	Policy Name
006 DNS Servers	Standard	172.21.146.10, 172.21.146.11	None
011 Resource Location Servers	Standard	172.21.146.67, 172.21.146.66, 172.21.146.195, 172.21.146.196	None
015 DNS Domain Name	Standard	HCIEUC.Demo	None
017 Root Path	Standard	pvs:[cp.hcieuc.demo]:17:6910	None
042 NTP Servers	Standard	10.54.17.30	None
066 Boot Server Host Name	Standard	PVS.HCIEUC.Demo	None
067 Bootfile Name	Standard	PvsNbpX64.efi	None

The high-level operation to create a machine catalog based on Citrix provisioning is as follows:

1. On the template VM, install the target agent before installing VDA.
2. Assign an additional disk for caching and format it with MBR. This step is optional. At least verify that the PVS store has a write cache path.
3. Start the Target Image Wizard and respond to its questions. Remember to provide a single Citrix Provisioning server when prompted.
4. The device boots with PXE or with ISO. The Imaging wizard continues to capture the image.
5. Select the vDisk that is created and right click to select Load Balancing and enable it.
6. For vDisk Properties, change the access mode to Standard and the Cache Type to Cache in Device RAM with Overflow on Hard Disk.
7. Right click on the site to pick the Create Virtual Desktops Setup Wizard and respond to the questions.

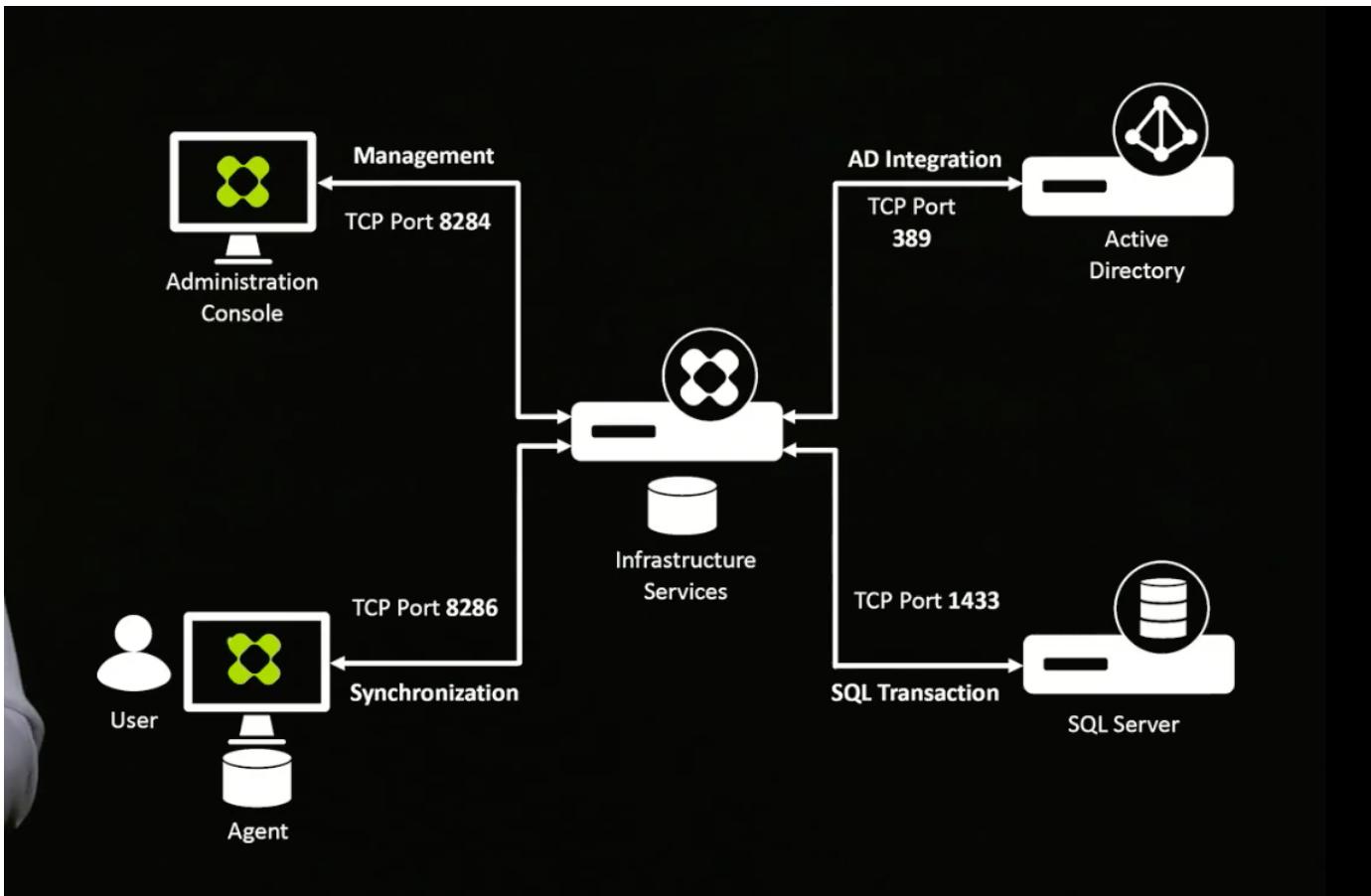
## **Studio**

Citrix Studio is the central management console used by the Citrix Virtual Apps and Desktops. The management of machine catalogs, delivery groups, applications, policies, and the configuration of resource hosting, licenses, zones, roles, and scopes are handled by the Citrix Studio. Citrix Studio also provides PowerShell snap-ins to manage Citrix Virtual Apps and Desktops.

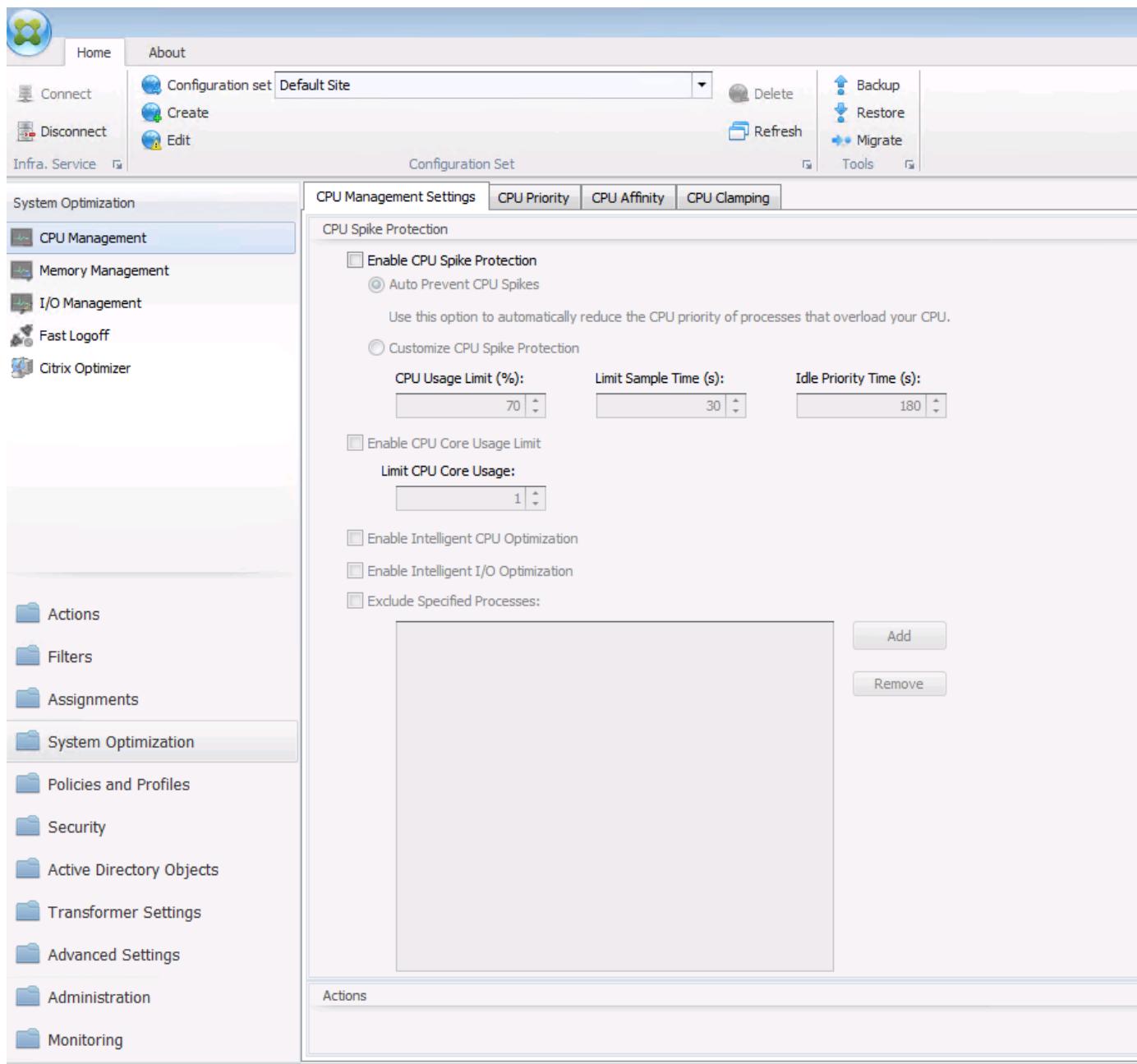
## **Workspace Environment Management**

Workspace Environment Management (WEM) provides intelligent resource management and profile management technologies to deliver the best possible performance, desktop login, and application response times for Citrix Virtual Apps and Desktops in a software-only, driver-free solution.

WEM requires a SQL database to store configuration information. To provide high availability to infrastructure services, multiple instances are used with a load balancer virtual server connection. The following figure depicts the WEM architecture.



The following figure depicts the WEM console.



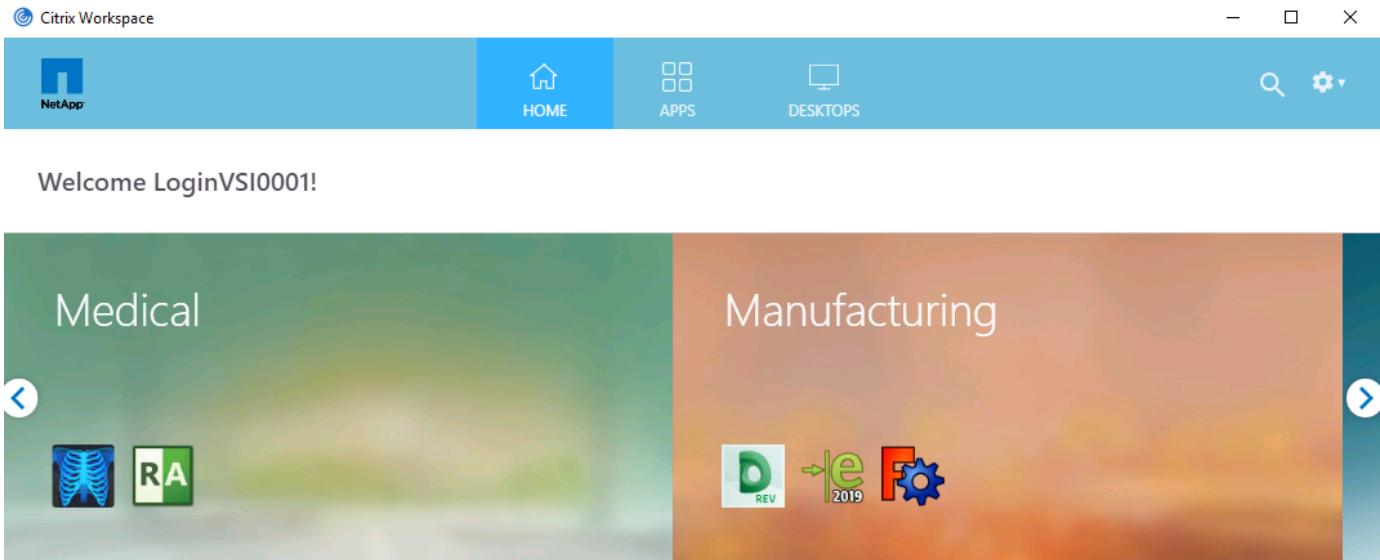
The key features of WEM are as follows:

- The ability to control resources for certain tasks or applications
- An easy interface to manage windows icons, network drives, start menu items, and so on
- The ability to reuse an old machine and manage it as a thin client
- Role-based access control
- Control policies based on various filters

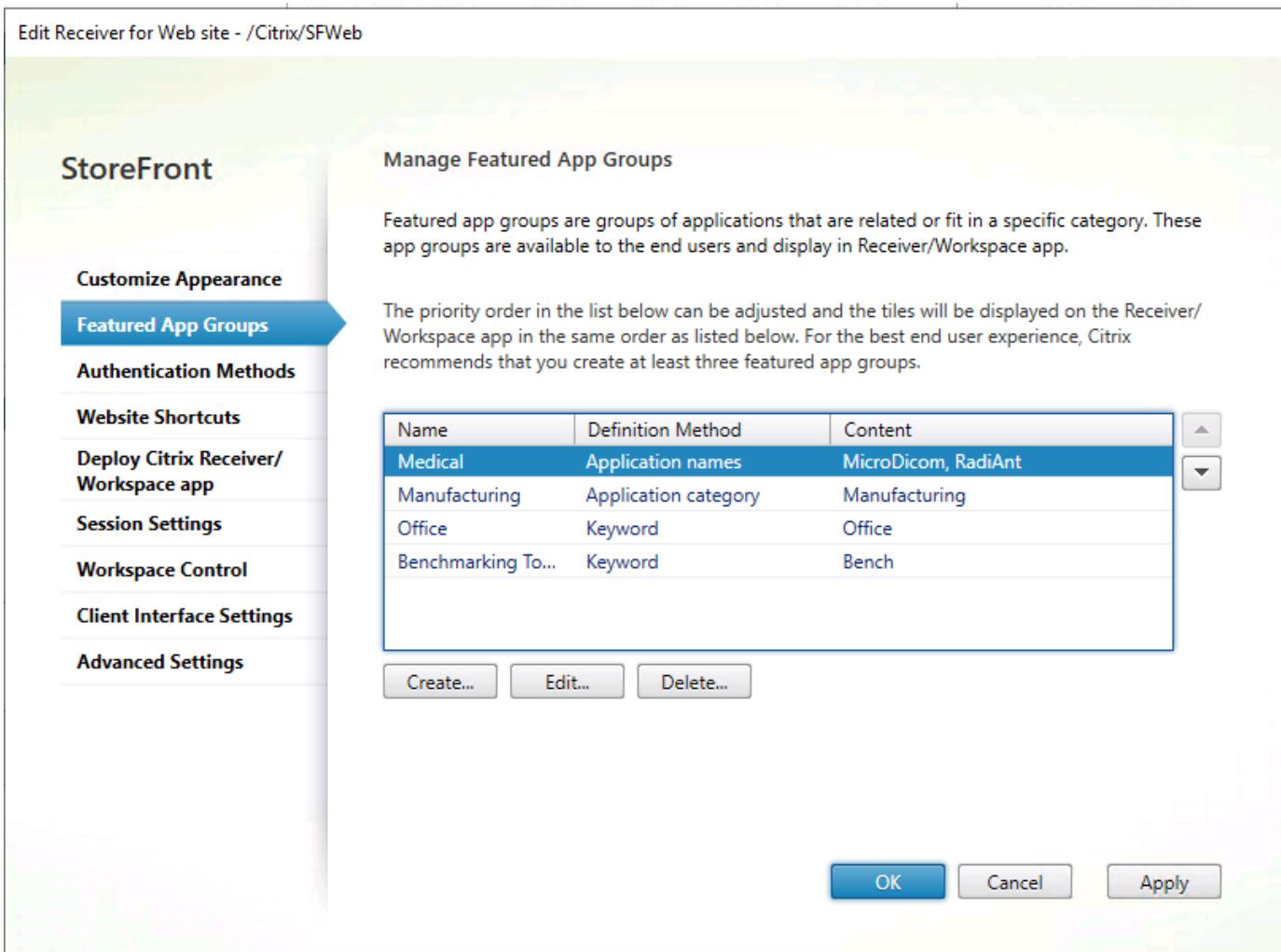
## Access Layer

### StoreFront

StoreFront consolidates resources published from multiple delivery controllers and presents unique items to users. Users connect to StoreFront and hides the infrastructure changes on the backend.



Users connect to StoreFront with the Citrix Workspace application or with a web browser. The user experience remains the same. An administrator can manage StoreFront using Microsoft Management Console. The StoreFront portal can be customized to meet customer branding demands. Applications can be grouped into categories to promote new applications. Desktops and applications can be marked as favorites for easy access. Administrators can also use tags for ease of troubleshooting and to keep track of resources in multitenant environments. The following screenshot depicts featured app groups.



## Unified Gateway

To provide secure access to Citrix Virtual Apps and Desktops from the public internet to resources hosted behind a corporate firewall, Unified Gateway is deployed in a DMZ network. Unified Gateway provides access to multiple services like an SSL VPN, a reverse proxy to intranet resources, load balancer and so on by using a single IP address or URL.

Users have the same experience whether they are accessing the resources internally or externally to an organization. Application Delivery Controller (ADC) provides enhanced networking features for Virtual Apps and Desktops, and HDX Network Insights enhances HDX monitoring information with Citrix Director.

## User Layer

Citrix Virtual Apps and Desktops enables users to access their workspace environment from anywhere with internet access and from any device with a web browser that has HTML5 support or with the Citrix Workspace application.

Users can be categorized as task workers, office workers, knowledge workers, and power users. Task workers primarily use predefined applications throughout the day for their work. Hosted Windows Apps can serve their needs. Office workers require desktop interfaces that run office applications, a web browser, and so on. Typically, they are not allowed to install applications on their workspace. They are best served by either a shared desktop with multi-session on server OS or with pooled desktops.

Knowledge workers typically require a desktop experience working with multiple applications simultaneously and must be able to persist the applications that they installed on their workspace. Static desktops (also referred to as personal desktops) allow this. Power users typically work on graphic-intensive applications or other applications that require more hardware resources. Static desktops created with an appropriate master image address the needs of power users.

## NetApp Value

### Data Fabric

Infrastructure built with the data fabric powered by NetApp allows you to migrate data or perform disaster recovery from one site to another (including the cloud). The data in Citrix Virtual Apps and Desktops can be categorized as follows:

- Infrastructure components
- Machine images
- Applications
- User profiles
- User data

Based on your needs, sites can be configured as active/active or active/passive. Infrastructure components can be on-premises or in the cloud and accessed as a service. VM templates must be distributed to each site to provision desktop and application pools. Application layers, user profiles, and data are stored in SMB file shares that must be available on each site.

You can create a global namespace using Azure NetApp Files, NetApp Cloud Volumes ONTAP, and FlexGroup volumes at the location where most of your users reside. Other locations can use Global FileCache to cache the content locally on a file server. If Citrix ShareFile is preferred, NetApp StorageGRID provides high-performance, S3-compatible storage to host data on-premises with NAS gateway access.

### Cloud Insights

Cloud Insights allows you to monitor, optimize, and troubleshoot resources deployed in the public cloud as well as on private datacenters.

Cloud Insights helps you in the following ways:

- **Reduce the mean time to resolution by as much as 90%.** Stop lengthy log hunting and failing to manually correlate infrastructure; use our dynamic topology and correlation analysis to pinpoint the problem area immediately.
- **Reduce cloud infrastructure costs by an average of 33%.** Remove inefficiencies by identifying abandoned and unused resources and right-size workloads to optimized performance and cost tiers.
- **Prevent as much as 80% of cloud issues from affecting end users.** Stop searching through vast amounts of data to find the relevant item by using advanced analytics and machine learning to identify issues before they become critical outages.

### Appendix iSCSI Device Configuration

Edit the multipath configuration file at `/etc/multipath.conf` as follows:

```

# This is a basic configuration file with some examples, for device mapper
# multipath.
## Use user friendly names, instead of using WWIDs as names.
defaults {
user_friendly_names yes
}
##
devices {
device {
vendor "SolidFir"
product "SSD SAN"
path_grouping_policy multibus path_selector "round-robin 0"
path_checker tur hardware_handler "0"
fallback immediate rr_weight uniform rr_min_io 10 rr_min_io_rq 10
features "0"
no_path_retry 24
prio const
}
}
## Device black list
## Enter devices you do NOT want to be controlled by multipathd
## Example: internal drives
#blacklist {
#}

```

## Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- [NetApp Cloud Central](#)
- [NetApp Element Software Configuration for Linux](#)
- [NetApp Product Documentation](#)
- [Citrix Security Recommendations](#)
- [Citrix Monitoring in Healthcare Environment with Goliath](#)
- [Citrix User Profile and FSLogix Integration](#)
- [Citrix App Layering Login VSI Test Results](#)
- [Citrix App Layering FAQ](#)
- [Citrix App Layering Reference Architecture](#)
- [Citrix App Layering](#)
- [Multi-session write back to FSLogix Profile Container](#)
- [Citrix XAPI Backup](#)

# **Virtual Desktop Applications**

# Containers

## Archived Solutions

### NVA-1149: NetApp HCI for Red Hat OpenShift on Red Hat Virtualization

Alan Cowles and Nikhil M Kulkarni, NetApp

NetApp HCI for Red Hat OpenShift on Red Hat Virtualization (RHV) is a best-practice deployment guide for the fully automated install of Red Hat OpenShift through the Installer Provisioned Infrastructure (IPI) method onto the verified enterprise architecture of [NVA-1148: NetApp HCI with Red Hat Virtualization](#). The purpose of this NetApp Verified Architecture deployment guide is to provide a concise set of verified instructions to be followed for the deployment of the solution. The architecture and deployment methods described in this document have been validated jointly by subject matter experts at NetApp and Red Hat to provide a best-practice implementation of the solution.

#### Use Cases

The NetApp HCI for Red Hat OpenShift on RHV solution is architected to deliver exceptional value for customers with the following use cases:

- Infrastructure to scale on demand with NetApp HCI
- Enterprise virtualized workloads in RHV
- Enterprise containerized workloads in Red Hat OpenShift

#### Business Value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure, such as the following:

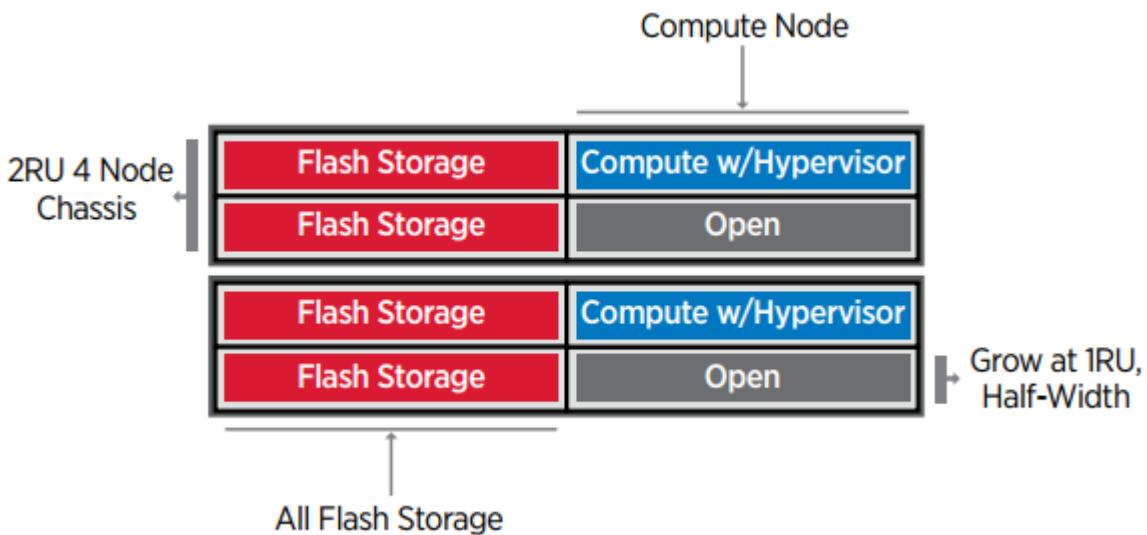
- High availability at all layers in the stack
- Ease of deployment procedures
- Nondisruptive operations and upgrades
- API-driven and programmable infrastructure to keep up with microservices agility
- Multitenancy with performance guarantees
- Ability to run virtualized and containerized workloads simultaneously
- Ability to scale infrastructure independently based on workload demands

NetApp HCI for Red Hat OpenShift on RHV acknowledges these challenges and presents a solution that helps address each concern by implementing the fully automated deployment of Red Hat OpenShift IPI on the RHV enterprise hypervisor. The remainder of this document details the components used in this verified architecture.

#### Technology Overview

## NetApp HCI

NetApp HCI is an enterprise-scale, disaggregated hybrid cloud infrastructure (HCI) solution that delivers compute and storage resources in an agile, scalable, and easy-to-manage two-rack unit (2RU), four-node building block. It can also be configured with 1RU compute and server nodes. The minimum deployment depicted in the figure below consists of four NetApp HCI storage nodes and two NetApp HCI compute nodes. The compute nodes are installed as Red Hat Virtualization Hosts (RHV-H) hypervisors in a high-availability (HA) cluster. This minimum deployment can be easily scaled to fit customer enterprise workload demands by adding additional NetApp HCI storage or compute nodes to expand available resources.



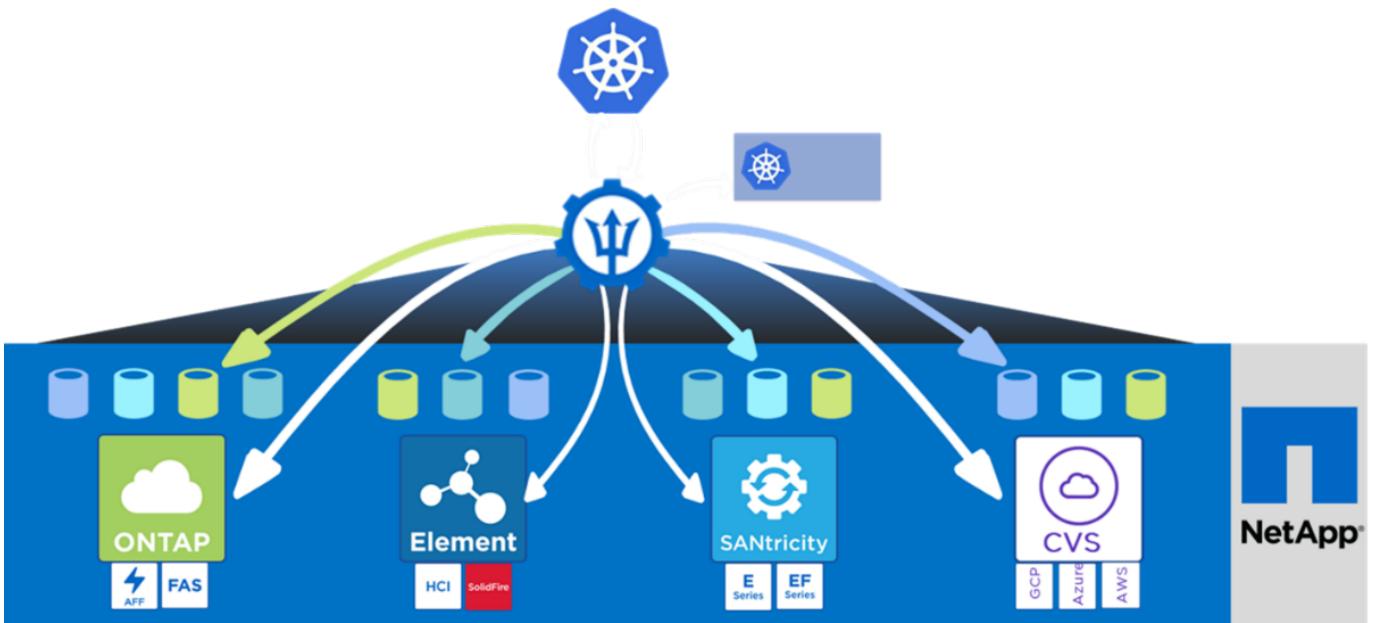
The design for NetApp HCI for Red Hat Virtualization consists of the following components in a minimum starting configuration:

- NetApp H-Series all-flash storage nodes running NetApp Element software
- NetApp H-Series compute nodes running the Red Hat Virtualization RHV-H hypervisor

For more information about compute and storage nodes in NetApp HCI, see [NetApp HCI Datasheet](#).

## NetApp Trident

Trident is a NetApp open-source and fully supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift. It works with the entire NetApp storage portfolio, including the NetApp Element storage system that is deployed as a part of the NetApp HCI solution. Trident provides the ability to accelerate the DevOps workflow by allowing end users to provision and manage storage from their NetApp storage systems, without requiring intervention from a storage administrator. An administrator can configure a number of storage backends based on project needs, and storage system models that allow for any number of advanced storage features, such as: compression, specific disk types, or QoS levels that guarantee a certain performance. After they are defined, these backends can be leveraged by developers as part of their projects to create persistent volume claims (PVCs) and attach persistent storage to their containers on demand.



## Red Hat Virtualization

RHV is an enterprise virtual data center platform that runs on Red Hat Enterprise Linux (RHEL) and uses the KVM hypervisor.

For more information about RHV, see the [Red Hat Virtualization website](#).

RHV provides the following features:

- **Centralized management of VMs and hosts.** The RHV manager runs as a physical or virtual machine (VM) in the deployment and provides a web-based GUI for the management of the solution from a central interface.
- **Self-hosted engine.** To minimize the hardware requirements, RHV allows RHV Manager (RHV-M) to be deployed as a VM on the same hosts that run guest VMs.
- **High availability.** In event of host failures, to avoid disruption, RHV allows VMs to be configured for high availability. The highly available VMs are controlled at the cluster level using resiliency policies.
- **High scalability.** A single RHV cluster can have up to 200 hypervisor hosts enabling it to support requirements of massive VMs to hold resource-greedy, enterprise-class workloads.
- **Enhanced security.** Inherited from RHV, Secure Virtualization (sVirt) and Security Enhanced Linux (SELinux) technologies are employed by RHV for the purposes of elevated security and hardening for the hosts and VMs. The key advantage from these features is logical isolation of a VM and its associated resources.

## Red Hat Virtualization Manager

RHV-M provides centralized enterprise-grade management for the physical and logical resources within the RHV virtualized environment. A web-based GUI with different role-based portals are provided to access RHV-M features.

RHV-M exposes configuration and management of RHV resources via open-source, community-driven RESTful API. It also supports full-fledged integration with Red Hat CloudForms and Red Hat Ansible for automation and orchestration.

## Red Hat Virtualization Hosts

Hosts (also called hypervisors) are the physical servers that provide hardware resources for the VMs to run on. Kernel-based Virtual Machine (KVM) provides full virtualization support, and Virtual Desktop Server Manager (VDSM) is the host agent that is responsible for communication of the hosts with the RHV-M.

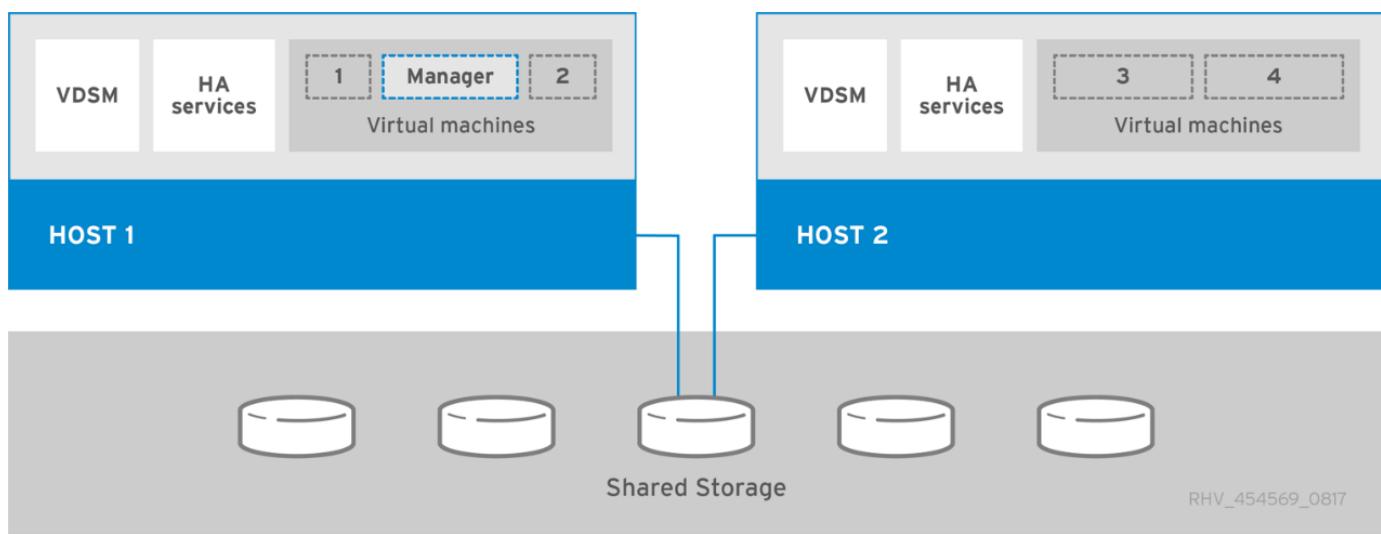
Two types of hosts are supported in RHV are RHV-H and RHEL hosts:

- RHV-H is a light-weight minimal operating system based on RHEL, optimized for ease of setting up physical servers as RHV hypervisors.
- RHEL hosts are servers that run the standard RHEL operating system and are later configured with the required subscriptions to install the packages required to permit the physical servers to be used as RHV hosts.

## Red Hat Virtualization Architecture

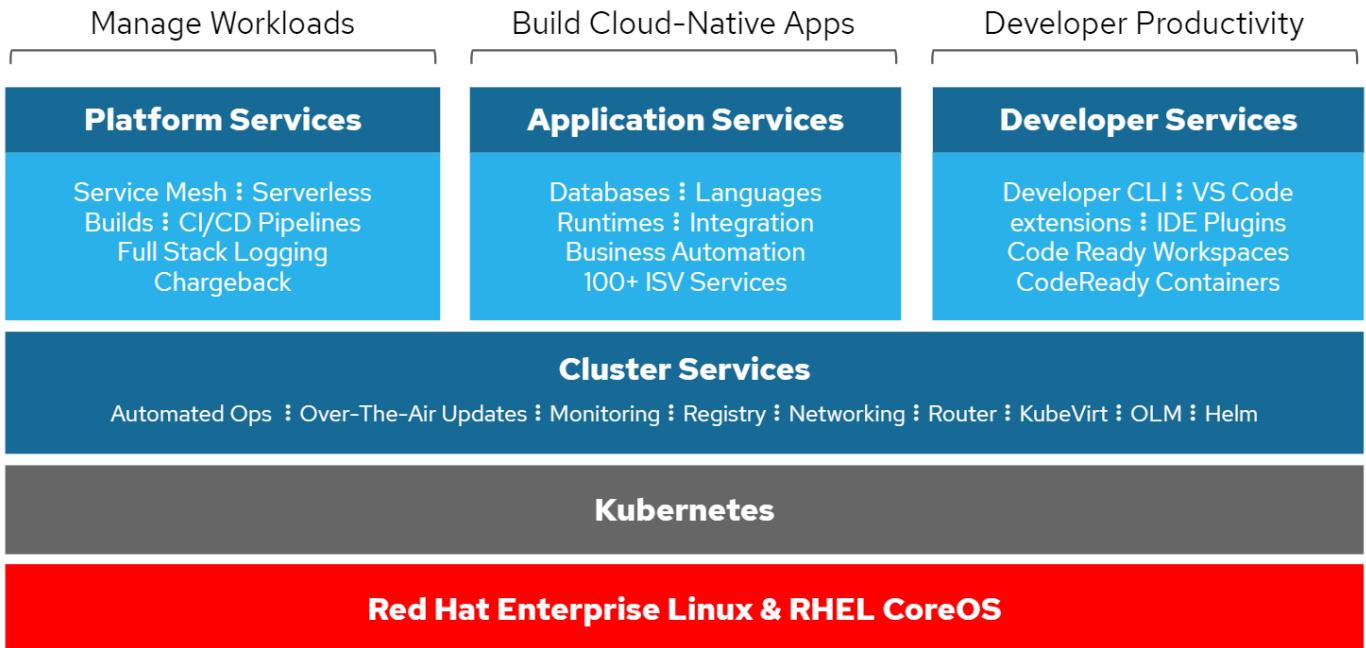
RHV can be deployed in two different architectures: with the RHV-M as a physical server in the infrastructure or with the RHV-M configured as a self-hosted engine. The self-hosted engine deployment, where the RHV-M is a VM hosted in the same environment as other VMs, is recommended and used specifically in this deployment guide.

A minimum of two self-hosted nodes are required for high availability of guest VMs and RHV-M as depicted in the figure below. For ensuring the high availability of the manager VM, HA services are enabled and run on all the self-hosted engine nodes.



## Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform is a fully supported enterprise Kubernetes platform. Red Hat makes several enhancements to open-source Kubernetes to deliver an application platform with all the components fully integrated to build, deploy, and manage containerized applications. With Red Hat OpenShift 4.4, the installation and management processes have been streamlined through the IPI method which has been deployed in this solution. By leveraging this deployment method, a fully functional OpenShift cluster providing metering and monitoring at both the cluster and application level can be fully configured and deployed on top of Red Hat Virtualization in less than an hour. OpenShift nodes are based upon RHEL CoreOS, an immutable system image designed to run containers, based on RHEL, which can be upgraded or scaled easily on demand as the needs of the end user require, helping to deliver the benefits of the public cloud to the local data center.



[Next: Architectural Overview: NetApp HCI for Red Hat OpenShift on RHV.](#)

## Abstract

This NetApp HCI for Red Hat OpenShift on Red Hat Virtualization (RHV) deployment guide is for the fully automated installation of Red Hat OpenShift through the Installer Provisioned Infrastructure (IPI) method onto the verified enterprise architecture of NetApp HCI for Red Hat Virtualization described in NVA-1148: NetApp HCI with Red Hat Virtualization. This reference document provides deployment validation of the Red Hat OpenShift solution, integration of the NetApp Trident storage orchestrator, and a solution verification consisting of an example application deployment.

## Architectural Overview: NetApp HCI for Red Hat OpenShift on RHV

### Hardware Requirements

The following table lists the minimum number of hardware components that are required to implement the solution. The hardware components that are used in specific implementations of the solution might vary based on customer requirements.

Hardware	Model	Quantity
NetApp HCI compute nodes	NetApp H410C	2
NetApp HCI storage nodes	NetApp H410S	4
Data switches	Mellanox SN2010	2
Management switches	Cisco Nexus 3048	2

## Software Requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any implementation of the solution might vary based on customer requirements.

Software	Purpose	Version
NetApp HCI	Infrastructure (compute/storage)	1.8
NetApp Element	Storage	12.0
NetApp Trident	Storage orchestration	20.04
RHV	Virtualization	4.3.9
Red Hat OpenShift	Container orchestration	4.4.6

[Next: Design Considerations: NetApp HCI for Red Hat OpenShift on RHV](#)

## Design Considerations: NetApp HCI for Red Hat OpenShift on RHV

### Network Design

The Red Hat OpenShift on RHV on HCI solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality. OCP uses the logical network on the RHV for the cluster management. This section describes the arrangement and purpose of each virtual network segment used in the solution and outlines the pre-requisites for deployment of the solution.

### VLAN Requirements

The NetApp HCI for Red Hat OpenShift on RHV solution is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). NetApp HCI requires a minimum of three network segments. However, this configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution, as well as the specific VLAN IDs that are used later in the verified architecture deployment.

VLANs	Purpose	VLAN ID
Out-of-band management network	Management for HCI nodes and IPMI	16
In-band management network	Management for HCI nodes, ovirtmgmt, and VMs	1172
Storage network	Storage network for NetApp Element	3343
Migration network	Network for virtual guest migration	3345

### Network Infrastructure Support Resources

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform (OCP) on Red Hat Virtualization on NetApp HCI solution:

- At least one DNS server which provides a full host-name resolution that is accessible from the in-band management network and the VM network.

- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.
- RHV cluster should have at least 28x vCPUs, 112GB RAM, and 840GB of available storage (depending on the production workload requirements).

[Next: Deploying NetApp HCI for Red Hat OpenShift on RHV](#)

### **Deployment Summary: NetApp HCI for Red Hat OpenShift on RHV**

The detailed steps provided in this section provide a validation for the minimum hardware and software configuration required to deploy and validate the NetApp HCI for Red Hat OpenShift on RHV solution.

Deploying Red Hat OpenShift Container Platform through IPI on Red Hat Virtualization consists of the following steps:

1. [Create storage network VLAN](#)
2. [Download OpenShift installation files](#)
3. [Download CA cert from RHV](#)
4. [Register API/Apps in DNS](#)
5. [Generate and add SSH private key](#)
6. [Install OpenShift Container Platform](#)
7. [Access console/web console](#)
8. [Configure worker nodes to run storage services](#)
9. [Download and install Trident through Operator](#)

[Next: Validation Results: NetApp HCI for Red Hat OpenShift on RHV](#)

#### **1. Create Storage Network VLAN: NetApp HCI for Red Hat OpenShift on RHV**

To create a storage network VLAN, complete the following steps:

To support Element storage access for NetApp Trident to attach persistent volumes to pods deployed in OpenShift, the machine network being used for each worker in the OCP deployment must be able to reach the storage resources. If the machine network cannot access the Element storage network by default, an additional network/VLAN can be created in the Element cluster to allow access:

1. Using any browser, log in to the Element Cluster at the cluster's MVIP.
2. Navigate to Cluster > Network and click Create VLAN.
3. Before you provide the details, reserve at least five IP addresses from the network that is reachable from the OCP network (one for the virtual network storage VIP and one for virtual network IP on each storage node).

Enter a VLAN name of your choice, enter the VLAN ID, SVIP, and netmask, select the Enable VRF option, and enter the gateway IP for the network. In the IP address blocks, enter the starting IP of the other addresses reserved for the storage nodes. In this example, the size is four because there are four storage nodes in this cluster. Click Create VLAN.

## Create a New VLAN

X

VLAN Name

VLAN Tag

SVIP

Netmask

Enable VRF

Gateway

Description

IP Address Blocks

Starting IP

Size

Add A Block

**Create VLAN**

**Cancel**

Next: 2. Download OpenShift Installation Files

### 2. Download OpenShift Installation Files: NetApp HCI for Red Hat OpenShift on RHV

To download the OpenShift installation files, complete the following steps:

1. Go to the [Red Hat login page](#) and log in with your Red Hat credentials.

2. On the Clusters page, click Create Cluster.

The screenshot shows the Red Hat OpenShift Cluster Manager web interface. At the top, there's a navigation bar with the Red Hat logo and a user profile for 'Nikhil M Kulkarni'. Below the header is a sidebar with links: Home (selected), Red Hat OpenShift Cluster Manager, Clusters (selected), Subscriptions, Documentation, Support Cases, Cluster Manager Feedback, and Red Hat Marketplace. The main content area has a large red circular 'refresh' icon. Below it, the text 'No OpenShift clusters to display' is centered. A descriptive paragraph explains the purpose of the Cluster Manager. At the bottom of the main area are three buttons: 'Create cluster' (blue), 'Register cluster' (light blue), and 'View archived clusters' (light blue).

3. Select OpenShift Container Platform.

The screenshot shows the 'Create a Cluster to Get Started' page. At the top left, there's a breadcrumb trail: 'Clusters > Create'. The main title is 'Create a Cluster to Get Started'. Below the title are two large rectangular boxes. The left box is titled 'Red Hat OpenShift Container Platform' and contains text: 'Create an OCP cluster using the command-line installer. Your cluster will automatically register to the Cluster Manager after installation completes.' The right box is titled 'Red Hat OpenShift Dedicated' and contains text: 'Create a Red Hat-managed cluster (OSD), provisioned on Amazon Web Services or Google Cloud Platform.'

4. Select Run on Red Hat Virtualization.

## Install OpenShift Container Platform 4

Select an infrastructure provider

Run on Amazon Web Services	Run on Microsoft Azure	Run on Google Cloud Platform	Run on VMware vSphere
Run on Red Hat OpenStack	Run on Red Hat Virtualization	Run on Bare Metal	IBM Z IBM LinuxONE
Run on Power	Run on Laptop Powered by Red Hat CodeReady Containers		

5. The next page allows you to download the OpenShift installer (available for Linux and MacOS), a unique pull secret that is required to create the `install-config` file and the `oc` command-line tools (available for Linux, Windows, and MacOS).

Download the files, transfer them to a RHEL administrative workstation from where you can run the OpenShift installation, or download these files directly using wget or curl on a RHEL administrative workstation.

### Downloads

**OpenShift installer**

Download and extract the install program for your operating system and place the file in the directory where you will store the installation configuration files. Note: The OpenShift install program is only available for Linux and macOS at this time.

Linux ▾ [Download installer](#)

**Pull secret**

Download or copy your pull secret. The install program will prompt you for your pull secret during installation.

[Download pull secret](#)  [Copy pull secret](#)

**Command-line interface**

Download the OpenShift command-line tools and add them to your PATH.

Linux ▾ [Download command-line tools](#)

When the installer is complete you will see the console URL and credentials for accessing your new cluster. A kubeconfig file will also be generated for you to use with the oc CLI tools you downloaded.

Next: 3. Download CA Certificate from RHV

### 3. Download CA Certificate from RHV: NetApp HCI for Red Hat OpenShift on RHV

To download the CA certificate from RHV, complete the following steps:

1. In order to access the RHV manager from the RHEL machine during the deployment process, the CA certificate trust must be updated on the machine to trust connections to RHV-M. To download the RHV Manager's CA certificate, run the following commands:

```
sudo curl -k 'https://<engine-fqdn>/ovirt-engine/services/pki-
resource?resource=ca-certificate&format=X509-PEM-CA' -o /tmp/ca.pem
[user@rhel7 ~]$ sudo curl -k 'https://rhv-m.cie.netapp.com/ovirt-
engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA'
-o /tmp/ca.pem

% Total    % Received % Xferd  Average Speed   Time     Time     Time
Current                                         Dload  Upload   Total   Spent   Left
Speed
100  1376  100  1376      0       0    9685      0  --::--  --::--  --::--
9690
```

2. Copy the CA certificate to the directory for server certificates and update the CA trust.

```
[user@rhel7 ~]$ sudo cp /tmp/ca.pem /etc/pki/ca-
trust/source/anchors/ca.pem
[user@rhel7 ~]$ sudo update-ca-trust
```

Next: [4. Register API/Apps in DNS](#)

#### 4. Register API/Apps in DNS: NetApp HCI for Red Hat OpenShift on RHV

To register API/Apps in DNS, complete the following steps:

1. Reserve three static IP addresses from the network being used for OCP: the first IP address for OpenShift Container Platform REST API, the second IP address for pointing to the wildcard application ingress, and the third IP address for the internal DNS service. The first two IPs require an entry in the DNS server.



The default value of the `machineNetwork` subnet as created by IPI during OpenShift install is `10.0.0.0/16`. If the IPs you intend to use for your cluster's management network fall outside of this range, you might need to customize your deployment and edit these values before deploying the cluster. For more information, see the section [Use a Custom Install File for OpenShift Deployment](#).

2. Configure the API domain name by using the format `api.<openshift-cluster-name>.<base-domain>` pointing to the reserved IP.

## New Host

X

Name (uses parent domain name if blank):

Fully qualified domain name (FQDN):

IP address:

Create associated pointer (PTR) record

Allow any authenticated user to update DNS records with the same owner name

Add Host

Cancel

3. Configure the wildcard application ingress domain name by using the format `*.apps.<openshift-cluster-name>.<base-domain>` pointing to the reserved IP.

## New Host

Name (uses parent domain name if blank):  
\*.apps.rhv-ocp-cluster

Fully qualified domain name (FQDN):  
\*.apps.rhv-ocp-cluster.cie.netapp.com.

IP address:  
10.63.172.152

Create associated pointer (PTR) record  
 Allow any authenticated user to update DNS records with the same owner name

Add Host      Cancel

Next: 5. Generate and Add SSH Private Key

### 5. Generate and Add SSH Private Key: NetApp HCI for Red Hat OpenShift on RHV

To generate and add an SSH private key, complete the following steps:

1. For the installation debugging or disaster recovery on the OpenShift cluster, you must provide an SSH key to both the `ssh-agent` and the installation program. Create an SSH key if one does not already exist for password-less authentication on the RHEL machine.

```
[user@rhel7 ~]$ ssh-keygen -t rsa -b 4096 -N '' -f ~/.ssh/id_rsa
```

2. Start the `ssh-agent` process and configure it as a background running task.

```
[user@rhel7 ~]$ eval "$(ssh-agent -s)"  
Agent pid 31874
```

3. Add the SSH private key that you created in step 2 to the `ssh-agent`, which enables you to SSH directly

to the nodes without having to interactively pass the key.

```
[user@rhel7 ~]$ ssh-add ~/.ssh/id_rsa
```

## Next: 6. Install OpenShift Container Platform

### 6. Install OpenShift Container Platform: NetApp HCI for Red Hat OpenShift on RHV

To install OpenShift Container Platform, complete the following steps:

1. Create a directory for OpenShift installation and transfer the downloaded files to it. Extract the OpenShift installer files from the tar archive.

```
[user@rhel7 ~]$ mkdir openshift-deploy
[user@rhel7 ~]$ cd openshift-deploy
[user@rhel7 openshift-deploy]$ tar xvf openshift-install-linux.tar.gz
README.md
openshift-install
[user@rhel7 openshift-deploy]$ ls -la
total 453260
drwxr-xr-x.  2 user user      146 May 26 16:01 .
dr-xr-x---. 16 user user     4096 May 26 15:58 ..
-rw-r--r--.  1 user user  25249648 May 26 15:59 openshift-client-
linux.tar.gz
-rwxr-xr-x.  1 user user 354664448 Apr 27 01:37 openshift-install
-rw-r--r--.  1 user user  84207215 May 26 16:00 openshift-install-
linux.tar.gz
-rw-r--r--.  1 user user     2736 May 26 15:59 pull-secret.txt
-rw-r--r--.  1 user user      706 Apr 27 01:37 README.md
```



The installation program creates several files in the directory used for installation of the cluster. Both the installation program and the files created by the installation program must be kept even after the cluster is up.



The binary files that you previously downloaded, such as `openshift-install` or `oc`, can be copied to a directory that is in the user's path (for example, `/usr/local/bin`) to make them easier to run.

2. Create the cluster by running the `openshift-install create cluster` command and respond to the installation program prompts. Pass the SSH public key, select ovirt from the platform, provide the RHV infrastructure details, provide the three reserved IP addresses and the downloaded pull secret to the installation program prompts. After all the inputs are provided, the installation program creates and configures a bootstrap machine with a temporary Kubernetes control plane which then creates and configures the master VMs with the production Kubernetes control plane. The control plane on the master nodes creates and configures the worker VMs.

It can take approximately 30–45 minutes to get the complete cluster up and running.

```
[user@rhel7 openshift-deploy]$ ./openshift-install create cluster
--dir=/home/user/openshift-deploy --log-level=info ??
SSH Public Key /home/user/.ssh/id_rsa.pub
? Platform ovirt
? oVirt cluster Default
? oVirt storage domain data_domain
? oVirt network ovirtmgmt
? Internal API virtual IP 10.63. 172.151
? Internal DNS virtual IP 10.63. 172.153
? Ingress virtual IP 10.63. 172.152
? Base Domain cie.netapp.com
? Cluster Name rhv-ocp-cluster
? Pull Secret [? for help]
*****
*****  

*****  

*****  

*****  

*****  

*****  

*****  

*****  

*****  

*****  

*****  

*****  

INFO Obtaining RHCOS image file from 'https://releases-art-
rhcose.svc.ci.openshift.org/art/storage/releases/rhcos-
4.4/44.81.202004250133-0/x86_64/rhcos-44.81.202004250133-0-
openstack.x86_64.qcow2.gz?sha256=f8a44e0ea8cc45882dc22eb632a63afb90b4148
39b8aa92f3836ede001dfe9cf'
INFO The file was found in cache: /home/user/.cache/openshift-
installer/image_cache/e263efbc53c0caf612bcfaad10e3dff0. Reusing...
INFO Creating infrastructure resources...
INFO Waiting up to 20m0s for the Kubernetes API at https://api.rhv-ocp-
cluster.cie.netapp.com:6443...
INFO API v1.17.1 up
INFO Waiting up to 40m0s for bootstrapping to complete...
INFO Destroying the bootstrap resources...
INFO Waiting up to 30m0s for the cluster at https://api.rhv-ocp-
cluster.cie.netapp.com:6443 to initialize...
INFO Waiting up to 10m0s for the openshift-console route to be
created...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run
'export KUBECONFIG=/home/user/openshift-deploy/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.rhv-ocp-cluster.cie.netapp.com
INFO Login to the console with user: kubeadmin, password: NtsqU-p3qUb-
8Hscu-JfAq7
```

3. When the cluster deployment is complete, the directions for accessing the OpenShift cluster, including a link to its web console and credentials for the kubeadmin user, are displayed. Make sure to take a note of

these details.

4. Log in to the RHV Manager and observe that the VMs relating to the OCP cluster are up and running.

Name	Host	IP Addresses	FQDN	Cluster	Data Center	Memory	CPU	Network	Graphics	Status	Uptime
HostedEngine	rhv-h02.cie.netapp.o	10.63.172.150 feb...	rhv-m.cie.netapp....	Default	Default	30%	15%	0% SPICE + ...	Up	5 day	
NetApp-mNode	rhv-h02.cie.netapp.o			Default	Default	24%	2%	0% SPICE + ...	Up	25 mi	
rhv-ocp-cluster-hdr7k-master-0	rhv-h01.cie.netapp.o			Default	Default	69%	53%	0% SPICE + ...	Up	1 h	
rhv-ocp-cluster-hdr7k-master-1	rhv-h02.cie.netapp.o			Default	Default	50%	35%	0% SPICE + ...	Up	1 h	
rhv-ocp-cluster-hdr7k-master-2	rhv-h01.cie.netapp.o			Default	Default	59%	51%	0% SPICE + ...	Up	1 h	
rhv-ocp-cluster-hdr7k-worker-0-ghszk	rhv-h02.cie.netapp.o			Default	Default	16%	16%	0% SPICE + ...	Up	1 h	
rhv-ocp-cluster-hdr7k-worker-0-xdl99	rhv-h01.cie.netapp.o			Default	Default	14%	12%	0% SPICE + ...	Up	1 h	
rhv-ocp-cluster-hdr7k-worker-0-zkomt	rhv-h02.cie.netapp.o			Default	Default	15%	14%	0% SPICE + ...	Up	1 h	
tmpvm-for-rhv-ocp-cluster-hdr7k-rhcos				Default	Default	--	--	--	None	Down	

Next: [7. Access Console/Web Console](#)

## 7. Access Console/Web Console: NetApp HCI for Red Hat OpenShift on RHV

To access the console or web console, complete the following steps:

1. To access the OCP cluster through the CLI, extract the `oc` command-line tools tar file and place its content in a directory that is in the user's path.

```
[user@rhel7 openshift-deploy]$ tar xvf openshift-client-linux.tar.gz  
README.md  
oc  
kubectl  
[user@rhel7 openshift-deploy]$ echo $PATH  
/usr/local/bin: /usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin  
  
[user@rhel7 openshift-deploy]$ cp oc /usr/local/bin
```

2. To interact with the cluster through the CLI, you can use the `kubeconfig` file provided by the IPI process located in the `/auth` directory inside the folder from where you launched the installation program. To easily interact with the cluster, export the file that is created in the directory. After a successful cluster deployment, the file location and the following command are displayed.

```
[user@rhel7 openshift-deploy]$ export KUBECONFIG=/home/user/openshift-deploy/auth/kubeconfig
```

3. Verify whether you have access to the cluster and whether the nodes are in the Ready state.

```
[user@rhel7 openshift-deploy]$ oc get nodes
NAME                               STATUS  ROLES     AGE   VERSION
rhv-ocp-cluster-hdr7k-master-0    Ready   master   93m   v1.17.1
rhv-ocp-cluster-hdr7k-master-1    Ready   master   93m   v1.17.1
rhv-ocp-cluster-hdr7k-master-2    Ready   master   93m   v1.17.1
rhv-ocp-cluster-hdr7k-worker-0-ghskz Ready   worker   83m   v1.17.1
rhv-ocp-cluster-hdr7k-worker-0-xdl199 Ready   worker   86m   v1.17.1
rhv-ocp-cluster-hdr7k-worker-0-zkxmt Ready   worker   85m   v1.17.1
```

4. Log in to the web console URL by using the credentials, both of which were provided after the successful deployment of the cluster, and then verify GUI access to the cluster.

The screenshot shows the Red Hat OpenShift Container Platform web interface. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and a user dropdown 'kube:admin'. The left sidebar has a 'Administrator' section with links for Home, Operators, Workloads, Networking, and Storage. The main content area is titled 'Overview' under 'Cluster'. It features a 'Status' section with three green checkmarks: 'Cluster', 'Control Plane', and 'Operators'. A blue banner at the top states: 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.'

[Next: 8. Configure Worker Nodes to Run Storage Services](#)

## 8. Configure Worker Nodes to Run Storage Services: NetApp HCI for Red Hat OpenShift on RHV

To configure the worker nodes to run storage services, complete the following steps:

1. To access storage from the Element system, each of the worker nodes must have iSCSI available and running as a service. To create a machine configuration that can enable and start the `iscisd` service, log in to the OCP web console and navigate to Compute > Machine Configs and click Create Machine Config. Paste the YAML file and click Create.

# Create Machine Config

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

[View shortcuts](#)

```
1  apiVersion: machineconfiguration.openshift.io/v1
2  kind: MachineConfig
3  metadata:
4    labels:
5      | machineconfiguration.openshift.io/role: worker
6      name: worker-iscsi-configuration
7  spec:
8    config:
9      ignition:
10     version: 2.2.0
11     systemd:
12       units:
13         - name: iscsid.service
14           enabled: true
15           state: started
16   osImageURL: ""|
```

[Create](#)

[Cancel](#)

[!\[\]\(e4b890b84880c7fcea8a596771aac717\_img.jpg\) Download](#)

2. After the configuration is created, it will take approximately 20–30 minutes to apply the configuration to the worker nodes and reload them. Verify whether the machine config is applied by using `oc get mcp` and make sure that the machine config pool for workers is updated. You can also log in to the worker nodes to confirm that the iscsid service is running.

```
[user@rhel7 openshift-deploy]$ oc get mcp
NAME      CONFIG                                     UPDATED     UPDATING
DEGRADED
master    rendered-master-a520ae930e1d135e0dee7168   True       False
False
worker    rendered-worker-de321b36eeba62df41feb7bc   True       False
False
[user@rhel7 openshift-deploy]$ ssh core@10.63.172.22 sudo systemctl
status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
   vendor preset: disabled)
     Active: active (running) since Tue 2020-05-26 13:36:22 UTC; 3 min ago
       Docs: man:iscsid(8)
              man:iscsiadm(8)
   Main PID: 1242 (iscsid)
     Status: "Ready to process requests"
      Tasks: 1
     Memory: 4.9M
        CPU: 9ms
      CGroup: /system.slice/iscsid.service
              └─1242 /usr/sbin/iscsid -f
```



It is also possible to confirm that the MachineConfig has been successfully applied and services have been started as expected by running the `oc debug` command with the appropriate flags.

[Next: 9. Download and Install NetApp Trident](#)

## 9. Download and Install NetApp Trident: NetApp HCI for Red Hat OpenShift on RHV

To download and install NetApp Trident, complete the following steps:

1. Make sure that the user that is logged in to the OCP cluster has sufficient privileges for installing Trident.

```
[user@rhel7 openshift-deploy]$ oc auth can-i '*' '*' --all-namespaces
yes
```

2. Verify that you can download an image from the registry and access the MVIP of the NetApp Element cluster.

```
[user@rhel7 openshift-deploy]$ oc run -i --tty ping --image=busybox  
--restart=Never --rm -- ping 10.63.172.140  
If you don't see a command prompt, try pressing enter.  
64 bytes from 10.63.172.140: seq=1 ttl=63 time=0.312 ms  
64 bytes from 10.63.172.140: seq=2 ttl=63 time=0.271 ms  
64 bytes from 10.63.172.140: seq=3 ttl=63 time=0.254 ms  
64 bytes from 10.63.172.140: seq=4 ttl=63 time=0.309 ms  
64 bytes from 10.63.172.140: seq=5 ttl=63 time=0.319 ms  
64 bytes from 10.63.172.140: seq=6 ttl=63 time=0.303 ms  
^C  
--- 10.63.172.140 ping statistics ---  
7 packets transmitted, 7 packets received, 0% packet loss  
round-trip min/avg/max = 0.254/0.387/0.946 ms  
pod "ping" deleted
```

3. Download the Trident installer bundle using the following commands and extract it to a directory.

```
[user@rhel7 ~]$ wget  
[user@rhel7 ~]$ tar -xf trident-installer-20.04.0.tar.gz  
[user@rhel7 ~]$ cd trident-installer
```

4. The Trident installer contains manifests for defining all the required resources. Using the appropriate manifests, create the TridentProvisioner custom resource definition.

```
[user@rhel7 trident-installer]$ oc create -f  
deploy/crds/trident.netapp.io_tridentprovisioners_crd_post1.16.yaml  
  
customresourcedefinition.apiextensions.k8s.io/tridentprovisioners.triden  
t.netapp.io created
```

5. Create a Trident namespace, which is required for the Trident operator.

```
[user@rhel7 trident-installer]$ oc create namespace trident  
namespace/trident created
```

6. Create the resources required for the Trident operator deployment, such as a ServiceAccount for the operator, a ClusterRole and ClusterRoleBinding to the ServiceAccount, a dedicated PodSecurityPolicy, or the operator itself.

```
[user@rhel7 trident-installer]$ oc kustomize deploy/ >
deploy/bundle.yaml
[user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

7. Verify that the Trident operator is deployed.

```
[user@rhel7 trident-installer]$ oc get deployment -n trident
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator   1/1      1          1          56s
[user@rhel7 trident-installer]$ oc get pods -n trident
NAME                           READY   STATUS    RESTARTS   AGE
trident-operator-564d7d66f-qrz7v   1/1     Running   0          71s
```

8. After the Trident operator is installed, install Trident using this operator. In this example, TridentProvisioner custom resource (CR) was created. The Trident installer comes with definitions for creating a TridentProvisioner CR. These can be modified based on the requirements.

```
[user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentprovisioner_cr.yaml
tridentprovisioner.trident.netapp.io/trident created
```

9. Approve the Trident serving CSR certificates by using `oc get csr -o name | xargs oc adm certificate approve`.

```
[user@rhel7 trident-installer]$ oc get csr -o name | xargs oc adm  
certificate approve  
certificatesigningrequest.certificates.k8s.io/csr-4b7zh approved  
certificatesigningrequest.certificates.k8s.io/csr-4hkwc approved  
certificatesigningrequest.certificates.k8s.io/csr-5bgh5 approved  
certificatesigningrequest.certificates.k8s.io/csr-5g4d6 approved  
certificatesigningrequest.certificates.k8s.io/csr-5j9hz approved  
certificatesigningrequest.certificates.k8s.io/csr-5m8qb approved  
certificatesigningrequest.certificates.k8s.io/csr-66hv2 approved  
certificatesigningrequest.certificates.k8s.io/csr-6rdgg approved  
certificatesigningrequest.certificates.k8s.io/csr-6t24f approved  
certificatesigningrequest.certificates.k8s.io/csr-76wgv approved  
certificatesigningrequest.certificates.k8s.io/csr-78qsq approved  
certificatesigningrequest.certificates.k8s.io/csr-7r58n approved  
certificatesigningrequest.certificates.k8s.io/csr-8ghmk approved  
certificatesigningrequest.certificates.k8s.io/csr-8sn5q approved
```

10. Verify that Trident 20.04 is installed by using the TridentProvisioner CR, and verify that the pods related to Trident are.

```
[user@rhel7 trident-installer]$ oc get tprov -n trident  
NAME      AGE  
trident   9m49s  
  
[user@rhel7 trident-installer]$ oc describe tprov trident -n trident  
Name:          trident  
Namespace:     trident  
Labels:        <none>  
Annotations:   <none>  
API Version:  trident.netapp.io/v1  
Kind:          TridentProvisioner  
Metadata:  
  Creation Timestamp: 2020-05-26T18:49:19Z  
  Generation:        1  
  Resource Version:  640347  
  Self Link:  
/apis/trident.netapp.io/v1/namespaces/trident/tridentprovisioners/triden  
t  
  UID:              52656806-0414-4ed8-b355-fc123fafbf4e  
Spec:  
  Debug:            true  
Status:  
  Message:          Trident installed  
  Status:           Installed  
  Version:          v20.04
```

```

Events:
Type      Reason     Age   From
Message
-----  -----
Normal    Installing  9m32s  trident-operator.netapp.io
Installing Trident
Normal    Installed   3m47s (x5 over 8m56s)  trident-operator.netapp.io
Trident installed
[user@rhel7 trident-installer]$ oc get pods -n trident
NAME                  READY   STATUS    RESTARTS   AGE
trident-csi-7f769c7875-s6fmt   5/5     Running   0          10m
trident-csi-cp7wg           2/2     Running   0          10m
trident-csi-hhx94           2/2     Running   0          10m
trident-csi-l72bt           2/2     Running   0          10m
trident-csi-xfl9d           2/2     Running   0          10m
trident-csi-xrhqx           2/2     Running   0          10m
trident-csi-zb7ws           2/2     Running   0          10m
trident-operator-564d7d66f-qrz7v 1/1     Running   0          27m

[user@rhel7 trident-installer]$ ./tridentctl -n trident version
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 20.04.0        | 20.04.0       |
+-----+-----+

```

11. Create a storage backend that will be used by Trident to provision volumes. The storage backend specifies the Element cluster in NetApp HCI. You also can specify sample bronze, silver, and gold types with corresponding QoS specs.

```
[user@rhel7 trident-installer]$ vi backend.json
{
    "version": 1,
    "storageDriverName": "solidfire-san",
    "Endpoint": "https://admin: admin- password@10.63.172.140/json-
rpc/8.0",
    "SVIP": "10.61.185.205:3260",
    "TenantName": "trident",
    "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS": 2000, "burstIOPS": 4000}}, {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS": 6000, "burstIOPS": 8000}}, {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS": 8000, "burstIOPS": 10000}}]
}
[user@rhel7 trident-installer]$ ./tridentctl -n trident create backend
-f backend.json
+-----+-----+
+-----+-----+-----+
|       NAME          | STORAGE DRIVER |           UUID
| STATE   | VOLUMES |
+-----+-----+
+-----+-----+-----+
| solidfire_10.61.185.205 | solidfire-san | 40f48d99-5d2e-4f6c-89ab-
8aee2be71255 | online |      0 |
+-----+-----+
+-----+-----+-----+
```

Modify the `backend.json` to accommodate the details or requirements of your environment for the following values:

- Endpoint corresponds to the credentials and the MVIP of the NetApp HCI Element cluster.
- SVIP corresponds to the SVIP configured over the VM network in the section titled [Create Storage Network VLAN](#).
- Types corresponds to different QoS bands. New persistent volumes can be created with specific QoS settings by specifying the exact storage pool.

12. Create a StorageClass that specifies Trident as the provisioner and the storage backend as `solidfire-san`.

```
[user@rhel7 trident-installer]$ vi storage-class-basic.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
  provisioningType: "thin"

[user@rhel7 trident-installer]$ oc create -f storage-class-basic.yaml
storageclass.storage.k8s.io/basic created
```



In this example, the StorageClass created is set as a default, however an OpenShift administrator can define multiple storage classes corresponding to different QoS requirements and other factors based upon their applications. Trident selects a storage backend that can satisfy all the criteria specified in the parameters section in the storage class definition. End users can then provision storage as needed, without administrative intervention.

[Next: Validation Results: NetApp HCI for Red Hat OpenShift on RHV](#)

## Validation Results: NetApp HCI for Red Hat OpenShift on RHV

This section provides the steps to deploy a continuous integration/continuous delivery or deployment (CI/CD) pipeline with Jenkins in order to validate the operation of the solution.

### Create the Resources Required for Jenkins Deployment

To create the resources required for deploying the Jenkins application, complete the following steps:

1. Create a new project named Jenkins.

# Create Project

Name \*

Display Name

Description

Cancel

Create

2. In this example, we deployed Jenkins with persistent storage. To support the Jenkins build, create the PVC. Navigate to Storage > Persistent Volume Claims and click Create Persistent Volume Claim. Select the storage class that was created, make sure that the Persistent Volume Claim Name is jenkins, select the appropriate size and access mode, and then click Create.

## Create Persistent Volume Claim

[Edit YAML](#)**Storage Class** SC basic

Storage class for the new claim.

**Persistent Volume Claim Name \*** jenkins

A unique name for the storage claim within the project.

**Access Mode \*** Single User (RWO)    Shared Access (RWX)    Read Only (ROX)

Permissions to the mounted drive.

**Size \*** 100

GiB



Desired storage capacity.

 Use label selectors to request storage

Use label selectors to define how storage is created.

## Deploy Jenkins with Persistent Storage

To deploy Jenkins with persistent storage, complete the following steps:

1. In the upper left corner, change the role from Administrator to Developer. Click +Add and select From Catalog. In the Filter by Keyword bar, search jenkins. Select Jenkins Service, with Persistent Storage.

## Developer Catalog

Add shared apps, services, or source-to-image builders to your project from the Developer Catalog. Cluster admins can install additional apps which will show up here automatically.

All Items

Languages

Databases

Middleware

CI/CD

Other

Type

- Operator Backed (0)
- Helm Charts (0)
- Builder Image (0)
- Template (4)
- Service Class (0)

All Items

jenkins

Group By: None ▾


Jenkins
Template


Jenkins
Template


Jenkins (Ephemeral)
Template


Jenkins (Ephemeral)
Template

### 2. Click Instantiate Template.



**Jenkins**

Provided by Red Hat, Inc.

Instantiate Template

---

Provider	Description
Red Hat, Inc.	Jenkins service, with persistent storage.
<b>Support</b>	NOTE: You must have persistent volumes available in your cluster to use this template.
<a href="#">Get support ↗</a>	
<b>Created At</b>	<span style="font-size: small;">May 26, 3:58 am</span>
	<b>Documentation</b>
	<a href="https://docs.okd.io/latest/using_images/other_images/jenkins.html">https://docs.okd.io/latest/using_images/other_images/jenkins.html ↗</a>

### 3. By default, the details for the Jenkins application are populated. Based on your requirements, modify the parameters, and click Create. This process creates all the required resources for supporting Jenkins on

## OpenShift.

### Instantiate Template

Namespace \*

Jenkins Service Name

The name of the OpenShift Service exposed for the Jenkins container.

Jenkins JNLP Service Name

The name of the service used for master/slave communication.

Enable OAuth in Jenkins

Whether to enable OAuth OpenShift integration. If false, the static account 'admin' will be initialized with the password 'password'.

Memory Limit

Maximum amount of memory the container can use.

Volume Capacity \*

Volume space available for data, e.g. 512Mi, 2Gi.

Jenkins ImageStream Namespace

The OpenShift Namespace where the Jenkins ImageStream resides.

Disable memory intensive administrative monitors

Whether to perform memory intensive, possibly slow, synchronization with the Jenkins Update Center on start. If true, the Jenkins core update monitor and site warnings monitor are disabled.

Jenkins ImageStreamTag

Name of the ImageStreamTag to be used for the Jenkins image.

Fatal Error Log File

When a fatal error occurs, an error log is created with information and the state obtained at the time of the fatal error.

Allows use of Jenkins Update Center repository with invalid SSL certificate

Whether to allow use of a Jenkins Update Center that uses invalid certificate (self-signed, unknown CA). If any value other than 'false', certificate check is bypassed. By default, certificate check is enforced.

Create Cancel



Jenkins service, with persistent storage.

NOTE: You must have persistent volumes available in your cluster to use this template.

The following resources will be created:

- DeploymentConfig
- PersistentVolumeClaim
- RoleBinding
- Route
- Service
- ServiceAccount

4. The Jenkins pods take approximately 10–12 minutes to enter the Ready state.

Project: jenkins ▾

## Pods

Create Pod

Filter by name...

1 Running | 0 Pending | 0 Terminating | 0 CrashLoopBackOff | 1 Completed | 0 Failed | 0 Unknown

Select all filters

1 of 2 Items

Name	Namespace	Status	Ready	Owner	Memory	CPU	⋮
 jenkins-1-c77n9	 jenkins	 Running	1/1	 jenkins-1	-	0.004 cores	⋮

5. After the pods are instantiated, navigate to Networking > Routes. To open the Jenkins webpage, click the URL provided for the jenkins route.

Project: jenkins ▾

## Routes

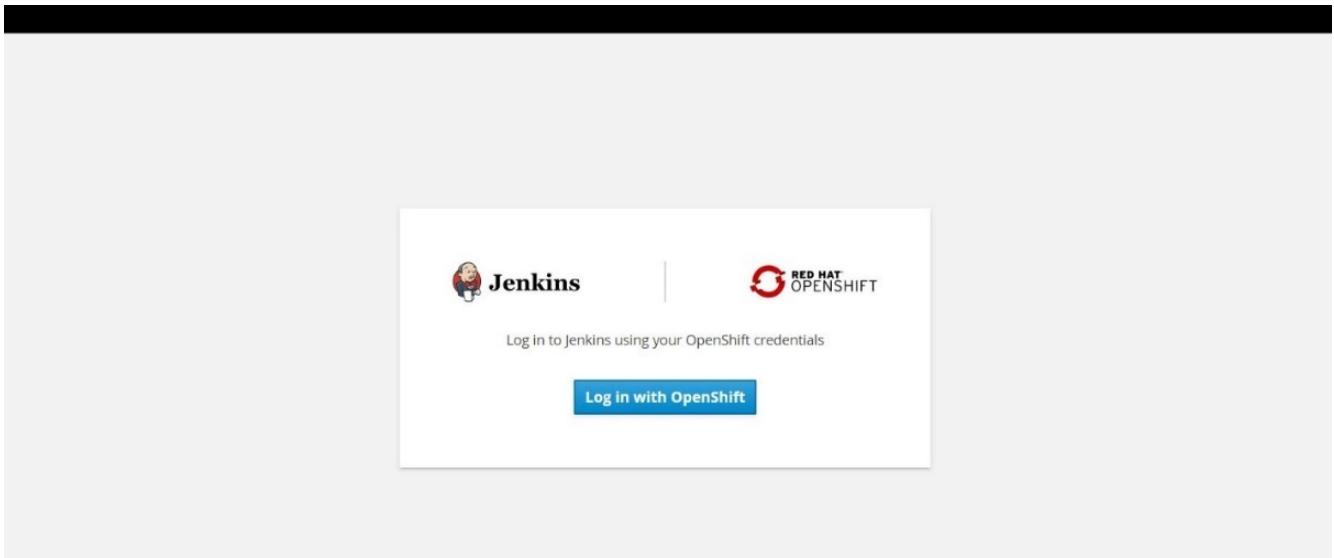
Create Route

Filter by name...

1 Accepted | 0 Rejected | 0 Pending | Select all filters | 1 Item

Name	Namespace	Status	Location	Service	⋮
 jenkins	 jenkins	 Accepted	https://jenkins-jenkins.apps.rhv-ocp-cluster.cie.netapp.com	 jenkins	⋮

6. Because the OpenShift OAuth was used while creating the Jenkins app, click Log in with OpenShift.



7. Authorize jenkins service-account to access the OpenShift users.

## Authorize Access

Service account jenkins in project jenkins is requesting permission to access your account (kube:admin)

### Requested permissions

#### user:info

Read-only access to your user information (including username, identities, and group membership)

#### user:check-access

Read-only access to view your privileges (for example, "can I create builds?")

You will be redirected to <https://jenkins-jenkins.apps.rhv-ocp-cluster.cie.netapp.com/securityRealm/finishLogin>

[Allow selected permissions](#) [Deny](#)

8. The Jenkins welcome page is displayed. Because we are using a Maven build, complete the Maven installation first. Navigate to Manage Jenkins > Global Tool Configuration, then in the Maven subhead, click Add Maven. Enter the name of your choice and make sure that the Install Automatically option is selected. Click Save.

Maven

Maven installations

Add Maven

Maven

Name: M3

Install automatically

Install from Apache

Version: 3.6.3

Add Installer

Delete Maven

Add Maven

List of Maven installations on this system

9. You can now create a pipeline to demonstrate the CI/CD workflow. On the home page, click Create New Jobs or New Item from the left-hand menu.

The screenshot shows the Jenkins home page. At the top, there is a navigation bar with the Jenkins logo, a search bar, and user information (kube:admin | log out). Below the navigation bar, there is a sidebar with links: New Item, People, Build History, Manage Jenkins, My Views, Open Blue Ocean, Lockable Resources, Credentials, and New View. The main content area features a "Welcome to Jenkins!" message with a sub-instruction: "Please [create new jobs](#) to get started." Below this, there are two sections: "Build Queue" (No builds in the queue) and "Build Executor Status" (1 Idle, 2 Idle).

10. On the Create Item page, enter the name of your choice, select Pipeline, and click Ok.

The screenshot shows the "Enter an item name" dialog. The input field contains "sample-demo" with the note "» Required field". Below the input field, there is a list of project types:

- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Bitbucket Team/Project**: Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**: Scans a GitHub organization (or user account) for all repositories matching some defined markers.

At the bottom of the dialog, there is an "OK" button and a note: "Creates a set of Pipeline projects according to detected branches in one SCM repository".

11. Select the Pipeline tab. From the Try Sample Pipeline drop-down menu, select Github + Maven. The code is automatically populated. Click Save.

General Build Triggers Advanced Project Options **Pipeline** Advanced...

## Pipeline

Definition Pipeline script

Script

```
1 node {
2     def mvnHome
3     stage('Preparation') { // for display purposes
4         // Get some code from a GitHub repository
5         git 'https://github.com/jglick/simple-maven-project-with-tests.git'
6         // Get the Maven tool.
7         // ** NOTE: This 'M3' Maven tool must be configured
8         // ** in the global configuration.
9         mvnHome = tool 'M3'
10    }
11    stage('Build') {
12        // Run the maven build
13        withEnv(["MVN_HOME=$mvnHome"]) {
14            if (isUnix()) {
15                sh '$MVN_HOME/bin/mvn' -Dmaven.test.failure.ignore clean package'
16            } else {
17                bat("%MVN_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package)
18            }
19        }
20    }
21}
```

GitHub + Maven

Use Groovy Sandbox

[Pipeline Syntax](#)

**Save** **Apply**

12. Click Build Now to trigger the development through the preparation, build, and testing phase. It can take several minutes to complete the whole build process and display the results of the build.

 Jenkins

Jenkins > sample-demo >

 Back to Dashboard

 Status

 Changes

 Build Now

 Delete Pipeline

 Configure

 Full Stage View

 Open Blue Ocean

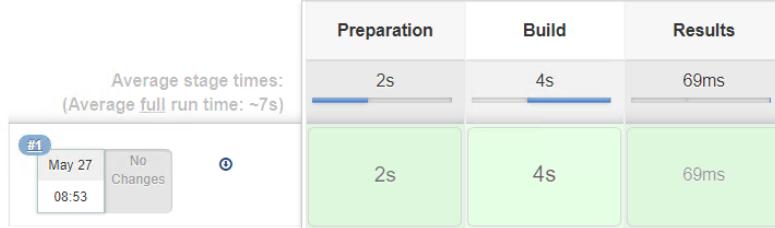
 Rename

 Pipeline Syntax

## Pipeline sample-demo



### Stage View



### Permalinks

- [Last build \(#1\), 1 min 23 sec ago](#)
- [Last stable build \(#1\), 1 min 23 sec ago](#)
- [Last successful build \(#1\), 1 min 23 sec ago](#)
- [Last completed build \(#1\), 1 min 23 sec ago](#)

13. Whenever there are any code changes, the pipeline can be rebuilt to patch the new version of software enabling continuous integration and continuous delivery. Click Recent Changes to track the changes from the previous version.

**Pipeline sample-demo**

**Stage View**

	Preparation	Build	Results
Average stage times: (Average full run time: ~6s)	2s	4s	86ms
#2 May 27, 2020 3:56 PM No Changes	1s	4s	104ms
#1 May 27, 08:53 No Changes	2s	4s	69ms

**Permalinks**

- [Last build \(#2\), 19 sec ago](#)
- [Last stable build \(#2\), 19 sec ago](#)
- [Last successful build \(#2\), 19 sec ago](#)
- [Last completed build \(#2\), 19 sec ago](#)

Next: Best Practices for Production Deployments

## Best Practices for Production Deployments - NetApp HCI for Red Hat OpenShift on RHV

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

### Deploy OpenShift to an RHV Cluster of at Least Three Nodes

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying two RHV-H hypervisor nodes and ensuring a fault tolerant configuration where both hosts can manage the hosted-engine and deployed VMs can migrate between the two hypervisors. Because Red Hat OpenShift initially deploys with three master nodes, it is ensured in a two-node configuration that at least two masters will occupy the same node, which can lead to a possible outage for OpenShift if that specific node becomes unavailable. Therefore, it is a Red Hat best practice that at least three RHV-H hypervisor nodes be deployed as part of the solution so that the OpenShift masters can be distributed evenly, and the solution receives an added degree of fault tolerance.

### Configure Virtual Machine/Host Affinity

Ensuring the distribution of the OpenShift masters across multiple hypervisor nodes can be achieved by enabling VM/host affinity. Affinity is a way to define rules for a set of VMs and/or hosts that determine whether

the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity. The conditions defined for the parameters can be either hard enforcement or soft enforcement. Hard enforcement ensures that the VMs in an affinity group always follows the positive/negative affinity strictly without any regards to external conditions. Soft enforcement, on the other hand, ensures that a higher preference is set out for the VMs in an affinity group to follow the positive/negative affinity whenever feasible. In a two or three hypervisor configuration as described in this document soft affinity is the recommended setting, in larger clusters hard affinity can be relied on to ensure OpenShift nodes are distributed. To configure affinity groups, see the [Red Hat 6.11. Affinity Groups documentation](#).

## Use a Custom Install File for OpenShift Deployment

IPI makes the deployment of OpenShift clusters extremely easy through the interactive wizard discussed earlier in this document. However, it is possible that there are some default values that might need to be changed as a part of a cluster deployment. In these instances, the wizard can be run and tasked without immediately deploying a cluster, but instead outputting a configuration file from which the cluster can be deployed later. This is very useful if any IPI defaults need to be changed, or if a user wants to deploy multiple identical clusters in their environment for other uses such as multitenancy. For more information about creating a customized install configuration for OpenShift, see [Red Hat OpenShift Installing a Cluster on RHV with Customizations](#).

Next: [Videos and Demos: NetApp HCI for Red Hat OpenShift on Red Hat Virtualization](#)

## Videos and Demos: NetApp HCI for Red Hat OpenShift on RHV

The following video demonstrates some of the capabilities documented in this document:

 | [NetApp HCI for Red Hat OpenShift on Red Hat Virtualization](#)

Next: [Additional Information: NetApp HCI for Red Hat OpenShift on Red Hat Virtualization](#)

## Additional Information: NetApp HCI for Red Hat OpenShift on RHV

To learn more about the information described in this document, review the following websites:

- NetApp HCI Documentation <https://www.netapp.com/us/documentation/hci.aspx>
- NetApp Trident Documentation <https://netapp-trident.readthedocs.io/en/stable-v20.04/>
- Red Hat Virtualization Documentation [https://access.redhat.com/documentation/en-us/red\\_hat\\_virtualization/4.3/](https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.3/)
- Red Hat OpenShift Documentation [https://access.redhat.com/documentation/en-us/openshift\\_container\\_platform/4.4/](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.4/)

# NVA-1160: Red Hat OpenShift with NetApp

Alan Cowles and Nikhil M Kulkarni, NetApp

This reference document provides deployment validation of the Red Hat OpenShift solution, deployed through Installer Provisioned Infrastructure (IPI) in several different data center environments as validated by NetApp. It also details storage integration with NetApp storage systems by making use of the Astra Trident storage orchestrator for the management of persistent storage. Lastly, a number of solution validations and real world use cases are explored and documented.

## Use Cases

The Red Hat OpenShift with NetApp solution is architected to deliver exceptional value for customers with the following use cases:

- Easy to deploy and manage Red Hat OpenShift deployed using IPI (Installer Provisioned Infrastructure) on bare metal, Red Hat OpenStack Platform, Red Hat Virtualization, and VMware vSphere.
- Combined power of enterprise container and virtualized workloads with Red Hat OpenShift deployed virtually on OSP, RHV, or vSphere, or on bare metal with OpenShift Virtualization.
- Real world configuration and use cases highlighting the features of Red Hat OpenShift when used with NetApp storage and Astra Trident, the open source storage orchestrator for Kubernetes.

## Business Value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure, such as the following:

- High availability at all layers in the stack
- Ease of deployment procedures
- Non-disruptive operations and upgrades
- API-driven and programmable infrastructure to keep up with microservices agility
- Multitenancy with performance guarantees
- Ability to run virtualized and containerized workloads simultaneously
- Ability to scale infrastructure independently based on workload demands

Red Hat OpenShift with NetApp acknowledges these challenges and presents a solution that helps address each concern by implementing the fully automated deployment of Red Hat OpenShift IPI in the customer's choice of data center environment.

## Technology Overview

The Red Hat OpenShift with NetApp solution is comprised of the following major components:

### Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform is a fully supported enterprise Kubernetes platform. Red Hat makes several enhancements to open-source Kubernetes to deliver an application platform with all the components fully integrated to build, deploy, and manage containerized applications.

For more information visit the OpenShift website [here](#).

### NetApp Storage Systems

NetApp has several storage systems perfect for enterprise data centers and hybrid cloud deployments. The NetApp portfolio includes NetApp ONTAP, NetApp Element, and NetApp e-Series storage systems, all of which can be utilized to provide persistent storage for containerized applications.

For more information visit the NetApp website [here](#).

## NetApp Storage Integrations

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, deployed in an on-prem environment, powered by NetApp's trusted data protection technology.

For more information visit the NetApp Astra website [here](#).

Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift.

For more information, visit the Astra Trident website [here](#).

## Advanced Configuration Options

This section is dedicated to customizations that real world users would likely need to perform when deploying this solution into production, such as creating a dedicated private image registry, or deploying custom load balancer instances.

## Current Support Matrix for Validated Releases

Technology	Purpose	Software Version
NetApp ONTAP	Storage	9.8
NetApp Element	Storage	12.3
NetApp Astra Control Center	Application Aware Data Management	21.08.57
NetApp Astra Trident	Storage Orchestration	21.04
Red Hat OpenShift	Container Orchestration	4.6 EUS, 4.7
Red Hat OpenStack Platform	Private Cloud Infrastructure	16.1
Red Hat Virtualization	Data Center Virtualization	4.4
VMware vSphere	Data Center Virtualization	6.7U3

Next: [Red Hat OpenShift Overview](#).

## OpenShift Overview: Red Hat OpenShift with NetApp

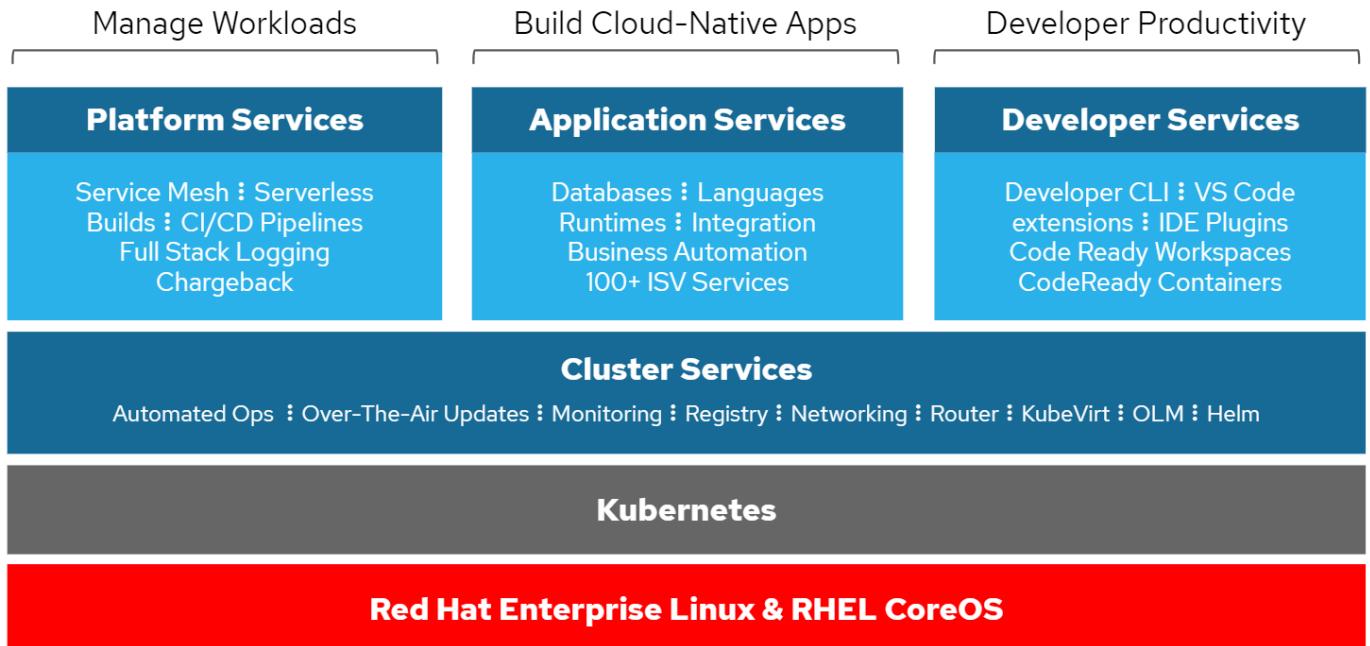
The Red Hat OpenShift Container Platform unites development and IT operations on a single platform to build, deploy, and manage applications consistently across on-premises and hybrid cloud infrastructures. Red Hat OpenShift is built on open-source innovation and industry standards, including Kubernetes and Red Hat Enterprise Linux CoreOS, the world's leading enterprise Linux distribution designed for container-based workloads. OpenShift is part of the Cloud Native Computing Foundation (CNCF) Certified Kubernetes program, providing portability and interoperability of container workloads.

### Red Hat OpenShift provides the following capabilities:

- **Self-service provisioning.** Developers can quickly and easily create applications on demand from the tools that they use most, while operations retain full control over the entire environment.
- **Persistent storage.** By providing support for persistent storage, OpenShift Container Platform allows you

to run both stateful applications and cloud-native stateless applications.

- **Continuous integration and continuous development (CI/CD).** This source-code platform manages build and deployment images at scale.
- **Open-source standards.** These standards incorporate the Open Container Initiative (OCI) and Kubernetes for container orchestration, in addition to other open-source technologies. You are not restricted to the technology or to the business roadmap of a specific vendor.
- **CI/CD pipelines.** OpenShift provides out-of-the-box support for CI/CD pipelines so that development teams can automate every step of the application delivery process and make sure it's executed on every change that is made to the code or configuration of the application.
- **Role-Based Access Control (RBAC).** This feature provides team and user tracking to help organize a large developer group.
- **Automated build and deploy.** OpenShift gives developers the option to build their containerized applications or have the platform build the containers from the application source code or even the binaries. The platform then automates deployment of these applications across the infrastructure based on the characteristic that was defined for the applications. For example, how quantity of resources that should be allocated and where on the infrastructure they should be deployed in order for them to be compliant with third-party licenses.
- **Consistent environments.** OpenShift makes sure that the environment provisioned for developers and across the lifecycle of the application is consistent from the operating system, to libraries, runtime version (for example, Java runtime), and even the application runtime in use (for example, tomcat) in order to remove the risks originated from inconsistent environments.
- **Configuration management.** Configuration and sensitive data management is built in to the platform to make sure that a consistent and environment agnostic application configuration is provided to the application no matter which technologies are used to build the application or which environment it is deployed.
- **Application logs and metrics.** Rapid feedback is an important aspect of application development. OpenShift integrated monitoring and log management provides immediate metrics back to developers in order for them to study how the application is behaving across changes and be able to fix issues as early as possible in the application lifecycle.
- **Security and container catalog.** OpenShift offers multitenancy and protects the user from harmful code execution by using established security with Security-Enhanced Linux (SELinux), CGroups, and Secure Computing Mode (seccomp) to isolate and protect containers, encryption through TLS certificates for the various subsystems, and providing access to Red Hat certified containers ([access.redhat.com/containers](http://access.redhat.com/containers)) that are scanned and graded with a specific emphasis on security to provide certified, trusted, and secure application containers to end users.



Physical



Virtual



Private cloud



Public cloud



Managed cloud  
(Azure, AWS, IBM, Red Hat)

## Deployment methods for Red Hat OpenShift

Starting with Red Hat OpenShift 4, the deployment methods for OpenShift include manual deployments using User Provisioned Infrastructure (UPI) for highly customized deployments, or fully automated deployments using Installer Provisioned Infrastructure (IPI).

The IPI installation method is the preferred method in most cases because it allows for the rapid deployment of OCP clusters for dev, test, and production environments.

### IPI installation of Red Hat OpenShift

The Installer Provisioned Infrastructure (IPI) deployment of OpenShift involves these high-level steps:

1. Visit the Red Hat OpenShift [website](#) and login with your SSO credentials.
2. Select the environment that you'd like to deploy Red Hat OpenShift into.

## Install OpenShift Container Platform 4

Select an infrastructure provider

 Run on Amazon Web Services	 Run on Microsoft Azure	 Run on Google Cloud Platform	 Run on VMware vSphere
 Run on Red Hat OpenStack	 Run on Red Hat Virtualization	 Run on Bare Metal	 Run on IBM Z
 Run on Power	 Run on Laptop Powered by Red Hat CodeReady Containers		

3. On the next screen download the installer, the unique pull secret, the CLI tools for management.

**Downloads**

**OpenShift installer**  
Download and extract the install program for your operating system and place the file in the directory where you will store the installation configuration files. Note: The OpenShift install program is only available for Linux and macOS at this time.

Linux ▾ [Download installer](#)

**Pull secret**  
Download or copy your pull secret. The install program will prompt you for your pull secret during installation.

[Download pull secret](#)  [Copy pull secret](#)

**Command-line interface**  
Download the OpenShift command-line tools and add them to your PATH.

Linux ▾ [Download command-line tools](#)

When the installer is complete you will see the console URL and credentials for accessing your new cluster. A kubeconfig file will also be generated for you to use with the `oc` CLI tools you downloaded.

4. Follow the [installation instructions](#) provided by Red Hat to deploy to your environment of choice.

**NetApp validated OpenShift deployments**

NetApp has tested and validated the deployment of Red Hat OpenShift in its labs using the Installer Provisioned Infrastructure (IPI) deployment method in each of the following data center environments:

- [OpenShift on Bare Metal](#)
- [OpenShift on Red Hat OpenStack Platform](#)
- [OpenShift on Red Hat Virtualization](#)
- [OpenShift on VMware vSphere](#)

[Next: NetApp Storage Overview.](#)

## OpenShift on Bare Metal: Red Hat OpenShift with NetApp

OpenShift on Bare Metal provides an automated deployment of OpenShift Container Platform on commodity servers.

Similar to virtual deployments of OpenShift, which are quite popular because they allow for ease of deployment, rapid provisioning, and scaling of OpenShift clusters, while also supplying the need to support virtualized workloads for applications that are not ready to be containerized. By deploying on bare metal, a customer does require the extra overhead in managing the host hypervisor environment, as well as the OpenShift environment. By deploying directly on bare metal servers, the customer can also reduce the physical overhead limitations of having to share resources between the host and OpenShift environment.

**OpenShift on Bare Metal provides the following features:**

- **IPI or Assisted Installer Deployment** With an OpenShift cluster deployed by Installer Provisioned Infrastructure (IPI) on bare metal servers, customers can deploy a highly versatile, easily scalable OpenShift environment directly on commodity servers, without the need to manage a hypervisor layer.
- **Compact Cluster Design** To minimize the hardware requirements, OpenShift on bare metal allows for users to deploy clusters of just 3 nodes, by enabling the OpenShift control plane nodes to also act as worker nodes and host containers.
- **OpenShift Virtualization** OpenShift can run virtual machines within containers by using OpenShift Virtualization. This container-native virtualization runs the KVM hypervisor inside of a container, and attaches persistent volumes for VM storage.
- **AI/ML-Optimized Infrastructure** Deploy applications like Kubeflow for Machine Learning applications by incorporating GPU-based worker nodes to your OpenShift environment and leveraging OpenShift Advanced Scheduling.

### Network design

The Red Hat OpenShift on NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality.

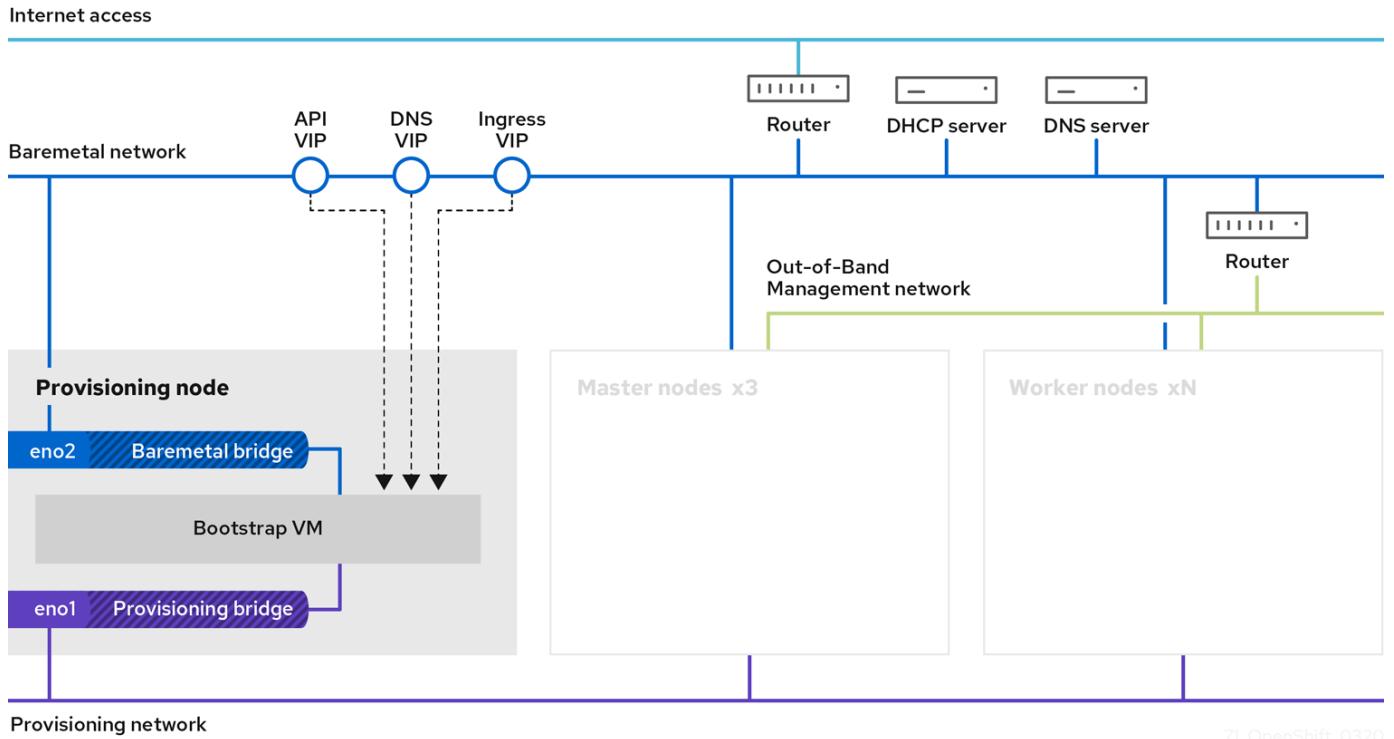
The OpenShift bare metal IPI deployment requires the customer to create a Provisioner node, a Red Hat Enterprise Linux 8 machine which will need to have network interfaces attached to separate networks.

- **Provisioning Network:** This network is used to boot the bare metal nodes and install the necessary images and packages to deploy the OpenShift cluster.
- **Baremetal Network:** This network is used for public facing communication of the cluster once it is deployed.

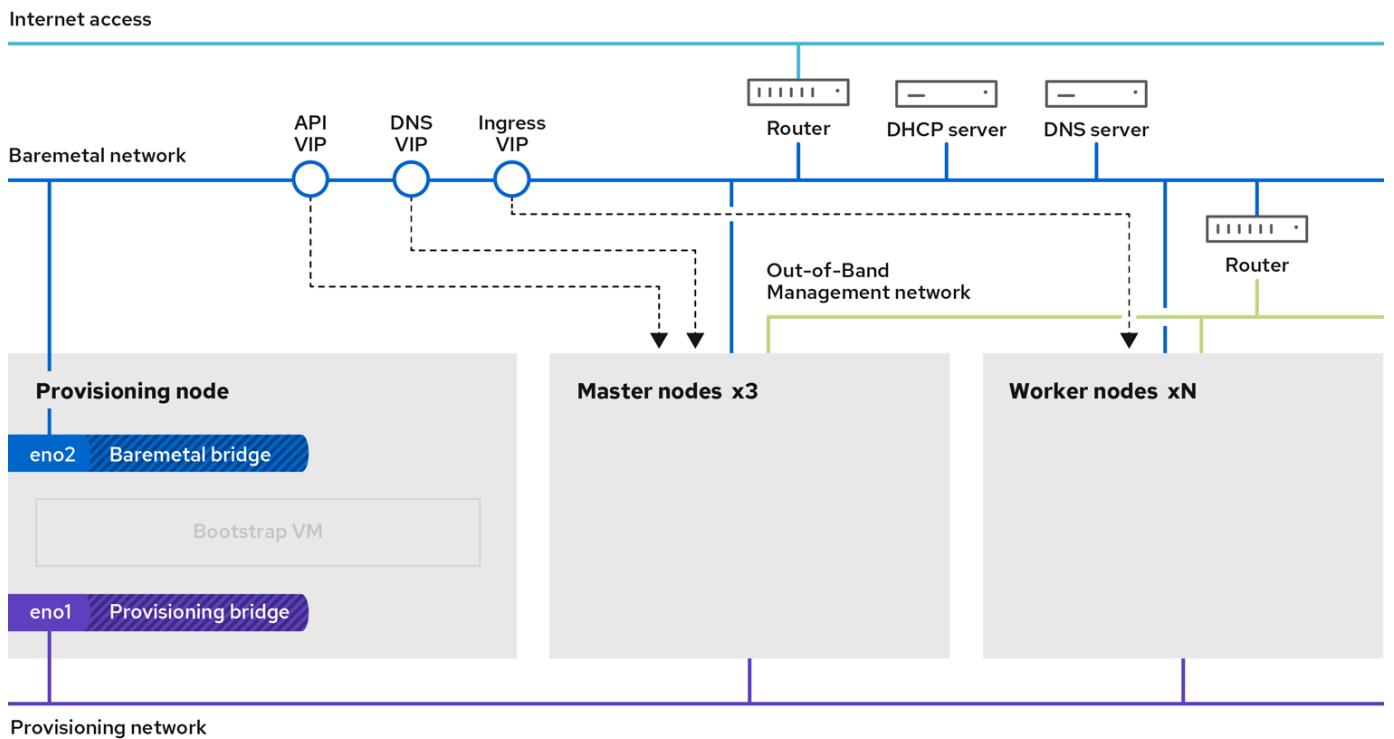
The setup of the provisioner node will have the customer create bridge interfaces that allow the traffic to route properly on both the node itself, and for the Bootstrap VM which will be provisioned for deployment purposes.

Once the cluster is deployed the API and Ingress VIP addresses are migrated from the bootstrap node to the newly deployed cluster.

The images below display the environment both during IPI deployment, and once the deployment is complete.



7L\_OpenShift\_0320



## VLAN requirements

The Red Hat OpenShift with NetApp solution is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs).

VLANs	Purpose	VLAN ID
Out-of-band Management Network	Management for bare metal nodes and IPMI	16
Bare Metal Network	Network for OpenShift services once cluster is available	181
Provisioning Network	Network for PXE boot and installation of bare metal nodes via IPI	3485



While each of these networks were virtually separated by VLANs, each physical port needed to be set up in "access mode" with the primary VLAN assigned, as there is no way to pass a VLAN tag during a PXE boot sequence.

## Network infrastructure support resources

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform:

- At least one DNS server which provides a full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.

[Next: NetApp Storage Overview.](#)

## OpenShift on Red Hat OpenStack Platform: Red Hat OpenShift with NetApp

Red Hat OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable private OpenStack cloud.

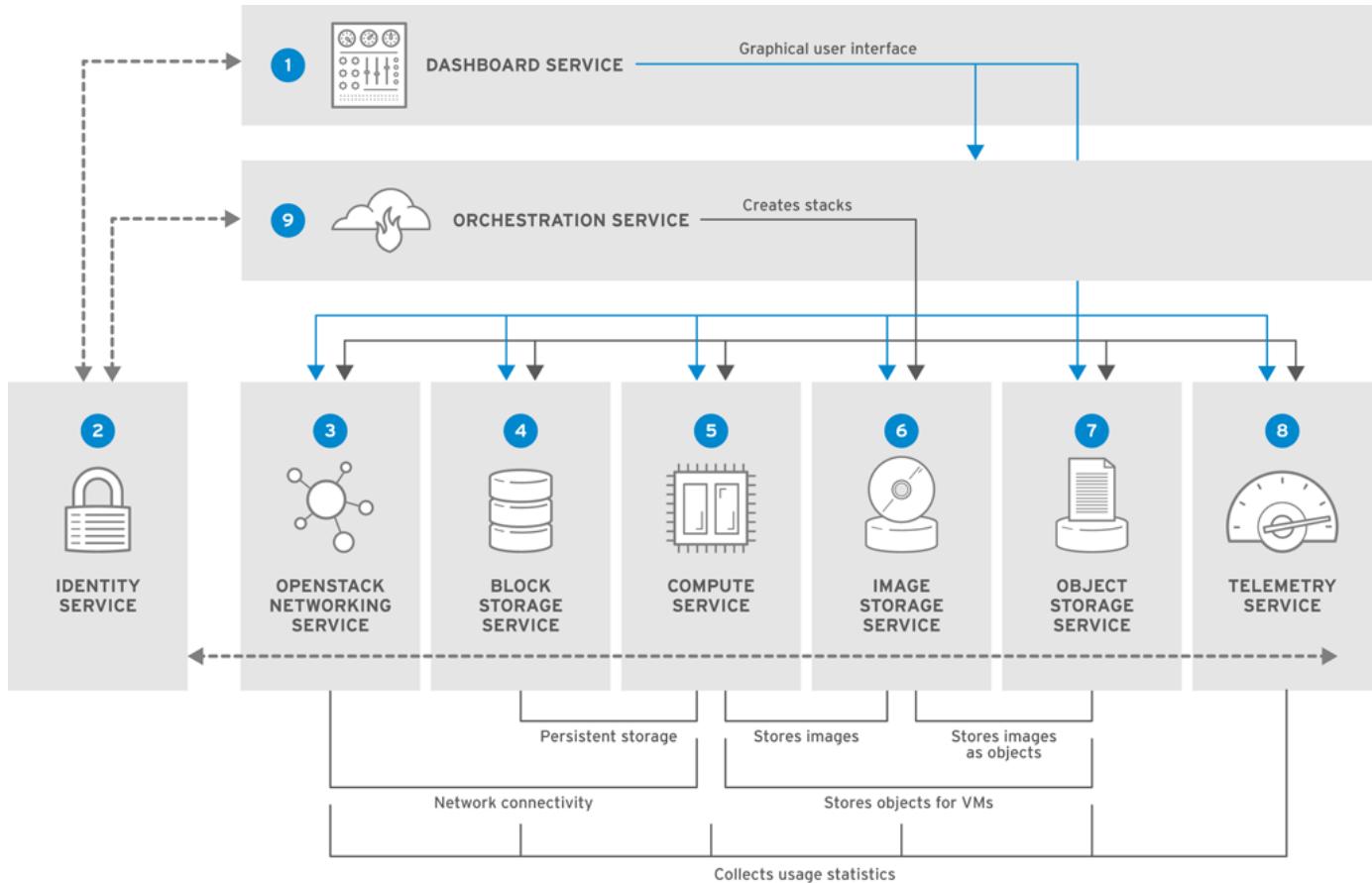
OSP is an infrastructure-as-a-service (IaaS) cloud implemented by a collection of control services that manage compute, storage, and networking resources. The environment is managed using a web-based interface that allows administrators and users to control, provision, and automate OpenStack resources. Additionally, the OpenStack infrastructure is facilitated through an extensive command line interface and API enabling full automation capabilities for administrators and end-users.

The OpenStack project is a rapidly developed community project, and provides updated releases every six months. Initially Red Hat OpenStack Platform kept pace with this release cycle by publishing a new release along with every upstream release, and providing long term support for every third release. Recently with the OSP 16.0 release (based on OpenStack Train) Red Hat has chosen not to keep pace with release numbers, but instead backport new features into sub-releases. The most recent release is Red Hat OpenStack Platform 16.1, which includes backported advanced features from the Ussuri and Victoria releases upstream.

For more information about OSP see the [Red Hat OpenStack Platform website](#).

## OpenStack services

OpenStack Platform services are deployed as containers, which isolates services from one another and enables easy upgrades. The OpenStack Platform uses a set of containers built and managed with Kolla. The deployment of services is performed by pulling container images from the Red Hat Custom Portal. These service containers are managed using the Podman command, and are deployed, configured, and maintained with Red Hat OpenStack Director.



Service	Project Name	Description
Dashboard	Horizon	Web browser-based dashboard that you use to manage OpenStack services.
Identity	Keystone	Centralized service for authentication and authorization of OpenStack services and for managing users, projects, and roles.
OpenStack Networking	Neutron	Provides connectivity between the interfaces of OpenStack services.
Block Storage	Cinder	Manages persistent block storage volumes for virtual machines (VMs).
Compute	Nova	Manages and provisions VMs running on compute nodes.
Image	Glance	Registry service used to store resources such as VM images and volume snapshots.
Object Storage	Swift	Allows users to storage and retrieve files and arbitrary data.

Telemetry	Ceilometer	Provides measurements of use of cloud resources.
Orchestration	Heat	Template-based orchestration engine that supports automatic creation of resource stacks.

### Network design

The Red Hat OpenShift with NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality.

IPMI functionality is required by Red Hat OpenStack Director to deploy Red Hat OpenStack Platform using the Ironic baremetal provision service.

### VLAN requirements

Red Hat OpenShift with NetApp is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). This configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution while validating the solution at NetApp.

VLANs	Purpose	VLAN ID
Out-of-band Management Network	Network used for management of physical nodes and IPMI service for Ironic.	16
Storage Infrastructure	Network used for controller nodes to map volumes directly to support infrastructure services like Swift.	201
Storage Cinder	Network used to map and attach block volumes directly to virtual instances deployed in the environment.	202
Internal API	Network used for communication between the OpenStack services using API communication, RPC messages, and database communication.	301
Tenant	Neutron provides each tenant with their own networks via tunneling through VXLAN. Network traffic is isolated within each tenant network. Each tenant network has an IP subnet associated with it, and network namespaces mean that multiple tenant networks can use the same address range without causing conflicts	302

VLANs	Purpose	VLAN ID
Storage Management	OpenStack Object Storage (Swift) uses this network to synchronize data objects between participating replica nodes. The proxy service acts as the intermediary interface between user requests and the underlying storage layer. The proxy receives incoming requests and locates the necessary replica to retrieve the requested data.	303
PXE	The OpenStack Director provides PXE boot as a part of the Ironic bare metal provisioning service to orchestrate the installation of the OSP Overcloud.	3484
External	Publicly available network which hosts the OpenStack Dashboard (Horizon) for graphical management, and allows for public API calls to manage OpenStack services.	3485
In-band management network	Provides access for system administration functions such as SSH access, DNS traffic, and Network Time Protocol (NTP) traffic. This network also acts as a gateway for non-controller nodes.	3486

## Network infrastructure support resources

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform:

- At least one DNS server which provides a full host-name resolution.
- At least three NTP servers which can keep time synchronized for the servers in the solution.
- (Optional) Outbound internet connectivity for the OpenShift environment.

## Best practices for production deployments

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

## Deploy OpenShift to an OSP private cloud with at least three compute nodes

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying three OSP controller nodes and two OSP compute nodes, and ensuring a fault tolerant configuration where both compute nodes can launch the virtual instances and deployed VMs can migrate between the two hypervisors.

Because Red Hat OpenShift initially deploys with three master nodes, it is ensured in a two-node configuration that at least two masters will occupy the same node, which can lead to a possible outage for OpenShift if that

specific node becomes unavailable. Therefore, it is a Red Hat best practice that at least three OSP compute nodes be deployed as part of the solution so that the OpenShift masters can be distributed evenly, and the solution receives an added degree of fault tolerance.

## Configure virtual machine/host affinity

Ensuring the distribution of the OpenShift masters across multiple hypervisor nodes can be achieved by enabling VM/host affinity.

Affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity. In Red Hat OpenStack Platform, host affinity and anti-affinity rules can be created and enforced by creating Server Groups and configuring filters so that instances deployed by Nova in a server group deploy on different compute nodes.

A server group has a default maximum of 10 virtual instances that it can manage placement for. This can be modified by updating the default quotas for Nova.



There is a specific hard affinity/anti-affinity limit for OSP Server Groups, where if there are not enough resources to deploy on separate nodes, or not enough resources to allow sharing of nodes, the VM will fail to boot.

To configure affinity groups, see [How do I configure Affinity and Anti-Affinity for OpenStack instances?](#).

## Use a custom install file for OpenShift deployment

IPI makes the deployment of OpenShift clusters extremely easy through the interactive wizard discussed earlier in this document. However, it is possible that there are some default values that might need to be changed as a part of a cluster deployment.

In these instances, the wizard can be run and tasked without immediately deploying a cluster, but instead outputting a configuration file from which the cluster can be deployed later. This is very useful if any IPI defaults need to be changed, or if a user wants to deploy multiple identical clusters in their environment for other uses such as multitenancy. For more information about creating a customized install configuration for OpenShift, see [Red Hat OpenShift Installing a Cluster on OpenStack with Customizations](#).

Next: [NetApp Storage Overview](#).

## OpenShift on Red Hat Virtualization: Red Hat OpenShift with NetApp

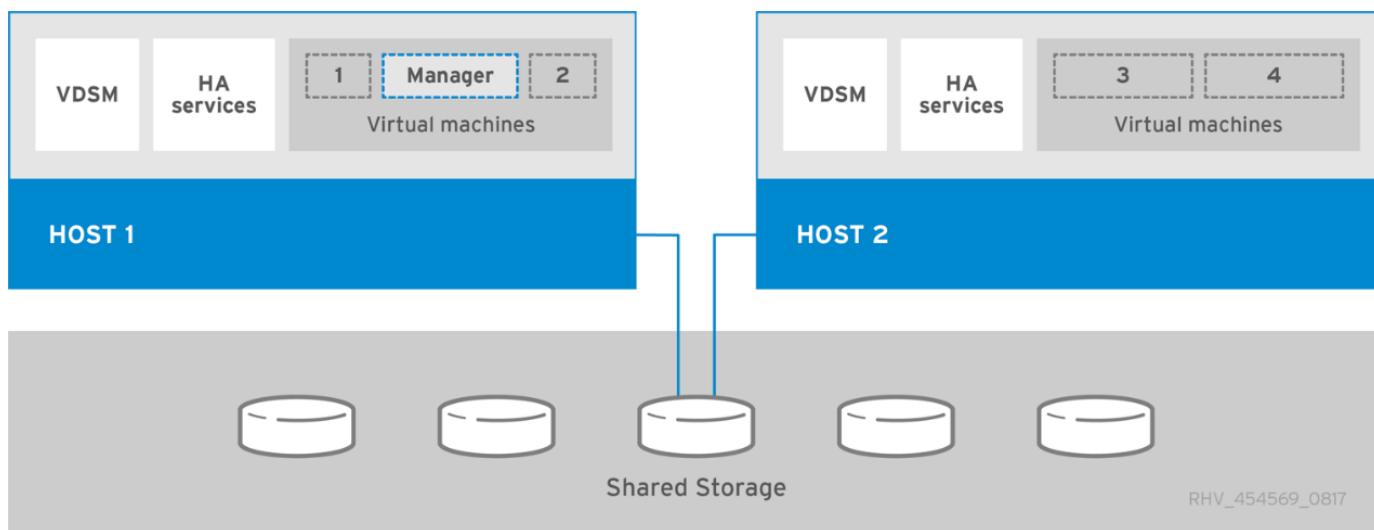
RHV is an enterprise virtual data center platform that runs on Red Hat Enterprise Linux (RHEL) and uses the KVM hypervisor.

For more information about RHV, see the [Red Hat Virtualization website](#).

RHV provides the following features:

- **Centralized management of VMs and hosts.** The RHV manager runs as a physical or virtual machine (VM) in the deployment and provides a web-based GUI for the management of the solution from a central interface.
- **Self-hosted engine.** To minimize the hardware requirements, RHV allows RHV Manager (RHV-M) to be deployed as a VM on the same hosts that run guest VMs.

- **High availability.** In event of host failures, to avoid disruption, RHV allows VMs to be configured for high availability. The highly available VMs are controlled at the cluster level using resiliency policies.
- **High scalability.** A single RHV cluster can have up to 200 hypervisor hosts enabling it to support requirements of massive VMs to hold resource-greedy, enterprise-class workloads.
- **Enhanced security.** Inherited from RHV, Secure Virtualization (sVirt) and Security Enhanced Linux (SELinux) technologies are employed by RHV for the purposes of elevated security and hardening for the hosts and VMs. The key advantage from these features is logical isolation of a VM and its associated resources.



## Network design

The Red Hat OpenShift on NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality. OCP uses the Virtual Machine logical network on RHV for its cluster management. This section describes the arrangement and purpose of each virtual network segment used in the solution and outlines the pre-requisites for deployment of the solution.

## VLAN requirements

Red Hat OpenShift on RHV is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). This configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution while validating the solution at NetApp.

VLANs	Purpose	VLAN ID
Out-of-band Management Network	Management for physical nodes and IPMI	16
VM Network	Virtual Guest Network Access	1172
In-band Management Network	Management for RHV-H Nodes, RHV-Manager, ovirtmgmt network	3343
Storage Network	Storage network for NetApp Element iSCSI	3344
Migration Network	Network for virtual guest migration	3345

## **Network infrastructure support resources**

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform:

- At least one DNS server which provides a full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.

## **Best practices for production deployments**

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

### **Deploy OpenShift to an RHV cluster of at least three nodes**

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying two RHV-H hypervisor nodes and ensuring a fault tolerant configuration where both hosts can manage the hosted-engine and deployed VMs can migrate between the two hypervisors.

Because Red Hat OpenShift initially deploys with three master nodes, it is ensured in a two-node configuration that at least two masters will occupy the same node, which can lead to a possible outage for OpenShift if that specific node becomes unavailable. Therefore, it is a Red Hat best practice that at least three RHV-H hypervisor nodes be deployed as part of the solution so that the OpenShift masters can be distributed evenly, and the solution receives an added degree of fault tolerance.

### **Configure virtual machine/host affinity**

You can distribute the OpenShift masters across multiple hypervisor nodes by enabling VM/host affinity.

Affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity.

The conditions defined for the parameters can be either hard enforcement or soft enforcement. Hard enforcement ensures that the VMs in an affinity group always follows the positive/negative affinity strictly without any regards to external conditions. Soft enforcement, on the other hand, ensures that a higher preference is set out for the VMs in an affinity group to follow the positive/negative affinity whenever feasible. In a two or three hypervisor configuration as described in this document soft affinity is the recommended setting, in larger clusters hard affinity can be relied on to ensure OpenShift nodes are distributed.

To configure affinity groups, see the [Red Hat 6.11. Affinity Groups documentation](#).

### **Use a custom install file for OpenShift deployment**

IPI makes the deployment of OpenShift clusters extremely easy through the interactive wizard discussed earlier in this document. However, it is possible that there are some default values that might need to be changed as a part of a cluster deployment.

In these instances, the wizard can be run and tasked without immediately deploying a cluster, but instead outputting a configuration file from which the cluster can be deployed later. This is very useful if any IPI defaults need to be changed, or if a user wants to deploy multiple identical clusters in their environment for other uses

such as multitenancy. For more information about creating a customized install configuration for OpenShift, see [Red Hat OpenShift Installing a Cluster on RHV with Customizations](#).

Next: NetApp Storage Overview.

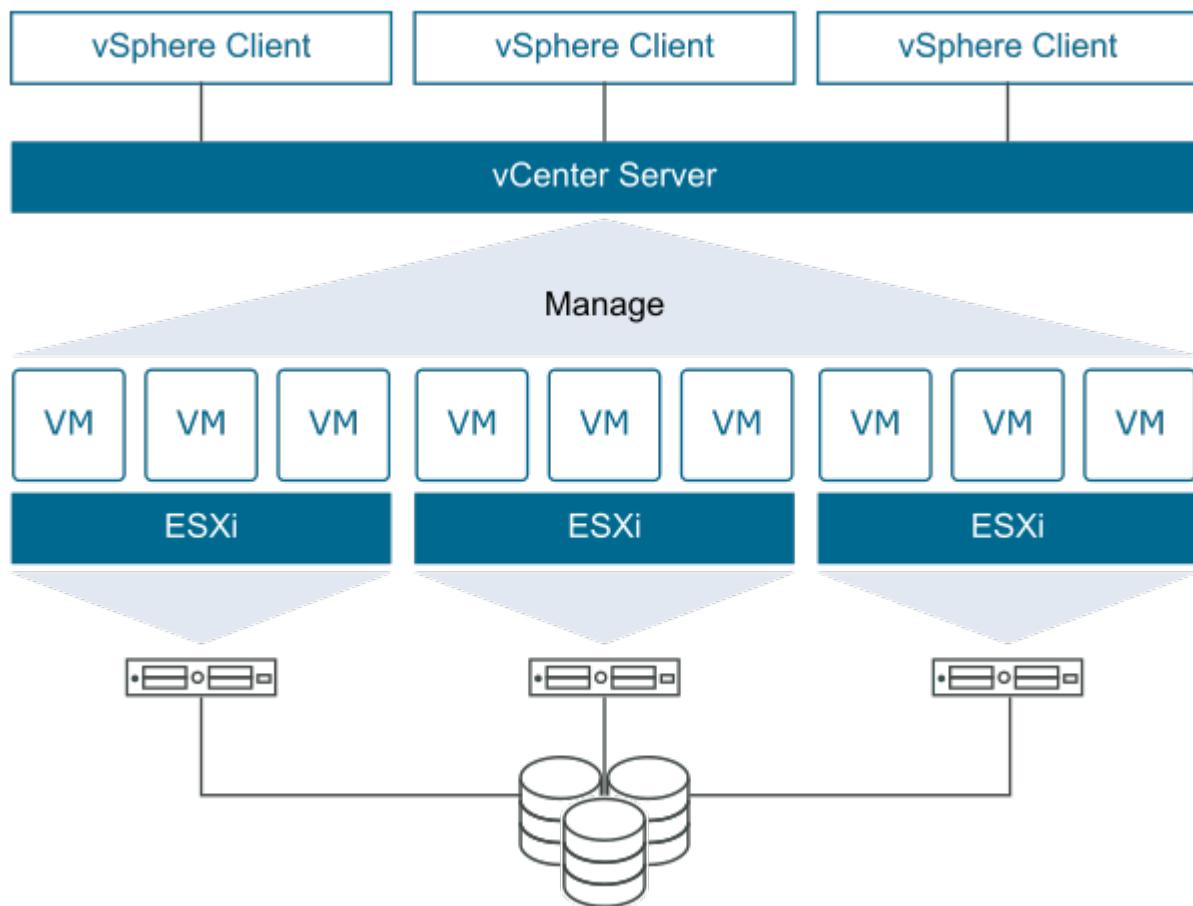
## OpenShift on VMware vSphere: Red Hat OpenShift with NetApp

VMware vSphere is a virtualization platform for centrally managing a large number of virtualized servers and networks running on the ESXi hypervisor.

For more information about VMware vSphere, see the [VMware vSphere website](#).

VMware vSphere provides the following features:

- **VMware vCenter Server** VMware vCenter Server provides unified management of all hosts and VMs from a single console and aggregates performance monitoring of clusters, hosts, and VMs.
- **VMware vSphere vMotion** VMware vCenter provides for the ability to hot migrate virtual machines between nodes in the cluster, upon user request in non-disruptive manner.
- **vSphere High Availability.** In event of host failures, to avoid disruption, VMware vSphere allows hosts to be clustered and configured for High Availability. VMs that are disrupted by host failure are rebooted shortly on other hosts in the cluster, restoring services.
- **Distributed Resource Scheduler (DRS)** A VMware vSphere cluster can be configured to load balance the resource needs of the VM's it is hosting. Virtual machines with resource contentions can be hot migrated to other nodes in the cluster to ensure that there are enough resources available.



## Network design

The Red Hat OpenShift on NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality. OCP uses the Virtual Machine logical network on VMware vSphere for its cluster management. This section describes the arrangement and purpose of each virtual network segment used in the solution and outlines the pre-requisites for deployment of the solution.

## VLAN requirements

Red Hat OpenShift on VMware vSphere is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). This configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution while validating the solution at NetApp.

VLANs	Purpose	VLAN ID
Out-of-band Management Network	Management for physical nodes and IPMI	16
VM Network	Virtual Guest Network Access	181
Storage Network	Storage network for ONTAP NFS	184
Storage Network	Storage network for ONTAP iSCSI	185
In-band Management Network	Management for ESXi Nodes, VCenter Server, ONTAP Select	3480
Storage Network	Storage network for NetApp Element iSCSI	3481
Migration Network	Network for virtual guest migration	3482

## Network infrastructure support resources

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform:

- At least one DNS server which provides a full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.

## Best practices for production deployments

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

## Deploy OpenShift to an ESXi cluster of at least three nodes

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying two ESXi hypervisor nodes and ensuring a fault tolerant configuration by enabling VMware vSphere HA and VMware vMotion, allowing deployed VMs to migrate between the two hypervisors and reboot should one host become unavailable.

Because Red Hat OpenShift initially deploys with three master nodes, it is ensured in a two-node configuration that at least two masters will occupy the same node, which can lead to a possible outage for OpenShift if that specific node becomes unavailable. Therefore, it is a Red Hat best practice that at least three ESXi hypervisor nodes be deployed as part of the solution so that the OpenShift masters can be distributed evenly, and the solution receives an added degree of fault tolerance.

### Configure virtual machine and host affinity

Ensuring the distribution of the OpenShift masters across multiple hypervisor nodes can be achieved by enabling VM/host affinity.

Affinity or Anti-Affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity.

To configure affinity groups, see the [vSphere 6.7 Documentation: Using DRS Affinity Rules](#).

### Use a custom install file for OpenShift deployment

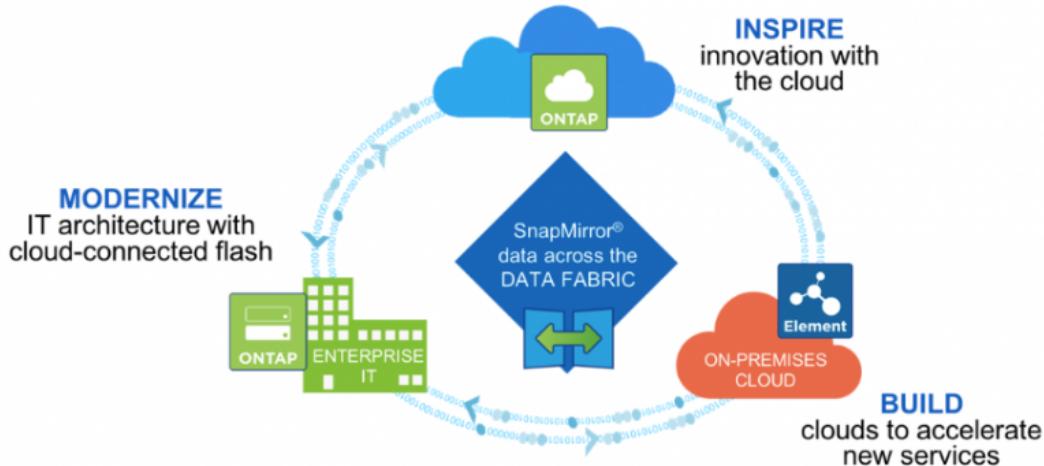
IPI makes the deployment of OpenShift clusters extremely easy through the interactive wizard discussed earlier in this document. However, it is possible that there are some default values that might need to be changed as a part of a cluster deployment.

In these instances, the wizard can be run and tasked without immediately deploying a cluster, but instead outputting a configuration file from which the cluster can be deployed later. This is very useful if any IPI defaults need to be changed, or if a user wants to deploy multiple identical clusters in their environment for other uses such as multitenancy. For more information about creating a customized install configuration for OpenShift, see [Red Hat OpenShift Installing a Cluster on vSphere with Customizations](#).

Next: [NetApp Storage Overview](#).

## NetApp Storage Overview: Red Hat OpenShift with NetApp

NetApp has several storage platforms that are qualified with our Astra Trident Storage Orchestrator to provision storage for applications deployed on Red Hat OpenShift.



- AFF and FAS systems run NetApp ONTAP, and provide storage for both file-based (NFS) and block-based (iSCSI) use cases.
- Cloud Volumes ONTAP and ONTAP-Select provide the same benefits in the cloud and virtual space respectively.
- NetApp Cloud Volumes Service (AWS/GCP) and Azure NetApp Files provide file-based storage in the cloud.
- NetApp Element storage systems provide for block-based (iSCSI) use cases in a highly scalable environment.



Each storage system in the Netapp portfolio can ease both data management and movement between on-premises sites and the cloud, ensuring that your data is where your applications are.

The following pages have additional information about the NetApp storage systems validated in the Red Hat OpenShift with NetApp solution:

- [NetApp ONTAP](#)
- [NetApp Element](#)

Next: [NetApp Storage Integrations Overview](#)

## NetApp ONTAP: Red Hat OpenShift with NetApp

NetApp ONTAP is a powerful storage-software tool with capabilities such as an intuitive GUI, REST APIs with automation integration, AI-informed predictive analytics and corrective action, non-disruptive hardware upgrades, and cross-storage import.

For more information about the NetApp ONTAP storage system, visit the [NetApp ONTAP website](#).

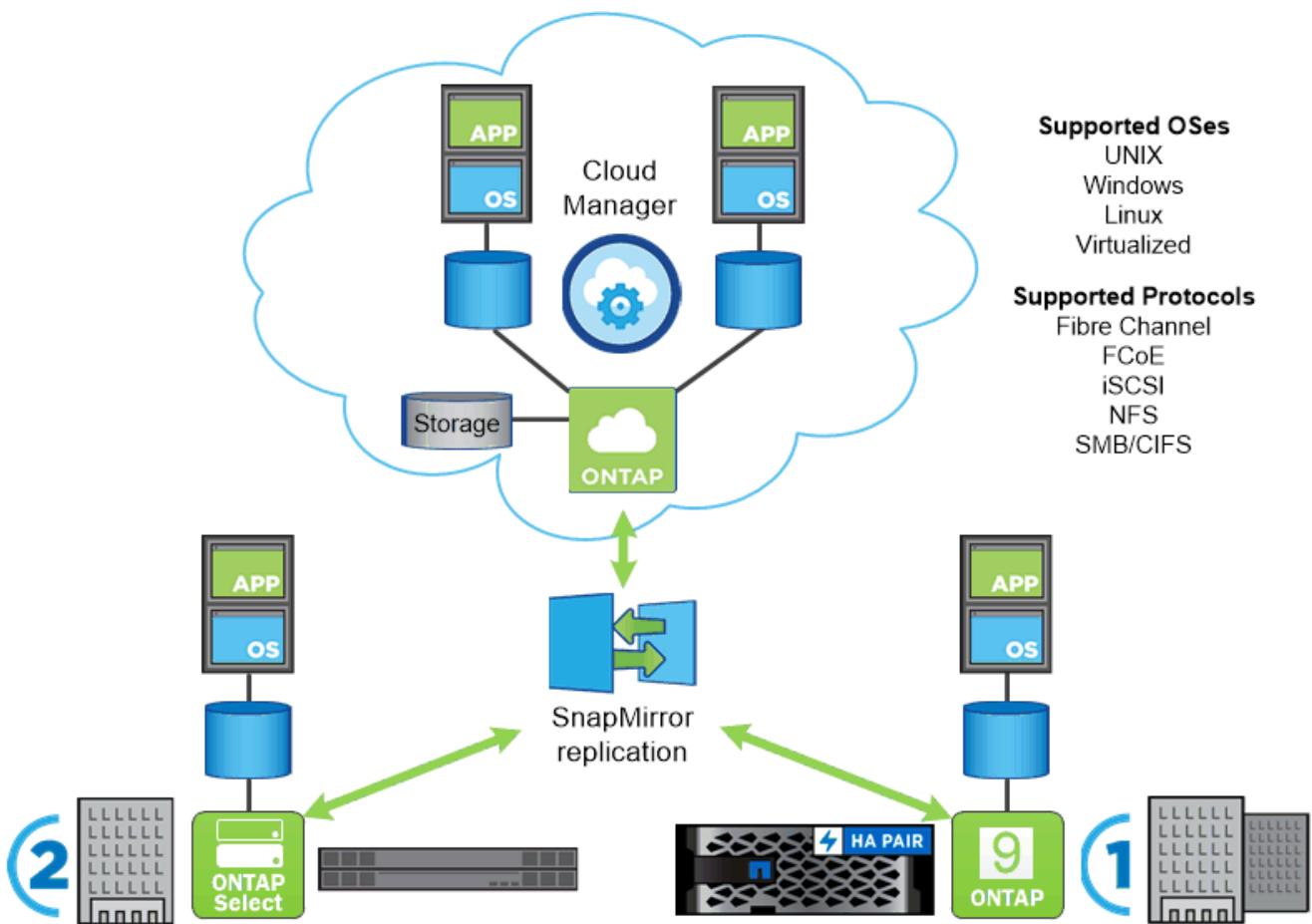
ONTAP provides the following features:

- A unified storage system with simultaneous data access and management of NFS, CIFS, iSCSI, FC, FCoE, and FC-NVMe protocols.
- Different deployment models include on-premises on all-flash, hybrid, and all-HDD hardware configurations; VM-based storage platforms on a supported hypervisor such as ONTAP Select; and in the cloud as Cloud Volumes ONTAP.
- Increased data storage efficiency on ONTAP systems with support for automatic data tiering, inline data compression, deduplication, and compaction.
- Workload-based, QoS-controlled storage.
- Seamless integration with a public cloud for tiering and protection of data. ONTAP also provides robust data protection capabilities that sets it apart in any environment:
  - **NetApp Snapshot copies.** A fast, point-in-time backup of data using a minimal amount of disk space with no additional performance overhead.
  - **NetApp SnapMirror.** Mirrors the Snapshot copies of data from one storage system to another. ONTAP supports mirroring data to other physical platforms and cloud-native services as well.
  - **NetApp SnapLock.** Efficiently administration of non-rewritable data by writing it to special volumes that cannot be overwritten or erased for a designated period.
  - **NetApp SnapVault.** Backs up data from multiple storage systems to a central Snapshot copy that serves as a backup to all designated systems.
  - **NetApp SyncMirror.** Provides real-time, RAID-level mirroring of data to two different plexes of disks that are connected physically to the same controller.
  - **NetApp SnapRestore.** Provides fast restoration of backed-up data on demand from Snapshot copies.
  - **NetApp FlexClone.** Provides instantaneous provisioning of a fully readable and writeable copy of a NetApp volume based on a Snapshot copy.

For more information about ONTAP, see the [ONTAP 9 Documentation Center](#).



NetApp ONTAP is available on-premises, virtualized, or in the cloud.



## NetApp platforms

### NetApp AFF/FAS

NetApp provides robust all-flash(AFF) and scale-out hybrid(FAS) storage platforms that are tailor-made with low-latency performance, integrated data protection, and multi-protocol support.

Both systems are powered by NetApp ONTAP data management software, the industry's most advanced data-management software for highly-available, cloud-integrated, simplified storage management to deliver enterprise-class speed, efficiency, and security your data fabric needs.

For more information about NETAPP AFF/FAS platforms, click [here](#).

### ONTAP Select

ONTAP Select is a software-defined deployment of NetApp ONTAP that can be deployed onto a hypervisor in your environment. It can be installed on VMware vSphere or on KVM and provides the full functionality and experience of a hardware-based ONTAP system.

For more information about ONTAP Select, click [here](#).

### Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP is a cloud-deployed version of NetApp ONTAP available to be deployed in a number of public clouds, including: Amazon AWS, Microsoft Azure, and Google Cloud.

For more information about Cloud Volumes ONTAP, click [here](#).

Next: [NetApp Storage Integrations Overview](#)

## NetApp Element: Red Hat OpenShift with NetApp

NetApp Element software provides modular, scalable performance, with each storage node delivering guaranteed capacity and throughput to the environment. NetApp Element systems can scale from 4 to 100 nodes in a single cluster, and offer a number of advanced storage management features.



For more information about NetApp Element storage systems, visit the [NetApp Solidfire website](#).

### iSCSI login redirection and self-healing capabilities

NetApp Element software leverages the iSCSI storage protocol, a standard way to encapsulate SCSI commands on a traditional TCP/IP network. When SCSI standards change or when the performance of Ethernet networks improves, the iSCSI storage protocol benefits without the need for any changes.

Although all storage nodes have a management IP and a storage IP, NetApp Element software advertises a single storage virtual IP address (SVIP address) for all storage traffic in the cluster. As a part of the iSCSI login process, storage can respond that the target volume has been moved to a different address and therefore it cannot proceed with the negotiation process. The host then reissues the login request to the new address in a process that requires no host-side reconfiguration. This process is known as iSCSI login redirection.

iSCSI login redirection is a key part of the NetApp Element software cluster. When a host login request is received, the node decides which member of the cluster should handle the traffic based on the IOPS and the capacity requirements for the volume. Volumes are distributed across the NetApp Element software cluster and are redistributed if a single node is handling too much traffic for its volumes or if a new node is added. Multiple copies of a given volume are allocated across the array.

In this manner, if a node failure is followed by volume redistribution, there is no effect on host connectivity beyond a logout and login with redirection to the new location. With iSCSI login redirection, a NetApp Element software cluster is a self-healing, scale-out architecture that is capable of non-disruptive upgrades and operations.

### NetApp Element software cluster QoS

A NetApp Element software cluster allows QoS to be dynamically configured on a per-volume basis. You can use per-volume QoS settings to control storage performance based on SLAs that you define. The following three configurable parameters define the QoS:

- **Minimum IOPS.** The minimum number of sustained IOPS that the NetApp Element software cluster provides to a volume. The minimum IOPS configured for a volume is the guaranteed level of performance for a volume. Per-volume performance does not drop below this level.

- **Maximum IOPS.** The maximum number of sustained IOPS that the NetApp Element software cluster provides to a particular volume.
- **Burst IOPS.** The maximum number of IOPS allowed in a short burst scenario. The burst duration setting is configurable, with a default of 1 minute. If a volume has been running below the maximum IOPS level, burst credits are accumulated. When performance levels become very high and are pushed, short bursts of IOPS beyond the maximum IOPS are allowed on the volume.

## Multitenancy

Secure multitenancy is achieved with the following features:

- **Secure authentication.** The Challenge-Handshake Authentication Protocol (CHAP) is used for secure volume access. The Lightweight Directory Access Protocol (LDAP) is used for secure access to the cluster for management and reporting.
- **Volume access groups (VAGs).** Optionally, VAGs can be used in lieu of authentication, mapping any number of iSCSI initiator-specific iSCSI Qualified Names (IQNs) to one or more volumes. To access a volume in a VAG, the initiator's IQN must be in the allowed IQN list for the group of volumes.
- **Tenant virtual LANs (VLANs).** At the network level, end-to-end network security between iSCSI initiators and the NetApp Element software cluster is facilitated by using VLANs. For any VLAN that is created to isolate a workload or a tenant, NetApp Element Software creates a separate iSCSI target SVIP address that is accessible only through the specific VLAN.
- **VRF-enabled VLANs.** To further support security and scalability in the data center, NetApp Element software allows you to enable any tenant VLAN for VRF-like functionality. This feature adds these two key capabilities:
  - **L3 routing to a tenant SVIP address.** This feature allows you to situate iSCSI initiators on a separate network or VLAN from that of the NetApp Element software cluster.
  - **Overlapping or duplicate IP subnets.** This feature enables you to add a template to tenant environments, allowing each respective tenant VLAN to be assigned IP addresses from the same IP subnet. This capability can be useful for in-service provider environments where scale and preservation of IPspace are important.

## Enterprise storage efficiencies

The NetApp Element software cluster increases overall storage efficiency and performance. The following features are performed inline, are always on, and require no manual configuration by the user:

- **Deduplication.** The system only stores unique 4K blocks. Any duplicate 4K blocks are automatically associated to an already stored version of the data. Data is on block drives and is mirrored by using the NetApp Element software Helix data protection. This system significantly reduces capacity consumption and write operations within the system.
- **Compression.** Compression is performed inline before data is written to NVRAM. Data is compressed, stored in 4K blocks, and remains compressed in the system. This compression significantly reduces capacity consumption, write operations, and bandwidth consumption across the cluster.
- **Thin-provisioning.** This capability provides the right amount of storage at the time that you need it, eliminating capacity consumption that caused by overprovisioned volumes or underutilized volumes.
- **Helix.** The metadata for an individual volume is stored on a metadata drive and is replicated to a secondary metadata drive for redundancy.

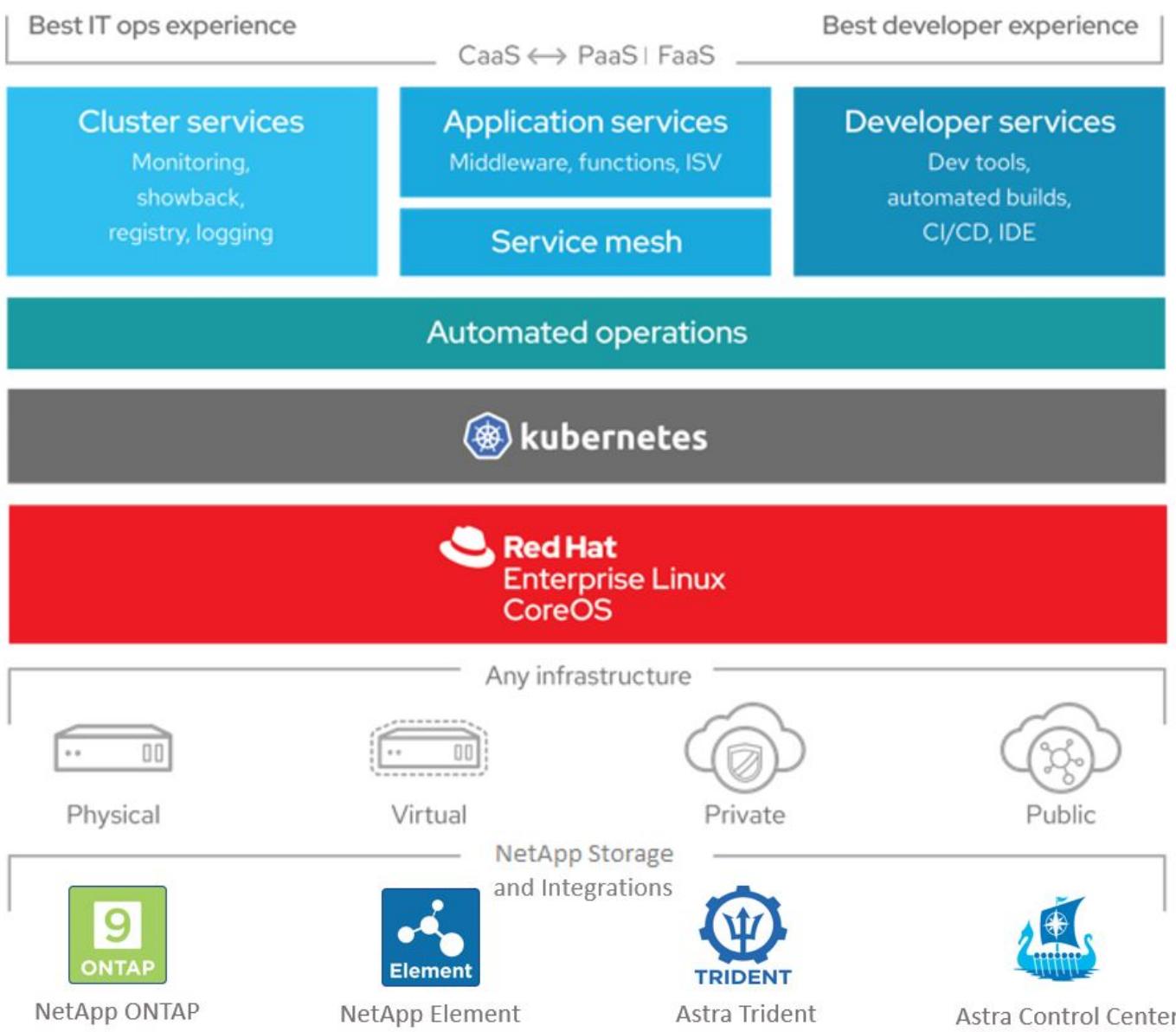


Element was designed for automation. All the storage features are available through APIs. These APIs are the only method that the UI uses to control the system.

Next: NetApp Storage Integrations Overview

## NetApp Storage Integration Overview: Red Hat OpenShift with NetApp

NetApp provides a number of products which assist our customers with orchestrating and managing persistent data in container based environments, like Red Hat OpenShift.



NetApp Astra Control offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, powered by NetApp's trusted data protection technology.

Astra Control Service is available to support stateful workloads in cloud-native Kubernetes deployments.

Astra Control Center is available to support stateful workloads in on-premises deployments, like Red Hat OpenShift.

For more information visit the NetApp Astra Control website [here](#).

NetApp Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift.

For more information visit the Astra Trident website [here](#).

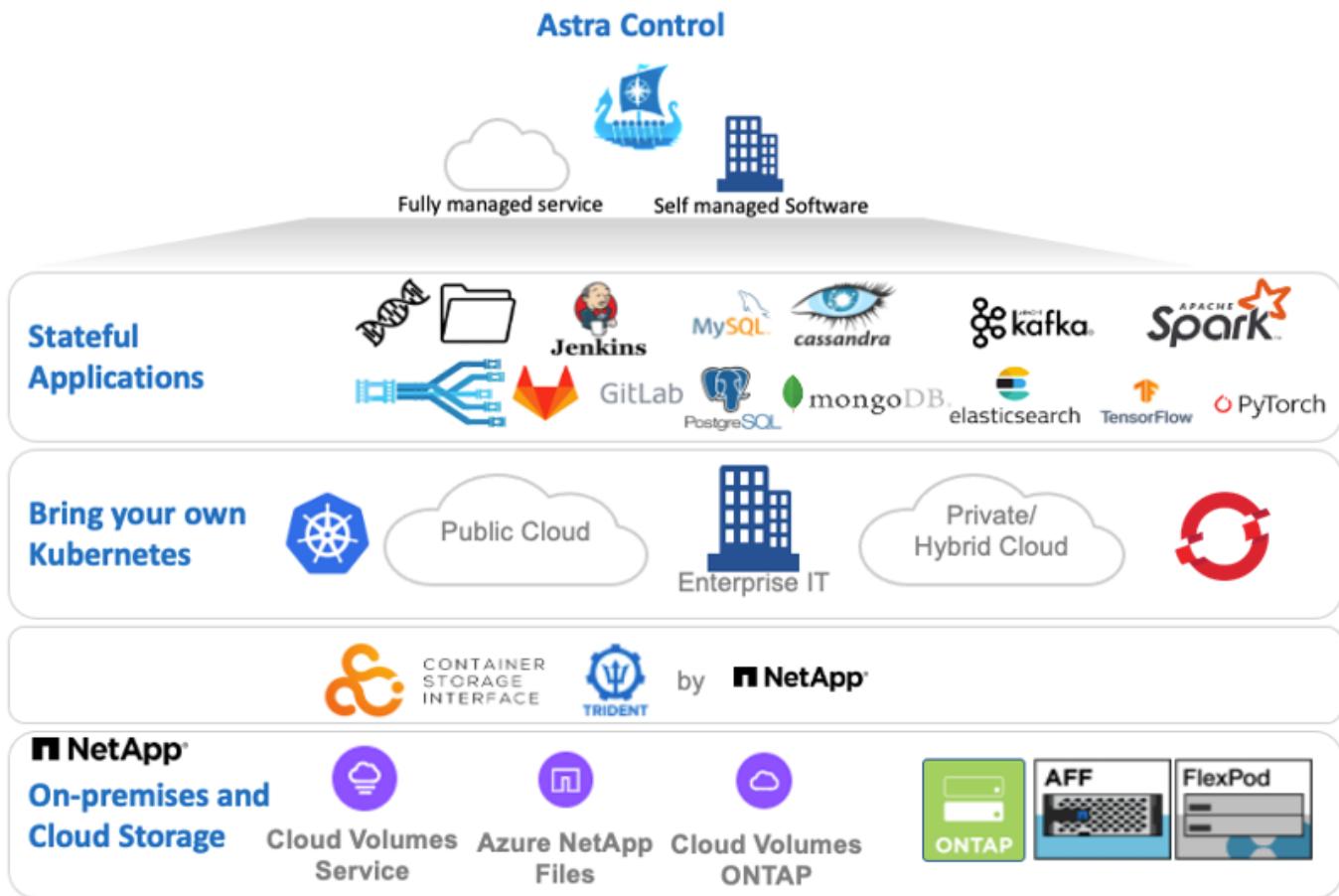
The following pages have additional information about the NetApp products that have been validated for application and persistent storage management in the Red Hat OpenShift with NetApp solution:

- [NetApp Astra Control Center](#)
- [NetApp Astra Trident](#)

Next: [NetApp Astra Control Center Overview](#)

## NetApp Astra Control Center Overview: Red Hat OpenShift with NetApp

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, deployed in an on-premises environment, powered by NetApp's trusted data protection technology.



NetApp Astra Control Center can be installed to a Red Hat OpenShift cluster which has Astra Trident storage orchestrator deployed, configured with storage classes, and storage backends to NetApp ONTAP storage systems.

For the installation and configuration of Astra Trident to support Astra Control Center, see the section in this document, [here](#).

In a cloud-connected environment, Astra Control Center uses Cloud Insights to provide advanced monitoring and telemetry. In the absence of a Cloud Insights connection, limited (7-days worth of metrics) monitoring and telemetry will be available and also exported to Kubernetes native monitoring tools (Prometheus/Grafana)

through open metrics endpoints.

Astra Control Center is fully integrated into the NetApp AutoSupport/Active IQ ecosystem to provide support for users, provide assistance with troubleshooting, and display usage statistics.

In addition to the paid version of Astra Control Center, a 90-day evaluation license will be made available to customers. The evaluation version will be supported through the email and community (Slack channel). Customers have access to these and other knowledge-base articles and the documentation available from the in-product support dashboard.

To get started with NetApp Astra Control Center, visit <https://cloud.netapp.com/astra>.

#### **Astra Control Center Installation Prerequisites**

1. One or more Red Hat OpenShift clusters.



Currently versions 4.6 EUS and 4.7 are supported.

2. Astra Trident must be already installed and configured on each Red Hat OpenShift cluster.

3. One or more NetApp ONTAP storage systems running ONTAP 9.5 or greater.



It's best practice for each OpenShift install at a site to have a dedicated SVM for persistent storage. Multi-site deployments would require additional storage systems.

4. A Trident storage backend must be configured on each OpenShift cluster with an SVM backed by an ONTAP cluster.

5. A default StorageClass configured on each OpenShift cluster with Astra Trident as the storage provisioner.

6. A load balancer must be installed and configured on each OpenShift cluster for load balancing and exposing OpenShift Services.



Refer to the link [here](#) for information about load balancers that have been validated for this purpose.

7. A private image registry must be configured to host the NetApp Astra Control Center images.



Refer the link [here](#) to install and configure an OpenShift private registry for this purpose.

8. You must have Cluster-Admin access to the Red Hat OpenShift cluster.

9. You must have Admin access to NetApp ONTAP clusters.

10. An admin workstation with docker or podman, tridentctl, and oc or kubectl tools installed and added to your \$PATH.



Docker or Podman installations must be at least version 20.10.

#### **Install Astra Control Center**

1. Log into NetApp Support Site and download the latest version of NetApp Astra Control Center. This requires a license to be attached to your NetApp account. Once you download the tarball, transfer it to the admin workstation.



To get started with a trial license for Astra Control, visit the site [here](#)

2. Unpack the tar ball and change the working directory to the resulting folder.

```
[netapp-user@rhel7 ~]$ tar -vxzf astra-control-center-21.08.57.tar.gz  
[netapp-user@rhel7 ~]$ cd astra-control-center-21.08.57
```

3. Before we start the installation, we need to push the Astra Control Center images to an image registry.



You can choose to do this with either Docker or Podman, instructions for both are provided in this step.

## podman

- Export the registry FQDN with the organization/namespace/project name as a environment variable 'registry'.

```
[netapp-user@rhel7 ~]$ export registry=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

- Log into the registry.

```
[netapp-user@rhel7 ~]$ podman login -u ocp-user -p password --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com
```

- Create a shell script file and paste the below content in it.

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar); do
    astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image(s): //')
    podman tag $astraImage $registry/$(echo $astraImage | sed
's/^[\^\/]\+\//')
    podman push $registry/$(echo $astraImage | sed 's/^[\^\/]\+\//')
done
```



If you are using untrusted certificates for your registry, edit the shell script and use `--tls-verify=false` for the podman push command - `podman push $registry/$(echo $astraImage | sed 's/[\^/]'\+\//) --tls-verify=false`

- Make the file executable.

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

- Execute the shell script.

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

## docker

- Export the registry FQDN with the organization/namespace/project name as a environment variable 'registry'.

```
[netapp-user@rhel7 ~]$ export registry=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

- b. Log into the registry.

```
[netapp-user@rhel7 ~]$ docker login -u ocp-user -p password --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com
```

- c. Create a shell script file and paste the below content in it.

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar); do
    astraImage=$(docker load --input ${astraImageFile} | sed
's/Loaded image: //')
    docker tag $astraImage $registry/$(echo $astraImage | sed
's/^[\^\/]\+\//')
    docker push $registry/$(echo $astraImage | sed 's/^[\^\/]\+\//')
done
```



If you are using untrusted certificates for your registry, edit the shell script and use --tls-verify=false for the docker push command - docker push \$registry/\$(echo \$astraImage | sed 's/[\^\/]\+\//') --tls-verify=false

- d. Make the file executable.

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

- e. Execute the shell script.

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

4. The next step is to upload the image registry TLS certificates to the OpenShift nodes. For that, create a configmap in openshift-config namespace using the TLS certificates and patch it to the cluster image config to make the certificate trusted.

```
[netapp-user@rhel7 ~]$ oc create configmap default-ingress-ca -n openshift-config --from-file=astra-registry.apps.ocp -vmw.cie.netapp.com=tls.crt
```

```
[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster --patch '{"spec":{"additionalTrustedCA":{"name":"default-ingress-ca"}}}' --type=merge
```



If you are using OpenShift internal registry with default TLS certificates from the ingress operator with a route, you will still need to follow the above step to patch the certificates to the route hostname. To extract the certificates from ingress operator, you can use the command - `oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator`

5. Create a namespace `acc-operator-system` for installing the Astra Control Center Operator.

```
[netapp-user@rhel7 ~]$ oc create ns acc-operator-system
```

6. Create a secret with credentials to log into the image registry in `acc-operator-system` namespace.

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-cred --docker-server=astra-registry.apps.ocp-vmw.cie.netapp.com --docker-username=ocp-user --docker-password=password -n acc-operator-system secret/astra-registry-cred created
```

7. Edit the Astra Control Center Operator CR `astra_control_center_operator_deploy.yaml` which is a set of all resources Astra Control Center deploys. In the operator CR, find the deployment definition for `acc-operator-controller-manager` and enter the FQDN for your registry along with the organization name as it was given while pushing the images to registry (in this example, `astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra`) by replacing the text `[your.registry.goes.here]` and provide the name of the secret we just created. Verify other details of the operator, save and close.

```
[netapp-user@rhel7 ~]$ vim astra_control_center_operator_deploy.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: acc-operator-system
spec:
  replicas: 1
  selector:
```

```
matchLabels:
  control-plane: controller-manager
template:
  metadata:
    labels:
      control-plane: controller-manager
spec:
  containers:
    - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
      image: astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-
astra/kube-rbac-proxy:v0.5.0
      name: kube-rbac-proxy
      ports:
        - containerPort: 8443
          name: https
    - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
      command:
        - /manager
      env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
      image: astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-
astra/acc-operator:21.05.68
      imagePullPolicy: IfNotPresent
      livenessProbe:
        httpGet:
          path: /healthz
          port: 8081
        initialDelaySeconds: 15
        periodSeconds: 20
      name: manager
      readinessProbe:
        httpGet:
          path: /readyz
          port: 8081
        initialDelaySeconds: 5
        periodSeconds: 10
      resources:
        limits:
```

```
cpu: 100m
memory: 150Mi
requests:
  cpu: 100m
  memory: 50Mi
securityContext:
  allowPrivilegeEscalation: false
imagePullSecrets: [name: astra-registry-cred]
securityContext:
  runAsUser: 65532
terminationGracePeriodSeconds: 10
```

8. Create the operator by running the following command.

```
[netapp-user@rhel7 ~]$ oc create -f
astra_control_center_operator_deploy.yaml
```

9. Create a dedicated namespace for installing all the Astra Control Center resources.

```
[netapp-user@rhel7 ~]$ oc create ns netapp-astra-cc
namespace/netapp-astra-cc created
```

10. Create the secret for accessing image registry in that namespace.

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-
cred --docker-server= astra-registry.apps.ocp-vmw.cie.netapp.com
--docker-username=ocp-user --docker-password=password -n netapp-astra-cc
secret/astra-registry-cred created
```

11. Next step is to edit the Astra Control Center CRD file [astra\\_control\\_center\\_min.yaml](#) and fill the FQDN, image registry details, administrator email address and other details.

```
[netapp-user@rhel7 ~]$ vim astra_control_center_min.yaml

apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  astraVersion: "21.08.57"
  astraAddress: "astra-control-center.cie.netapp.com"
  autoSupport:
    enrolled: true
  email: "solutions_tme@netapp.com"
  imageRegistry:
    name: "astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra"
  # use your registry
  secret: "astra-registry-cred"          # comment out if not
  needed
```

## 12. Create the Astra Control Center CRD in the namespace created for it.

```
[netapp-user@rhel7 ~]$ oc apply -f astra_control_center_min.yaml -n
netapp-astra-cc
astracontrolcenter.astra.netapp.io/astra created
```

 The above file `astra_control_center_min.yaml` is the minimum version of the Astra Control Center CRD. If you want to create the CRD with more control like defining storageclass other than default for creating PVCs or providing SMTP details for mail notifications, you can edit the file `astra_control_center.yaml`, fill those details and use it to create the CRD.

## Installation Verificaton

1. It might take several minutes for the installation to complete. Verify that all the pods and services in netapp-astra-cc namespace are up and running.

```
[netapp-user@rhel7 ~]$ oc get all -n netapp-astra-cc
```

2. Check the `acc-operator-controller-manager` logs to ensure that the installation is completed.

```
[netapp-user@rhel7 ~]$ oc logs deploy/acc-operator-controller-manager -n
acc-operator-system -c manager -f
```



The following message should be displayed to indicate the successful installation of Astra Control Center

```
{"level": "info", "ts": 1624054318.029971, "logger": "controllers.AstraControlCenter", "msg": "Successfully Reconciled AstraControlCenter in [seconds]s", "AstraControlCenter": "netapp-astra-cc/astra", "ae.Version": "[21.08.57]"}}
```

3. The username for logging into Astra Control Center is the email address of the administrator provided in the CRD file and the password is a string ‘ACC-’ appended to the Astra Control Center UUID. Run the following command –

```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc  
NAME      UUID  
astra     345c55a5-bf2e-21f0-84b8-b6f2bce5e95f
```

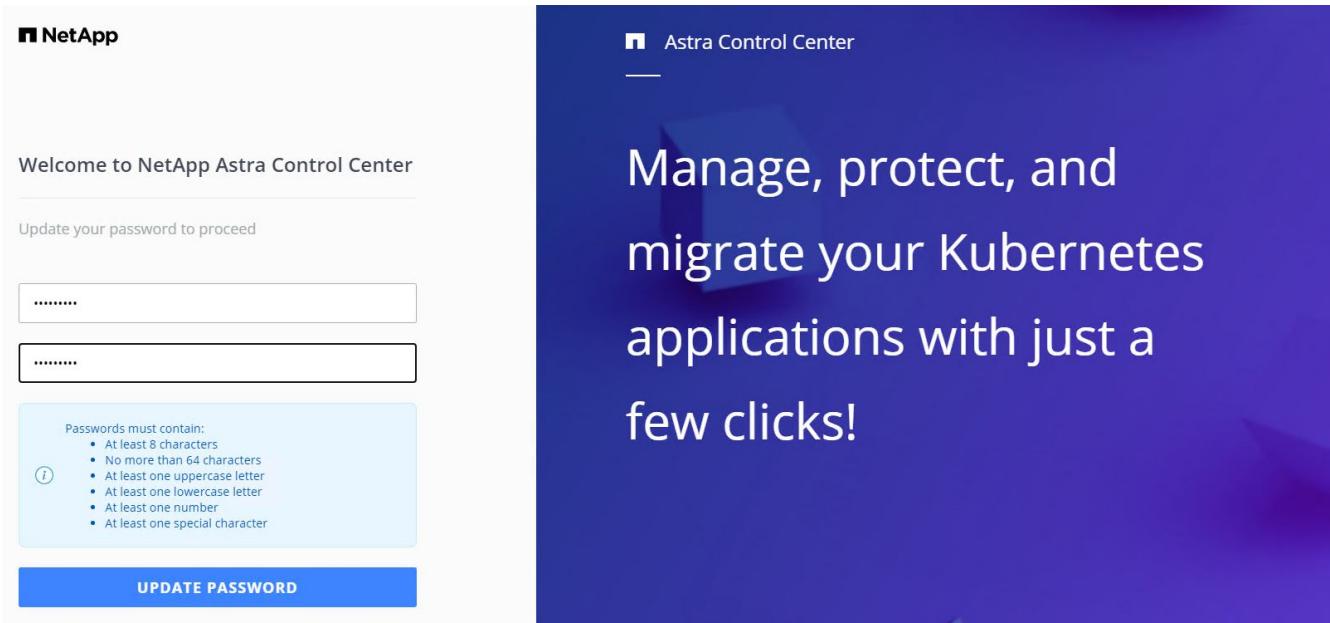


In this example, the password is – ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f

4. Now log into the Astra Control Center GUI by browsing to the FQDN you provided in the CRD file.

The screenshot shows two parts of the Astra Control Center interface. On the left, the login screen displays the NetApp logo and a form with fields for Email and Password, and a blue LOGIN button. On the right, the home page features the Astra Control Center logo and a large, bold text message: "Manage, protect, and migrate your Kubernetes applications with just a few clicks!".

5. When you log into Astra Control Center GUI for the first time using the admin email address provided in CRD, you will need to change the password.



6. If you wish to add a user to Astra Control Center, go to [Account](#) → [Users](#) and click on [Add](#) and enter the details of the user and click [Add](#).

**Add user**

**USER DETAILS**

First name Nikhil	Last name Kulkarni
Email address tme_nik@netapp.com	

**PASSWORD**

Temporary password .....	Confirm temporary password .....
-----------------------------	-------------------------------------

Passwords must contain:  

- At least 8 characters
- No more than 64 characters
- At least one lowercase letter
- At least one uppercase letter
- At least one number
- At least one special character

**USER ROLE** ?

Role Owner
---------------

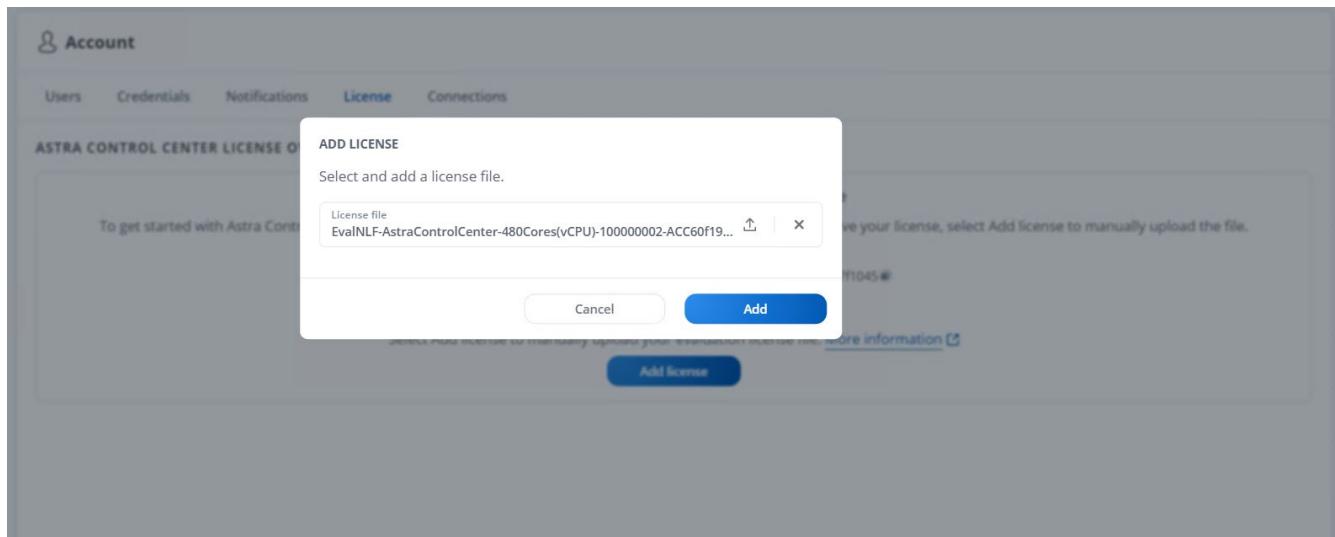
**ADD NEW USER**

Add new user

Add a new user to your Astra Control Center account. New users will be prompted to update their password the first time they log in to Astra Control Center. They will also inherit access to account-wide credentials according to their role. Read more in [users](#).

**Cancel** **Add ✓**

7. Astra Control Center requires a license for all of its functionalities to work. To add a license, go to [Account](#) → [License](#), click on [Add License](#) and upload the license file.



If you encounter issues with the install or configuration of NetApp Astra Control Center, the knowledge base of known issues is available [here](#).

Next: Register your Red Hat OpenShift Clusters: Red Hat OpenShift with NetApp.

### Register your Red Hat OpenShift Clusters with the Astra Control Center

To enable the Astra Control Center to manage your workloads, you must first register your Red Hat OpenShift cluster.

#### Register Red Hat OpenShift clusters

1. The first step is to add the OpenShift clusters to the Astra Control Center and manage them. Go to Clusters and click Add a Cluster, upload the kubeconfig file for the OpenShift cluster, and click Select Storage.

A screenshot of the 'Add cluster' wizard. The title is 'Add cluster' and the subtitle is 'STEP 1/3: CREDENTIALS'. On the left, there's a 'CREDENTIALS' section with instructions: 'Provide Astra Control access to your Kubernetes and OpenShift clusters by entering a kubeconfig credential.' It includes a link 'Follow instructions' and two buttons: 'Upload file' and 'Paste from clipboard'. A file input field shows 'Kubeconfig YAML file ocp-vmw kubeconfig.txt'. To the right, there's a 'ADDING A CLUSTER' sidebar with the text 'Adding a cluster is needed for Astra Control to discover your Kubernetes applications.', a 'Select a cloud provider and input credentials to get started.' link, and a 'Read more in Clusters' link. At the bottom, there are 'Cancel' and 'Configure storage →' buttons.



The kubeconfig file can be generated to authenticate with a username and password or a token. Tokens expire after a limited amount of time and might leave the registered cluster unreachable. NetApp recommends using a kubeconfig file with a username and password to register your OpenShift clusters to Astra Control Center.

2. Astra Control Center detects the eligible storage classes. Now select the way that storageclass provisions volumes using Trident backed by an SVM on NetApp ONTAP, and click Review. In the next pane, verify the details and click Add Cluster.

**Add cluster**      STEP 2/3: STORAGE      X

**STORAGE**

Existing storage classes are discovered and verified as eligible for use with Astra Control. You can use your existing default, or choose to set a new default at this time. Applications with persistent volumes on eligible storage classes are validated for use with Astra Control.

Set default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligible
<input checked="" type="radio"/>	ocp-trident <small>Default</small>	csi.trident.netapp.io	Delete	Immediate	<span style="color: green;">✓</span>
<input type="radio"/>	ocp-trident-iscsi	csi.trident.netapp.io	Delete	Immediate	<span style="color: green;">✓</span>
<input type="radio"/>	project-1-sc	csi.trident.netapp.io	Delete	Immediate	<span style="color: orange;">⚠</span>
<input type="radio"/>	thin	kubernetes.io/vsphere-volume	Delete	Immediate	<span style="color: orange;">⚠</span>

[← Select credentials](#)      [Review →](#)

3. Register both OpenShift clusters as described in step 1. When added, the clusters move to the Discovering status while Astra Control Center inspects them and installs the necessary agents. Cluster status changes to Running after they are successfully registered.

**admin**

**Clusters**

Name	Ready	Type	Version	Actions
ocp-vmw	<span style="color: green;">✓</span>	Red Hat OpenShift	v1.20.0+df9c838	<span style="color: green;">Running</span>
ocp-vmware2	<span style="color: green;">✓</span>	Red Hat OpenShift	v1.20.0+c8905da	<span style="color: green;">Running</span>



All Red Hat OpenShift clusters to be managed by Astra Control Center should have access to the image registry that was used for its installation as the agents installed on the managed clusters will pull the images from that registry.

4. Import ONTAP clusters as storage resources to be managed as backends by Astra Control Center. When OpenShift clusters are added to Astra and a storageclass is configured, it automatically discovers and inspects the ONTAP cluster backing the storageclass but does not import it into the Astra Control Center to be managed.

The screenshot shows the 'Backends' section of the Astra Control Center interface. On the left, there's a sidebar with 'Dashboard', 'MANAGE YOUR APPS' (Apps, Clusters), 'MANAGE YOUR STORAGE' (Backends selected, Buckets), and 'MANAGE YOUR ACCOUNT' (Account, Activity, Support). The main area has a header 'Backends' with a '+ Manage' button, a search bar, and filters for 'Managed' and 'Discovered'. It lists two entries:

Name	Status	Capacity	Type	Actions
172.21.224.201(ontapsan_10.61.181.243)	⚠️	Not available yet	ONTAP	Discovered
172.21.224.211(ocp-trident-replication)	⚠️	Not available yet	ONTAP	Discovered

5. To import the ONTAP clusters, go to Backends, and click the dropdown, and select Manage next to the ONTAP cluster to be managed. Enter the ONTAP cluster credentials, click Review Information, and then click Import Storage Backend.

The screenshot shows the 'Manage ONTAP storage backend' dialog. It's titled 'STEP 1/2: CREDENTIALS'. The left panel has a 'CREDENTIALS' section with a note: 'Enter cluster administrator credentials for the ONTAP storage backend you want to manage.' It includes fields for 'Cluster management IP address' (172.21.224.201), 'User name' (admin), and 'Password' (redacted). The right panel has a 'MANAGE STORAGE BACKEND' section with a note: 'Storage backends provide storage to your Kubernetes applications.' It explains that managing storage clusters in Astra Control as a storage backend will allow linkages between PVs and the storage backend, and details performance metrics if Cloud Insights are connected. There's a 'Read more in Storage backend' link. At the bottom are 'Cancel' and 'Review information →' buttons.

6. After the backends are added, the status changes to Available. These backends now have the information about the persistent volumes in the OpenShift cluster and the corresponding volumes on the ONTAP system.

The screenshot shows the Astra Control Center's 'Backends' page. On the left sidebar, under 'MANAGE YOUR STORAGE', 'Backends' is selected. The main table displays two entries:

Name	Status	Capacity	Type	Actions
K8s-OnTap	✓	0.11/1.07 TiB: 9.9%	ONTAP 9.8.0	<span>Available</span>
ONTAP-Select-02	✓	0.07/2.07 TiB: 3.3%	ONTAP 9.8.0	<span>Available</span>

- For backup and restore across OpenShift clusters using Astra Control Center, you must provision an object storage bucket that supports the S3 protocol. Currently supported options are ONTAP S3, StorageGRID, and AWS S3. For the purpose of this installation, we are going to configure an AWS S3 bucket. Go to Buckets, click Add bucket, and select Generic S3. Enter the details about the S3 bucket and credentials to access it, click the checkbox "Make this bucket the default bucket for the cloud," and then click Add.

**Add bucket**

**STORAGE BUCKET**

Enter the access details of your existing object store bucket to allow Astra Control to store your application backups.

Type Generic S3	Existing bucket name ocp-vmware2-astra-cc
Description (optional) s3.us-east-1.amazonaws.com	S3 server name or IP address s3.us-east-1.amazonaws.com

Make this bucket the default bucket for this cloud

**SELECT CREDENTIALS**

Astra Control requires S3 access credentials with the roles necessary to facilitate Kubernetes application data management.

<b>Add</b> Use existing	
Access ID AMWSTCFKDSU6HWSZXABD	Secret key .....
Credential name AWS-S3	

**ADDING STORAGE BUCKETS**

Astra Control stores backups in your existing object store buckets. The first bucket added for a selected cloud will be designated as the default bucket for backup and clone operations.

Read more in [storage buckets](#).

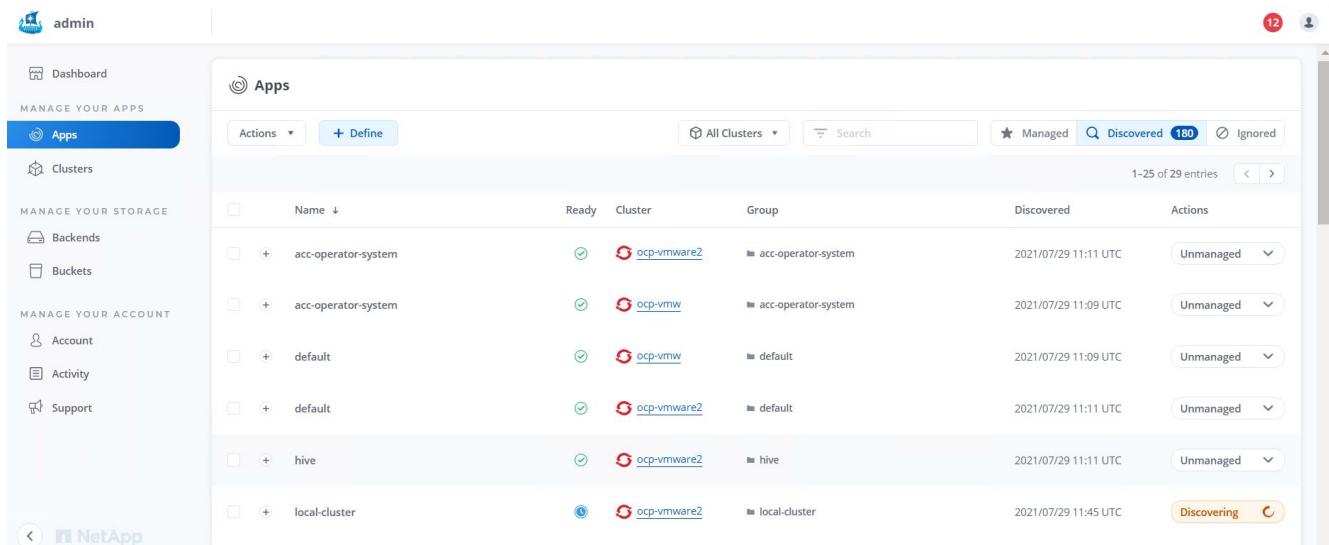
Next: Choose the Applications To Protect: Red Hat OpenShift with NetApp.

### Choose the applications to protect

After you have registered your Red Hat OpenShift clusters, you can discover the applications that are deployed and manage them via the Astra Control Center.

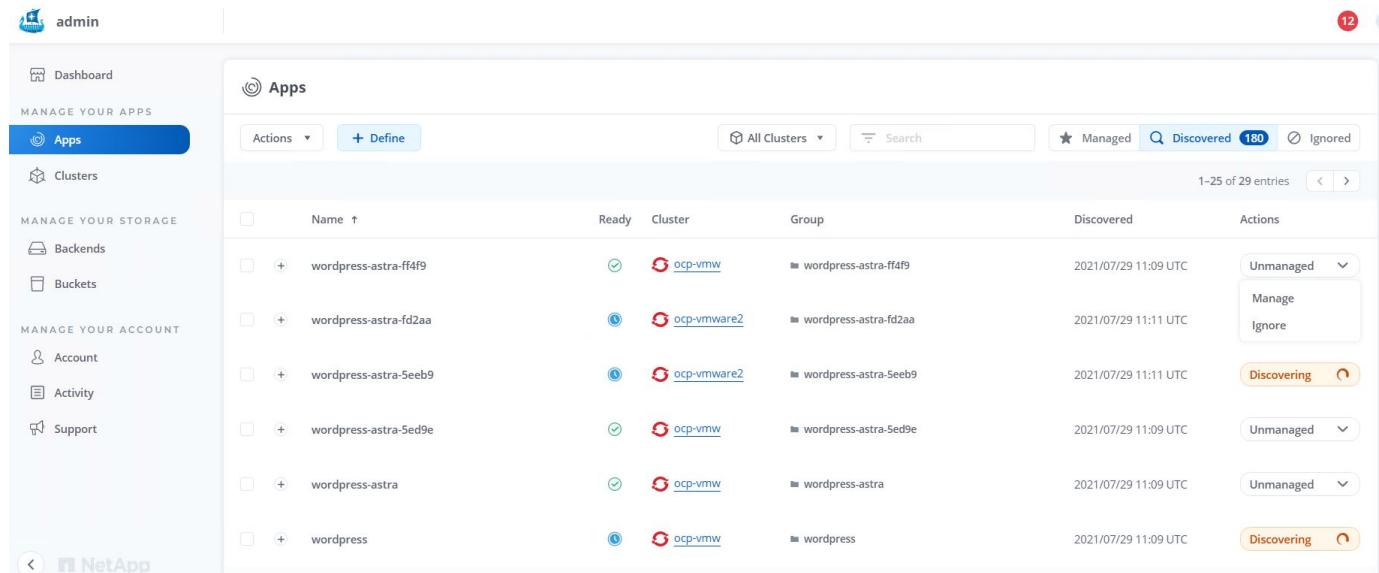
## Manage applications

- After the OpenShift clusters and ONTAP backends are registered with the Astra Control Center, the control center automatically starts discovering the applications in all the namespaces that are using the storageclass configured with the specified ONTAP backend.



The screenshot shows the Astra Control Center dashboard. On the left, there's a sidebar with links for Dashboard, Clusters, Backends, Buckets, Account, Activity, and Support. The main area is titled 'Apps' and has tabs for 'Actions', '+ Define', 'All Clusters', 'Search', 'Managed', 'Discovered' (which is selected), and 'Ignored'. It displays a table of discovered applications with columns for Name, Ready status, Cluster, Group, Discovered date, and Actions. The table shows several entries, including 'acc-operator-system', 'default', 'hive', and 'local-cluster', all associated with the 'ocp-vmware2' cluster and grouped under 'acc-operator-system', 'default', and 'hive' respectively. One entry, 'local-cluster', is currently in the 'Discovering' state.

- Navigate to Apps > Discovered and click the dropdown menu next to the application you would like to manage using Astra. Then click Manage.



This screenshot is similar to the previous one, showing the 'Discovered' tab in the Apps section. However, a context menu is open over the first application entry, 'wordpress-astra-ff4f9'. The menu options shown are 'Unmanaged', 'Manage', and 'Ignore'. The rest of the table and sidebar are identical to the first screenshot.

- The application enters the Available state and can be viewed under the Managed tab in the Apps section.

The screenshot shows the 'Apps' section of the Astra Control Center. At the top, there are buttons for 'Actions', '+ Define', 'All Clusters' (with a dropdown), 'Search', 'Managed' (with a star icon), 'Discovered' (with a count of 175), and 'Ignored'. Below this is a table header with columns: Name, Ready, Protected, Cluster, Group, Discovered, and Actions. A single row is listed: 'wordpress-astra-ff4f9' with status 'Ready' (green checkmark), 'Protected' (blue info icon), 'Cluster' 'ocp-vmw' (red shield icon), 'Group' 'wordpress-astra-ff4f9' (black bar chart icon), 'Discovered' '2021/07/29 11:09 UTC', and 'Actions' 'Available' (green button with dropdown).

[Next: Protect Your Applications: Red Hat OpenShift with NetApp.](#)

### Protect your applications

After application workloads are managed by Astra Control Center, you can configure the protection settings for those workloads.

### Creating an application snapshot

A snapshot of an application creates an ONTAP Snapshot copy that can be used to restore or clone the application to a specific point in time based on that Snapshot copy.

1. To take a snapshot of the application, navigate to the Apps > Managed tab and click the application you would like to make a Snapshot copy of. Click the dropdown menu next to the application name and click Snapshot.

The screenshot shows the application details for 'wordpress-astra-ff4f9'. On the left is a sidebar with 'Dashboard', 'MANAGE YOUR APPS' (selected 'Apps'), 'Clusters', 'MANAGE YOUR STORAGE' (selected 'Backends' and 'Buckets'). The main area shows the application card for 'wordpress-astra-ff4f9'. It includes sections for 'App status' (Healthy), 'Protection schedule' (Disabled), 'Group' (wordpress-astra-ff4f9), and 'Cluster' (ocp-vmw). On the right, there is a dropdown menu set to 'Available' with options: 'Snapshot' (selected), 'Backup', 'Clone', and 'Unmanage'. There are also notification icons (12) and user profile icons.

2. Enter the snapshot details, click Review, and then click Snapshot. It takes about a minute to create the snapshot, and the status becomes Available after the snapshot is successfully created.

**Snapshot Application**

**STEP 1/2: DETAILS**

**SNAPSHOT DETAILS**

Name  
wordpress-astra-ff4f9-snapshot-20210729120451

**OVERVIEW**

Application snapshots  
Astra Control can take a quick snapshot of your application configuration and persistent storage. Enter a snapshot name to get started.

Read more in [Protect apps](#).

**Application**  
wordpress-astra-ff4f9

**Namespace**  
wordpress-astra-ff4f9

**Cluster**  
ocp-vmw

**Cancel** **Review →**

## Creating an application backup

A backup of an application captures the active state of the application and the configuration of its resources, converts them into files, and stores them in a remote object storage bucket.

For the backup and restore of managed applications in the Astra Control Center, you must configure superuser settings for the backing ONTAP systems as a prerequisite. To do so, enter the following commands.

```
ONTAP::> export-policy rule modify -vserver ocp-trident -policyname
default -ruleindex 1 -superuser sys
ONTAP::> export-policy rule modify -policyname default -ruleindex 1 -anon
65534 -vserver ocp-trident
```

- To create a backup of the managed application in the Astra Control Center, navigate to the Apps > Managed tab and click the application that you want to take a backup of. Click the dropdown menu next to the application name and click Backup.

**admin**

**MANAGE YOUR APPS**

**Apps**

**wordpress-astra-ff4f9**

**Available**

- Snapshot
- Backup
- Clone
- Unmanage

<b>App status</b> Healthy	<b>App protection status</b> Partially Protected
<b>Protection schedule</b> Disabled	<b>Group</b> wordpress-astra-ff4f9
<b>Images</b> docker.io/bitnami/mariadb:10.5.10-debian-10-r13 docker.io/bitnami/wordpress:5.7.2-debian-10-r10	<b>Cluster</b> ocp-vmw

2. Enter the backup details, select the object storage bucket to hold the backup files, click Review, and, after reviewing the details, click Backup. Depending on the size of the application and data, the backup can take several minutes, and the status of the backup becomes Available after the backup is completed successfully.

**STEP 1/2: DETAILS**

**BACKUP DETAILS**

Name: wordpress-astra-ff4f9-backup-20210729120857

Backup from an existing snapshot

**BACKUP DESTINATION**

Bucket: ocp-vmware2-astra-cc (Default)

**OVERVIEW**

Application backups

Astra Control can take a backup of your application configuration and persistent storage. Persistent storage backups are transferred to your object store. Enter a backup name to get started.

Read more in [Application backups](#).

- Application: wordpress-astra-ff4f9
- Namespace: wordpress-astra-ff4f9
- Cluster: ocp-vmw

**Cancel** **Review →**

## Restoring or cloning an application

At the push of a button, you can restore an application to the originating cluster or clone it to a remote cluster for dev/test or application protection and disaster recovery purposes.

1. To restore or clone an application, navigate to the Apps > Managed tab and click the app in question. Click the dropdown menu next to the application name and click Clone.

**wordpress-astra-ff4f9**

**App status**: Healthy

**Protection schedule**: Disabled

**Group**: wordpress-astra-ff4f9

**Cluster**: ocp-vmw

**Available**

- Snapshot
- Backup
- Clone**
- Unmanage

2. Enter the details of the new namespace, select the cluster you want to restore or clone it to, and choose if you want to restore or clone it from an existing snapshot or from a backup of the current state of the application. Then click Review and click Clone after you have reviewed the details.

**Clone application**

**STEP 1/2: DETAILS**

**CLONE DETAILS**

Clone name wordpress-astra-ff4f9-9e4b6	Clone namespace wordpress-astra-ff4f9-9e4b6
Destination cluster ocp-vmw	<input checked="" type="checkbox"/> Clone from an existing snapshot or backup

**CLONE SOURCE**

App Backup				Ready	On-Schedule/On-Demand	Created ↑
<input checked="" type="radio"/>	wordpress-astra-ff4f9-backup-20210729120857			On-Demand	2021/07/29 11:57 UTC	

**OVERVIEW**

**Application cloning**

Astra Control can create a clone of your application configuration and persistent storage. Persistent storage backups are transferred from your object store, so choosing a clone from an existing backup will complete the fastest. Enter a clone name to get started.

Read more in [Clone apps](#).

**Cancel** **Review →**

- The new application goes to the Discovering state while Astra Control Center creates the application on the selected cluster. After all the resources of the application are installed and detected by Astra, the application goes to the Available state.

**Dashboard**

**MANAGE YOUR APPS**

**Apps** **Define**

**MANAGE YOUR STORAGE**

**Backends**

**Buckets**

**MANAGE YOUR ACCOUNT**

**Account**

**Activity**

**Support**

**Apps**

**Actions** **+ Define**

**Managed** **Discovered 179** **Ignored**

Name	Ready	Protected	Cluster	Group	Discovered	Actions
<a href="#">wordpress-astra-ff4f9</a>				wordpress-astra-ff4f9	2021/07/29 11:09 UTC	
<a href="#">wordpress-astra-ff4f9-9e4b6</a>				wordpress-astra-ff4f9-9e4b6	2021/07/29 13:24 UTC	

[Next: Solution Validation/Use Cases: Red Hat OpenShift with NetApp.](#)

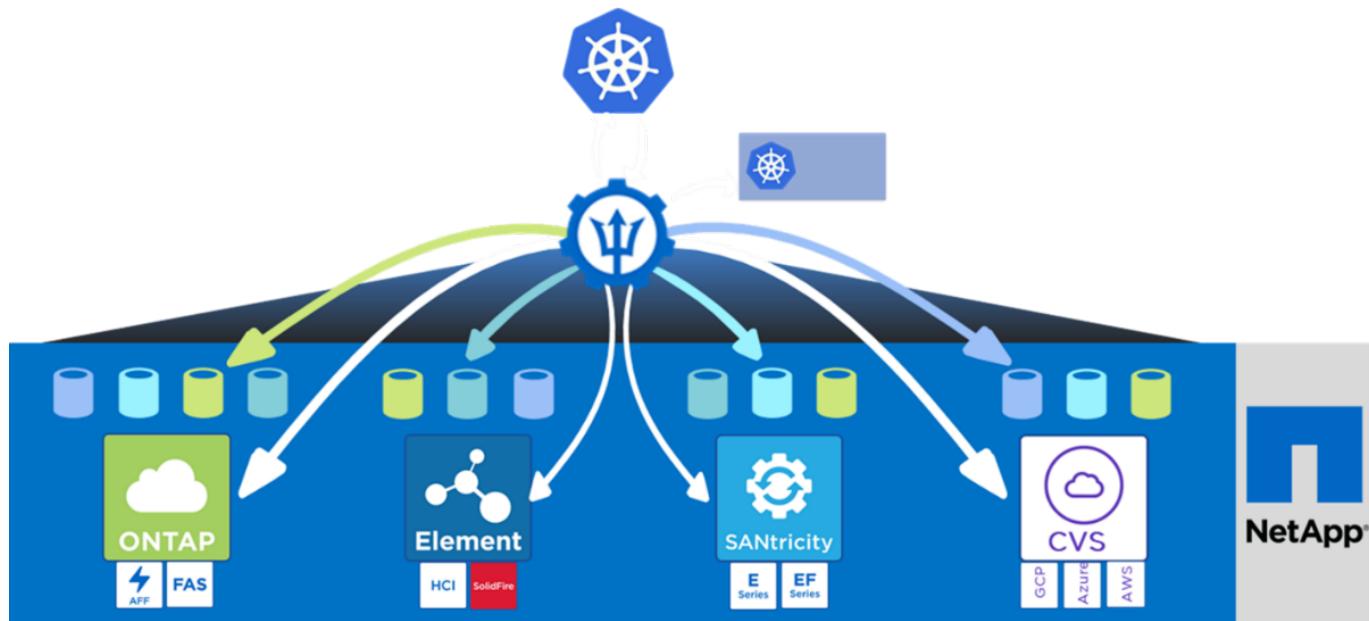
## Astra Trident Overview: Red Hat OpenShift with NetApp

Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift.

Trident works with the entire NetApp storage portfolio, including the NetApp ONTAP and Element storage systems, and it also supports NFS and iSCSI connections.

Trident accelerates the DevOps workflow by allowing end users to provision and manage storage from their NetApp storage systems without requiring intervention from a storage administrator.

An administrator can configure a number of storage backends based on project needs and storage system models that enable advanced storage features, including compression, specific disk types, or QoS levels that guarantee a certain level of performance. After they are defined, these backends can be used by developers in their projects to create persistent volume claims (PVCs) and to attach persistent storage to their containers on demand.



Astra Trident has a rapid development cycle, and just like Kubernetes, is released four times a year.

The latest version of Astra Trident is 21.04 released in April 2021. A support matrix for what version of Trident has been tested with which Kubernetes distribution can be found [here](#).

Starting with the 20.04 release, Trident setup is performed by the Trident operator. The operator makes large scale deployments easier and provides additional support including self healing for pods that are deployed as a part of the Trident install.

With the 21.01 release, a Helm chart was made available to ease the installation of the Trident Operator.

#### Download Astra Trident

To install Trident on the deployed user cluster and provision a persistent volume, complete the following steps:

1. Download the installation archive to the admin workstation and extract the contents. The current version of Trident is 21.04, which can be downloaded [here](#).

```
[netapp-user@rhel7 ~]$ wget https://github.com/NetApp/trident/releases/download/v21.04/trident-installer-21.04.tar.gz --2021-05-06 15:17:30-- https://github.com/NetApp/trident/releases/download/v21.04/trident-installer-21.04.tar.gz Resolving github.com (github.com)... 140.82.114.3 Connecting to github.com (github.com)|140.82.114.3|:443... connected. HTTP request sent, awaiting response... 302 Found
```

```

Location: https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
21.04.tar.gz&response-content-type=application%2Foctet-stream
[following]
--2021-05-06 15:17:30--  https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
21.04.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-
releases.githubusercontent.com) ... 185.199.108.154, 185.199.109.154,
185.199.110.154, ...
Connecting to github-releases.githubusercontent.com (github-
releases.githubusercontent.com)|185.199.108.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38349341 (37M) [application/octet-stream]
Saving to: 'trident-installer-21.04.tar.gz'

100%[=====] 38,349,341 88.5MB/s
in 0.4s

2021-05-06 15:17:30 (88.5 MB/s) - 'trident-installer-21.04.tar.gz' saved
[38349341/38349341]

```

## 2. Extract the Trident install from the downloaded bundle.

```

[netapp-user@rhel7 ~]$ tar -xzf trident-installer-21.04.tar.gz
[netapp-user@rhel7 ~]$ cd trident-installer/
[netapp-user@rhel7 trident-installer]$

```

## Install the Trident Operator with Helm

1. First set the location of the user cluster's `kubeconfig` file as an environment variable so that you don't have to reference it, because Trident has no option to pass this file.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-install/auth/kubeconfig
```

2. Run the Helm command to install the Trident operator from the tarball in the helm directory while creating the trident namespace in your user cluster.

```
[netapp-user@rhel7 trident-installer]$ helm install trident
helm/trident-operator-21.04.tgz --create-namespace --namespace trident
NAME: trident
LAST DEPLOYED: Fri May 7 12:54:25 2021
NAMESPACE: trident
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing trident-operator, which will deploy and manage
NetApp's Trident CSI
storage provisioner for Kubernetes.
```

Your release is named 'trident' and is installed into the 'trident' namespace.

Please note that there must be only one instance of Trident (and trident-operator) in a Kubernetes cluster.

To configure Trident to manage storage resources, you will need a copy of `tridentctl`, which is available in pre-packaged Trident releases. You may find all Trident releases and source code online at <https://github.com/NetApp/trident>.

To learn more about the release, try:

```
$ helm status trident
$ helm get all trident
```

3. You can verify that Trident is successfully installed by checking the pods that are running in the namespace or by using the `tridentctl` binary to check the installed version.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                      READY   STATUS    RESTARTS   AGE
trident-csi-5z451          1/2     Running   2          30s
trident-csi-696b685cf8-htdb2 6/6     Running   0          30s
trident-csi-b74p2          2/2     Running   0          30s
trident-csi-lrw4n          2/2     Running   0          30s
trident-operator-7c748d957-gr2gw 1/1     Running   0          36s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 21.04           | 21.04           |
+-----+-----+
```



In some cases, customer environments might require the customization of the Trident deployment. In these cases, it is also possible to manually install the Trident operator and update the included manifests to customize the deployment.

#### Manually install the Trident Operator

1. First, set the location of the user cluster's `kubeconfig` file as an environment variable so that you don't have to reference it, because Trident has no option to pass this file.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-
install/auth/kubeconfig
```

2. The `trident-installer` directory contains manifests for defining all the required resources. Using the appropriate manifests, create the `TridentOrchestrator` custom resource definition.

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.tride
nt.netapp.io created
```

3. If one does not exist, create a Trident namespace in your cluster using the provided manifest.

```
[netapp-user@rhel7 trident-installer]$ oc apply -f deploy/namespace.yaml
namespace/trident created
```

4. Create the resources required for the Trident operator deployment, such as a `ServiceAccount` for the operator, a `ClusterRole` and `ClusterRoleBinding` to the `ServiceAccount`, a dedicated `PodSecurityPolicy`, or the operator itself.

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

5. You can check the status of the operator after it's deployed with the following commands:

```
[netapp-user@rhel7 trident-installer]$ oc get deployment -n trident
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator   1/1     1           1          23s
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                           READY   STATUS    RESTARTS   AGE
trident-operator-66f48895cc-lzczk   1/1     Running   0          41s
```

6. With the operator deployed, we can now use it to install Trident. This requires creating a [TridentOrchestrator](#).

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
[netapp-user@rhel7 trident-installer]$ oc describe torc trident
Name:           trident
Namespace:
Labels:         <none>
Annotations:    <none>
API Version:   trident.netapp.io/v1
Kind:          TridentOrchestrator
Metadata:
  Creation Timestamp:  2021-05-07T17:00:28Z
  Generation:        1
  Managed Fields:
    API Version:   trident.netapp.io/v1
    Fields Type:   FieldsV1
    fieldsV1:
      f:spec:
        .:
      f:debug:
      f:namespace:
  Manager:       kubectl-create
  Operation:     Update
  Time:          2021-05-07T17:00:28Z
  API Version:   trident.netapp.io/v1
```

```
Fields Type: FieldsV1
fieldsV1:
  f:status:
    .:
  f:currentInstallationParams:
    .:
    f:IPv6:
    f:autosupportHostname:
    f:autosupportImage:
    f:autosupportProxy:
    f:autosupportSerialNumber:
    f:debug:
    f:enableNodePrep:
    f:imagePullSecrets:
    f:imageRegistry:
    f:k8sTimeout:
    f:kubeletDir:
    f:logFormat:
    f:silenceAutosupport:
    f:tridentImage:
    f:message:
    f:namespace:
    f:status:
    f:version:
  Manager:          trident-operator
  Operation:        Update
  Time:            2021-05-07T17:00:28Z
  Resource Version: 931421
  Self Link:
  /apis/trident.netapp.io/v1/tridentorchestrators/trident
  UID:             8a26a7a6-dde8-4d55-9b66-a7126754d81f
Spec:
  Debug:           true
  Namespace:       trident
Status:
  Current Installation Params:
    IPv6:             false
    Autosupport Hostname:
    Autosupport Image: netapp/trident-autosupport:21.04
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:            true
    Enable Node Prep: false
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:       30
```

```

Kubelet Dir:          /var/lib/kubelet
Log Format:           text
Silence Autosupport: false
Trident Image:        netapp/trident:21.04
Message:              Trident installed
Namespace:            trident
Status:               Installed
Version:              v21.04

Events:
Type    Reason     Age   From                  Message
----  -----  ----  -----  -----
Normal  Installing  80s  trident-operator.netapp.io  Installing Trident
Normal  Installed   68s  trident-operator.netapp.io  Trident installed

```

7. You can verify that Trident is successfully installed by checking the pods that are running in the namespace or by using the `tridentctl` binary to check the installed version.

```

[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                           READY   STATUS    RESTARTS   AGE
trident-csi-bb64c6cb4-lmd6h    6/6     Running   0          82s
trident-csi-gn59q              2/2     Running   0          82s
trident-csi-m4szj              2/2     Running   0          82s
trident-csi-sb9k9              2/2     Running   0          82s
trident-operator-66f48895cc-lzczk 1/1     Running   0          2m39s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 21.04          | 21.04          |
+-----+-----+

```

### Prepare worker nodes for storage

Most Kubernetes distributions come with the packages and utilities to mount NFS backends installed by default, including Red Hat OpenShift.

To prepare worker nodes to allow for the mapping of block storage volumes through the iSCSI protocol, you must install the necessary packages to support that functionality.

In Red Hat OpenShift, this is handled by applying an MCO (Machine Config Operator) to your cluster after it is deployed.

To configure the worker nodes to run storage services, complete the following steps:

1. Log into the OCP web console and navigate to Compute > Machine Configs. Click Create Machine Config. Copy and paste the YAML file and click Create.

When not using multipathing:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-element-iscsi
spec:
  config:
    ignition:
      version: 3.2.0
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
  osImageURL: ""
```

When using multipathing:

```

apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-worker-ontap-iscsi
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-
8;base64,ZGVmYXVsdHMgewogICAgiHVzZXJfZnJpZW5kbH1fbmFtZXMgeWVzCiAgICAgI
CAgZmluZF9tdWx0aXBhdGhzIHllcwp9CgpibGFja2xpc3RfZXhjZXB0aW9ucyB7CiAgICAgICA
gcHJvcGVydHkgIihTQ1NjX01ERU5UX3xJRF9XV04pIgp9CgpibGFja2xpc3Qgewp9Cgo=
            verification: {}
      filesystem: root
      mode: 400
      path: /etc/multipath.conf
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
        - name: multipathd.service
          enabled: true
          state: started
  osImageURL: ""

```

- After the configuration is created, it takes approximately 20 to 30 minutes to apply the configuration to the worker nodes and reload them. Verify whether the machine config is applied by using `oc get mcp` and make sure that the machine config pool for workers is updated. You can also log into the worker nodes to confirm that the iscsid service is running (and the multipathd service is running if using multipathing).

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME      CONFIG                                     UPDATED     UPDATING
DEGRADED
master    rendered-master-a520ae930e1d135e0dee7168   True       False
False
worker    rendered-worker-de321b36eeba62df41feb7bc   True       False
False
```

```
[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
   vendor preset: disabled)
     Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
       Docs: man:iscsid(8)
              man:iscsiadm(8)
   Main PID: 1242 (iscsid)
     Status: "Ready to process requests"
      Tasks: 1
     Memory: 4.9M
        CPU: 9ms
      CGroup: /system.slice/iscsid.service
              └─1242 /usr/sbin/iscsid -f
```

```
[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status multipathd
● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled;
   vendor preset: enabled)
     Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
   Main PID: 918 (multipathd)
     Status: "up"
      Tasks: 7
     Memory: 13.7M
        CPU: 57ms
      CGroup: /system.slice/multipathd.service
              └─918 /sbin/multipathd -d -s
```



It is also possible to confirm that the MachineConfig has been successfully applied and services have been started as expected by running the `oc debug` command with the appropriate flags.

#### Create storage-system backends

After completing the Astra Trident Operator install, you must configure the backend for the specific NetApp

storage platform you are using. Follow the links below in order to continue the setup and configuration of Astra Trident.

- [NetApp ONTAP NFS](#)
- [NetApp ONTAP iSCSI](#)
- [NetApp Element iSCSI](#)

Next: [Solution Validation/Use Cases: Red Hat OpenShift with NetApp](#).

#### NetApp ONTAP NFS Configuration

To enable Trident integration with the NetApp ONTAP storage system, you must create a backend that enables communication with the storage system.

1. There are sample backend files available in the downloaded installation archive in the `sample-input` folder hierarchy. For NetApp ONTAP systems serving NFS, copy the `backend-ontap-nas.json` file to your working directory and edit the file.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-samples/ontap-nas/backend-ontap-nas.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-nas.json
```

2. Edit the `backendName`, `managementLIF`, `dataLIF`, `svm`, `username`, and `password` values in this file.

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nas+10.61.181.221",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.221",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "password"
}
```



Best practice is to define the custom `backendName` value as a combination of the `storageDriverName` and the `dataLIF` that is serving NFS for easy identification.

3. With this backend file in place, run the following command to create your first backend.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-nas.json
+-----+
+-----+-----+
|           NAME          | STORAGE DRIVER |             UUID
| STATE   | VOLUMES   |
+-----+-----+
+-----+-----+-----+
| ontap-nas+10.61.181.221 | ontap-nas      | be7a619d-c81d-445c-b80c-
5c87a73c5b1e | online |     0 |
+-----+-----+
+-----+-----+-----+
```

- With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

- The only edit that must be made to this file is to define the `backendType` value to the name of the storage driver from the newly created backend. Also note the `name`-field value, which must be referenced in a later step.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```



There is an optional field called `fsType` that is defined in this file. This line can be deleted in NFS backends.

- Run the `oc` command to create the storage class.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample `pvc-basic.yaml` file that can be used to perform this action located in `sample-input` as well.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. The only edit that must be made to this file is ensuring that the `storageClassName` field matches the one just created. The PVC definition can be further customized as required by the workload to be provisioned.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. Create the PVC by issuing the `oc` command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME      STATUS      VOLUME                                     CAPACITY
ACCESS MODES     STORAGECLASS     AGE
basic     Bound      pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d   1Gi
          RWO        basic-csi       7s
```

[Next: Solution Validation / Use Cases: Red Hat OpenShift with NetApp.](#)

#### NetApp ONTAP iSCSI Configuration

To enable Trident integration with the NetApp ONTAP storage system you must create a backend that enables communication with the storage system.

1. There are sample backend files available in the downloaded installation archive in the `sample-input` folder hierarchy. For NetApp ONTAP systems serving iSCSI, copy the `backend-ontap-san.json` file to your working directory and edit the file.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-samples/ontap-san/backend-ontap-san.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-san.json
```

2. Edit the managementLIF, dataLIF, svm, username, and password values in this file.

```
{  
    "version": 1,  
    "storageDriverName": "ontap-san",  
    "managementLIF": "172.21.224.201",  
    "dataLIF": "10.61.181.240",  
    "svm": "trident_svm",  
    "username": "admin",  
    "password": "password"  
}
```

3. With this backend file in place, run the following command to create your first backend.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create backend -f backend-ontap-san.json  
+-----+-----+  
+-----+-----+-----+-----+  
|       NAME          | STORAGE DRIVER |           UUID  
| STATE   | VOLUMES |  
+-----+-----+  
+-----+-----+-----+  
| ontapsan_10.61.181.241 | ontap-san      | 6788533c-7fea-4a35-b797-  
fb9bb3322b91 | online | 0 |  
+-----+-----+  
+-----+-----+-----+
```

4. With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. The only edit that must be made to this file is to define the `backendType` value to the name of the storage driver from the newly created backend. Also note the `name-field` value, which must be referenced in a later step.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```



There is an optional field called `fsType` that is defined in this file. In iSCSI backends, this value can be set to a specific Linux filesystem type (XFS, ext4, etc) or can be deleted to allow OpenShift to decide what filesystem to use.

6. Run the `oc` command to create the storage class.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample `pvc-basic.yaml` file that can be used to perform this action located in `sample-inputs` as well.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml .
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. The only edit that must be made to this file is ensuring that the `storageClassName` field matches the one just created. The PVC definition can be further customized as required by the workload to be provisioned.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. Create the PVC by issuing the `oc` command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created
```

```
[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME      STATUS    VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic     Bound    pvc-7ceac1ba-0189-43c7-8f98-094719f7956c   1Gi
RWO          basic-csi   3s
```

Next: [Solution Validation / Use Cases: Red Hat OpenShift with NetApp.](#)

#### NetApp Element iSCSI configuration

To enable Trident integration with the NetApp Element storage system you must create a backend that enables communication with the storage system using the iSCSI protocol.

1. There are sample backend files available in the downloaded installation archive in the `sample-input` folder hierarchy. For NetApp Element systems serving iSCSI, copy the `backend-solidfire.json` file to your working directory, and edit the file.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/solidfire/backend-solidfire.json ./
[netapp-user@rhel7 trident-installer]$ vi ./backend-solidfire.json
```

- a. Edit the user, password, and MVIP value on the `EndPoint` line.
- b. Edit the `SVIP` value.

```
{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://trident:password@172.21.224.150/json-
rpc/8.0",
  "SVIP": "10.61.180.200:3260",
  "TenantName": "trident",
  "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS": 2000, "burstIOPS": 4000},
             {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS": 6000, "burstIOPS": 8000}},
             {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS": 8000, "burstIOPS": 10000}}]
}
```

2. With this back-end file in place, run the following command to create your first backend.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-solidfire.json
+-----+
+-----+-----+
|       NAME          | STORAGE DRIVER |           UUID
| STATE   | VOLUMES   |
+-----+-----+
+-----+-----+-----+
| solidfire_10.61.180.200 | solidfire-san | b90783ee-e0c9-49af-8d26-
3ea87ce2efdf | online |      0 |
+-----+-----+
+-----+-----+
```

- With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

- The only edit that must be made to this file is to define the `backendType` value to the name of the storage driver from the newly created backend. Also note the `name`-field value, which must be referenced in a later step.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
```



There is an optional field called `fsType` that is defined in this file. In iSCSI backends, this value can be set to a specific Linux filesystem type (XFS, ext4, etc) or can be deleted to allow OpenShift to decide what filesystem to use.

- Run the `oc` command to create the storage class.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

6. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample `pvc-basic.yaml` file that can be used to perform this action located in sample-inputs as well.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

7. The only edit that must be made to this file is ensuring that the `storageClassName` field matches the one just created. The PVC definition can be further customized as required by the workload to be provisioned.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

8. Create the PVC by issuing the `oc` command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME      STATUS      VOLUME                                     CAPACITY
ACCESS MODES     STORAGECLASS     AGE
basic     Bound      pvc-3445b5cc-df24-453d-a1e6-b484e874349d   1Gi
          RWO        basic-csi       5s
```

[Next: Solution Validation / Use Cases: Red Hat OpenShift with NetApp.](#)

## Advanced Configuration Options For OpenShift

### Exploring load balancer options: Red Hat OpenShift with NetApp

In most of the cases, Red Hat OpenShift makes applications available to the outside world through routes. A service is exposed by giving it an externally reachable hostname. The defined route and the endpoints identified by its service can be consumed by an OpenShift router to provide this named connectivity to external clients.

However in some cases, applications require the deployment and configuration of customized load balancers to expose the appropriate services. One example of this is NetApp Astra Control Center. To meet this need, we have evaluated a number of custom load balancer options. Their installation and configuration are described in this section.

The following pages have additional information about load balancer options validated in the Red Hat OpenShift with NetApp solution:

- [MetalLB](#)

Next: [Solution validation/use cases: Red Hat OpenShift with NetApp](#).

### Installing MetalLB load balancers: Red Hat OpenShift with NetApp

This page lists the installation and configuration instructions for the MetalLB load balancer.

MetalLB is a self-hosted network load-balancer installed on your OpenShift cluster that allows the creation of OpenShift services of type load balancer in clusters that don't run on a cloud provider. The two main features of MetalLB that work together to support LoadBalancer services are address allocation and external announcement.

#### MetalLB configuration options

Based on how MetalLB announces the IP address assigned to LoadBalancer services outside of the OpenShift cluster, it operates in two modes:

- **Layer 2 mode.** In this mode, one node in the OpenShift cluster takes ownership of the service and responds to ARP requests for that IP to make it reachable outside of the OpenShift cluster. Because only the node advertises the IP, it has a bandwidth bottleneck and slow failover limitations. For more information, see the documentation [here](#).
- **BGP mode.** In this mode, all nodes in the OpenShift cluster establish BGP peering sessions with a router and advertise the routes to forward traffic to the service IPs. The pre-requisite for this is to integrate MetalLB with a router in that network. Owing to the hashing mechanism in BGP, it has certain limitation when IP-to-Node mapping for a service changes. For more information, refer to the documentation [here](#).



For the purpose of this document, we are configuring MetalLB in layer 2 mode.

### Installing The MetalLB Load Balancer

1. Download the MetalLB resources.

```
[netapp-user@rhel7 ~]$ wget https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/namespace.yaml  
[netapp-user@rhel7 ~]$ wget https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/metalb.yaml
```

2. Edit file `metallb.yaml` and remove `spec.template.spec.securityContext` from controller Deployment and the speaker DaemonSet.

**Lines to be deleted:**

```
securityContext:  
  runAsNonRoot: true  
  runAsUser: 65534
```

3. Create the `metallb-system` namespace.

```
[netapp-user@rhel7 ~]$ oc create -f namespace.yaml  
namespace/metallb-system created
```

4. Create the MetalLB CR.

```
[netapp-user@rhel7 ~]$ oc create -f metallb.yaml  
podsecuritypolicy.policy/controller created  
podsecuritypolicy.policy/speaker created  
serviceaccount/controller created  
serviceaccount/speaker created  
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created  
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created  
role.rbac.authorization.k8s.io/config-watcher created  
role.rbac.authorization.k8s.io/pod-lister created  
role.rbac.authorization.k8s.io/controller created  
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller  
created  
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker  
created  
rolebinding.rbac.authorization.k8s.io/config-watcher created  
rolebinding.rbac.authorization.k8s.io/pod-lister created  
rolebinding.rbac.authorization.k8s.io/controller created  
daemonset.apps/speaker created  
deployment.apps/controller created
```

5. Before configuring the MetalLB speaker, grant the speaker DaemonSet elevated privileges so that it can perform the networking configuration required to make the load balancers work.

```
[netapp-user@rhel7 ~]$ oc adm policy add-scc-to-user privileged -n  
metallb-system -z speaker  
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:privileged  
added: "speaker"
```

6. Configure MetalLB by creating a `ConfigMap` in the `metallb-system` namespace.

```
[netapp-user@rhel7 ~]$ vim metallb-config.yaml
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: metallb-config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 10.63.17.10-10.63.17.200
```

```
[netapp-user@rhel7 ~]$ oc create -f metallb-config.yaml
configmap/metallb-config created
```

7. Now when loadbalancer services are created, MetalLB assigns an externalIP to the services and advertises the IP address by responding to ARP requests.



If you wish to configure MetalLB in BGP mode, skip step 6 above and follow the procedure in the MetalLB documentation [here](#).

Next: [Solution validation/use cases: Red Hat OpenShift with NetApp](#).

### Creating Private Image Registries: Red Hat OpenShift with NetApp

For most deployments of Red Hat OpenShift, using a public registry like [Quay.io](#) or [DockerHub](#) meets most customer's needs. However there are times when a customer may want to host their own private or customized images.

This procedure documents creating a private image registry which is backed by a persistent volume provided by Astra Trident and NetApp ONTAP.



Astra Control Center requires a registry to host the images the Astra containers require. The following section describes the steps to setup a private registry on Red Hat OpenShift cluster and pushing the images required to support the installation of Astra Control Center.

#### Creating A Private Image Registry

1. Edit the imageregistry operator, enter the below storage parameters to `spec` section

```
[netapp-user@rhel7 ~]$ oc edit  
configs.imageregistry.operator.openshift.io  
  
storage:  
  pvc:  
    claim:
```

2. Then enter the following parameters to `spec` section for creating a OpenShift route with a custom hostname, save and exit

```
routes:  
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com  
    name: netapp-astra-route
```



The above route config is used when you want a custom hostname for your route. If you want OpenShift to create a route with default hostname, you can just add the following parameters to `spec` section – `defaultRoute: true`

## Custom TLS Certificates

When you are using custom hostname for the route, by default, it uses the default TLS configuration of OpenShift Ingress operator. However, you can add a custom TLS configuration to the route. To do so, following the below steps –

- a. Create a secret with the route's TLS certificates and key –

```
[netapp-user@rhel7 ~]$ oc create secret tls astra-route-tls -n  
openshift-image-registry -cert/home/admin/netapp-astra/tls.crt  
--key=/home/admin/netapp-astra/tls.key
```

- b. Edit the imageregistry operator and add the following parameters to the `spec` section –

```
[netapp-user@rhel7 ~]$ oc edit  
configs.imageregistry.operator.openshift.io  
  
routes:  
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com  
    name: netapp-astra-route  
    secretName: astra-route-tls
```

3. Next step is to edit the imageregistry operator again and change the management state of the operator to `Managed` state, save and exit –

```
oc edit configs.imageregistry/cluster
```

```
managementState: Managed
```

4. If all the pre-requisites are satisfied, it should start creating PVCs, pods and services for the private image registry. In a few minutes, the registry should be up.

```
[netapp-user@rhel7 ~]$oc get all -n openshift-image-registry
```

NAME	READY	STATUS
RESTARTS AGE		
pod/cluster-image-registry-operator-74f6d954b6-rb7zr 3 90d	1/1	Running
pod/image-pruner-1627257600-f5cpj 0 2d9h	0/1	Completed
pod/image-pruner-1627344000-swqx9 0 33h	0/1	Completed
pod/image-pruner-1627430400-rv5nt 0 9h	0/1	Completed
pod/image-registry-6758b547f-6pnj8 0 76m	1/1	Running
pod/node-ca-bwb5r 0 90d	1/1	Running
pod/node-ca-f8w54 0 90d	1/1	Running
pod/node-ca-gjx7h 0 90d	1/1	Running
pod/node-ca-lcx4k 0 33d	1/1	Running
pod/node-ca-v7zmx 0 7d21h	1/1	Running
pod/node-ca-xpppp 0 89d	1/1	Running

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
IP PORT(S) AGE			
service/image-registry 5000/TCP 15h	ClusterIP	172.30.196.167	<none>
service/image-registry-operator 60000/TCP 90d	ClusterIP	None	<none>

NAME	DESIRED	CURRENT	READY	UP-TO-DATE
AVAILABLE	NODE SELECTOR	AGE		
daemonset.apps/node-ca	6	6	6	6
kubernetes.io/os=linux	90d			

NAME	READY	UP-TO-DATE
deployment.apps/cluster-image-registry-operator	1/1	1
90d		1
deployment.apps/image-registry	1/1	1
15h		1

NAME	READY	AGE	DESIRED
replicaset.apps/cluster-image-registry-operator-74f6d954b6	1	90d	1
replicaset.apps/image-registry-6758b547f	1	76m	1
replicaset.apps/image-registry-78bfbd7f59	0	15h	0
replicaset.apps/image-registry-7fcc8d6cc8	0	80m	0
replicaset.apps/image-registry-864f88f5b	0	15h	0
replicaset.apps/image-registry-cb47ffffb	0	10h	0

NAME	COMPLETIONS	DURATION	AGE
job.batch/image-pruner-1627257600	1/1	10s	2d9h
job.batch/image-pruner-1627344000	1/1	6s	33h
job.batch/image-pruner-1627430400	1/1	5s	9h

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST
cronjob.batch/image-pruner	0 0 * * *	False	0	9h
	90d			

NAME	HOST/PORT			
PATH	SERVICES	PORT	TERMINATION	WILDCARD
route.route.openshift.io/public-routes	astra-registry.apps.ocp-vmw.cie.netapp.com	image-registry	<all>	reencrypt None

5. If you are using the default TLS certificates of Ingress operator OpenShift registry route, you can fetch the TLS certificates using the below command.

```
[netapp-user@rhel7 ~]$ oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator
```

6. To allow OpenShift nodes to access and pull the images from the registry, you need to add the certificates

to the docker client on the OpenShift nodes. Create a configmap in `openshift-config` namespace using the TLS certificates and patch it to the cluster image config to make the certificate trusted.

```
[netapp-user@rhel7 ~]$ oc create configmap astra-ca -n openshift-config  
--from-file=astra-registry.apps.ocp-vmw.cie.netapp.com=tls.crt  
  
[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster  
--patch '{"spec":{"additionalTrustedCA":{"name":"astra-ca"}}}'  
--type=merge
```

7. OpenShift internal registry is controlled by authentication. All the OpenShift users can access the OpenShift registry, but the operations that the logged in user can perform depends on the user permissions.

- To allow a user/group of users to pull images from the registry, the user/s must have `registry-viewer` role assigned.

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-viewer  
ocp-user  
  
[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-viewer  
ocp-user-group
```

- To allow a user/group of users to write or push images, the user/s must have `registry-editor` role assigned.

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-editor  
ocp-user  
  
[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-editor  
ocp-user-group
```

8. For OpenShift nodes to access the registry and push/pull the images, you will need to configure a pull secret.

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-  
credentials --docker-server= astra-registry.apps.ocp-vmw.cie.netapp.com  
--docker-username=ocp-user --docker-password=password
```

9. This pull secret can then be patched to serviceaccounts or be referenced in the corresponding pod definition.

- To patch it to service accounts

```
[netapp-user@rhel7 ~]$ oc secrets link <service_account_name> astra-registry-credentials --for=pull
```

- b. To reference the pull secret in Pod definition, add the following parameter to the 'spec' section.

```
imagePullSecrets:  
- name: astra-registry-credentials
```

10. To push/pull an image from workstations apart from OpenShift node.

- a. Add the TLS certificates to the docker client.

```
[netapp-user@rhel7 ~]$ sudo mkdir /etc/docker/certs.d/astra-registry.apps.ocp-vmw.cie.netapp.com  
[netapp-user@rhel7 ~]$ sudo cp /path/to/tls.crt  
/etc/docker/certs.d/astra-registry.apps.ocp-vmw.cie.netapp.com
```

- b. Log into OpenShift using oc login command.

```
[netapp-user@rhel7 ~]$ oc login --token=sha256~D49SpB_lesSrJYwrM0LIO  
-VRcjWHu0a27vKa0 --server=https://api.ocp-vmw.cie.netapp.com:6443
```

- c. Log into the registry using OpenShift user credentials via podman/docker command.

#### podman

```
[netapp-user@rhel7 ~]$ podman login astra-registry.apps.ocp-vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t)
```

#### docker

```
[netapp-user@rhel7 ~]$ docker login astra-registry.apps.ocp-vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t)
```

- d. Push/pull the images.

### **podman**

```
[netapp-user@rhel7 ~]$ podman push astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra/vault-controller:latest  
[netapp-user@rhel7 ~]$ podman pull astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```

### **docker**

```
[netapp-user@rhel7 ~]$ docker push astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra/vault-controller:latest  
[netapp-user@rhel7 ~]$ docker pull astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```

Next: [Solution Validation/Use Cases: Red Hat OpenShift with NetApp](#).

## **Solution Validation and Use Cases: Red Hat OpenShift with NetApp**

The examples provided on this page are solution validations and use cases for Red Hat OpenShift with NetApp.

- [Deploy a Jenkins CI/CD Pipeline with Persistent Storage](#)
- [Configure Multitenancy on Red Hat OpenShift with NetApp](#)
- [Red Hat OpenShift Virtualization with NetApp ONTAP](#)
- [Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp](#)

Next: [Videos and Demos](#).

## **Deploy a Jenkins CI/CD Pipeline with Persistent Storage: Red Hat OpenShift with NetApp**

This section provides the steps to deploy a continuous integration/continuous delivery or deployment (CI/CD) pipeline with Jenkins to validate solution operation.

### **Create the resources required for Jenkins deployment**

To create the resources required for deploying the Jenkins application, complete the following steps:

1. Create a new project named Jenkins.

# Create Project

Name \*

Display Name

Description

[Cancel](#)

[Create](#)

2. In this example, we deployed Jenkins with persistent storage. To support the Jenkins build, create the PVC. Navigate to [Storage > Persistent Volume Claims](#) and click [Create Persistent Volume Claim](#). Select the storage class that was created, make sure that the Persistent Volume Claim Name is jenkins, select the appropriate size and access mode, and then click Create.

## Create Persistent Volume Claim

[Edit YAML](#)**Storage Class** SC basic

Storage class for the new claim.

**Persistent Volume Claim Name \*** jenkins

A unique name for the storage claim within the project.

**Access Mode \*** Single User (RWO)    Shared Access (RWX)    Read Only (ROX)

Permissions to the mounted drive.

**Size \*** 100

GiB



Desired storage capacity.

 Use label selectors to request storage

Use label selectors to define how storage is created.

[Create](#)[Cancel](#)

### Deploy Jenkins with Persistent Storage

To deploy Jenkins with persistent storage, complete the following steps:

1. In the upper left corner, change the role from Administrator to Developer. Click [+Add](#) and select [From Catalog](#). In the [Filter by Keyword](#) bar, search for jenkins. Select Jenkins Service with Persistent Storage.

## Developer Catalog

Add shared apps, services, or source-to-image builders to your project from the Developer Catalog. Cluster admins can install additional apps which will show up here automatically.

All Items
All Items

Languages

Group By: None ▾

Middleware

CI/CD

Other

Type

- Operator Backed (0)
- Helm Charts (0)
- Builder Image (0)
- Template (4)
- Service Class (0)

Template

Jenkins  
provided by Red Hat, Inc.

Jenkins service, with persistent storage. NOTE: You must have persistent volumes available in...

Template

Jenkins  
provided by Red Hat, Inc.

Jenkins service, with persistent storage. NOTE: You must have persistent volumes available in...

Template

Jenkins (Ephemeral)  
provided by Red Hat, Inc.

Jenkins service, without persistent storage. WARNING: Any data stored will be lost upon...

Template

Jenkins (Ephemeral)  
provided by Red Hat, Inc.

Jenkins service, without persistent storage. WARNING:

2. Click **Instantiate Template**.

### Jenkins

Provided by Red Hat, Inc.

x

Instantiate Template

---

Provider	Description
Red Hat, Inc.	Jenkins service, with persistent storage.
Support	NOTE: You must have persistent volumes available in your cluster to use this template.
<a href="#">Get support ↗</a>	
Created At	May 26, 3:58 am
	<b>Documentation</b>
	<a href="https://docs.okd.io/latest/using_images/other_images/jenkins.html" style="color: #0070C0;">https://docs.okd.io/latest/using_images/other_images/jenkins.html ↗</a>

3. By default, the details for the Jenkins application are populated. Based on your requirements, modify the parameters and click **Create**. This process creates all the required resources for supporting Jenkins on

## OpenShift.

### Instantiate Template

Namespace \*

Jenkins Service Name

The name of the OpenShift Service exposed for the Jenkins container.

Jenkins JNLP Service Name

The name of the service used for master/slave communication.

Enable OAuth in Jenkins

Whether to enable OAuth OpenShift integration. If false, the static account 'admin' will be initialized with the password 'password'.

Memory Limit

Maximum amount of memory the container can use.

Volume Capacity \*

Volume space available for data, e.g. 512Mi, 2Gi.

Jenkins ImageStream Namespace

The OpenShift Namespace where the Jenkins ImageStream resides.

Disable memory intensive administrative monitors

Whether to perform memory intensive, possibly slow, synchronization with the Jenkins Update Center on start. If true, the Jenkins core update monitor and site warnings monitor are disabled.

Jenkins ImageStreamTag

Name of the ImageStreamTag to be used for the Jenkins image.

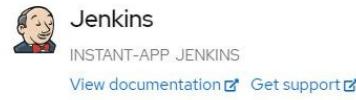
Fatal Error Log File

When a fatal error occurs, an error log is created with information and the state obtained at the time of the fatal error.

Allows use of Jenkins Update Center repository with invalid SSL certificate

Whether to allow use of a Jenkins Update Center that uses invalid certificate (self-signed, unknown CA). If any value other than 'false', certificate check is bypassed. By default, certificate check is enforced.

Create Cancel



Jenkins service, with persistent storage.

NOTE: You must have persistent volumes available in your cluster to use this template.

The following resources will be created:

- DeploymentConfig
- PersistentVolumeClaim
- RoleBinding
- Route
- Service
- ServiceAccount

4. The Jenkins pods take approximately 10–12 minutes to enter the Ready state.

Project: jenkins ▾

## Pods

Create Pod

Filter by name...

1 Running	0 Pending	0 Terminating	0 CrashLoopBackOff	1 Completed	0 Failed	0 Unknown
Select all filters						

1 of 2 Items

Name	Namespace	Status	Ready	Owner	Memory	CPU	⋮
P jenkins-1-c77n9	NS jenkins	Running	1/1	RC jenkins-1	-	0.004 cores	⋮

5. After the pods are instantiated, navigate to [Networking > Routes](#). To open the Jenkins webpage, click the URL provided for the jenkins route.

Project: jenkins ▾

## Routes

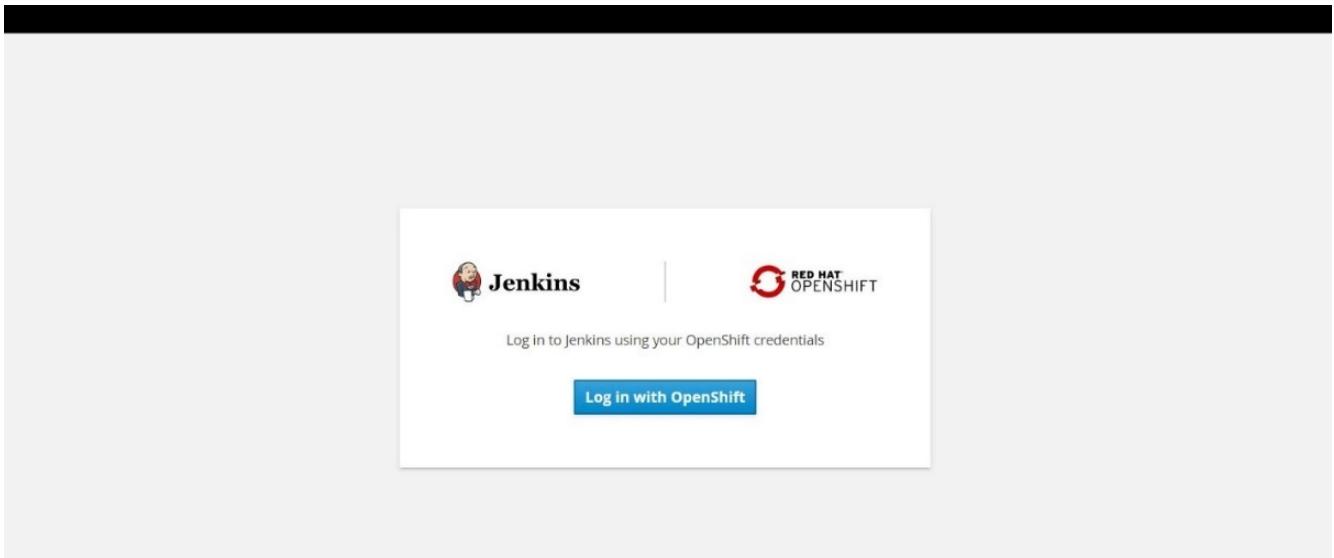
Create Route

Filter by name...

1 Accepted	0 Rejected	0 Pending	Select all filters	1 Item
------------	------------	-----------	--------------------	--------

Name	Namespace	Status	Location	Service	⋮
RT jenkins	NS jenkins	Accepted	https://jenkins-jenkins.apps.rhv-ocp-cluster.cie.netapp.com	S jenkins	⋮

6. Because OpenShift OAuth was used while creating the Jenkins app, click [Log in with OpenShift](#).



7. Authorize the Jenkins service-account to access the OpenShift users.

## Authorize Access

Service account jenkins in project jenkins is requesting permission to access your account (kube:admin)

### Requested permissions

#### user:info

Read-only access to your user information (including username, identities, and group membership)

#### user:check-access

Read-only access to view your privileges (for example, "can I create builds?")

You will be redirected to <https://jenkins-jenkins.apps.rhv-ocp-cluster.cie.netapp.com/securityRealm/finishLogin>

[Allow selected permissions](#) [Deny](#)

8. The Jenkins welcome page is displayed. Because we are using a Maven build, complete the Maven installation first. Navigate to [Manage Jenkins > Global Tool Configuration](#), and then, in the Maven subhead, click [Add Maven](#). Enter the name of your choice and make sure that the Install Automatically option is selected. Click Save.

Maven

Maven installations

Add Maven

Maven Name M3

Install automatically

Install from Apache Version 3.6.3

Delete Installer

Add Installer

Add Maven

List of Maven installations on this system

9. You can now create a pipeline to demonstrate the CI/CD workflow. On the home page, click [Create New Jobs](#) or [New Item](#) from the left-hand menu.

The screenshot shows the Jenkins home page. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user information (kube:admin). Below the bar, a sidebar on the left lists various Jenkins features: New Item, People, Build History, Manage Jenkins, My Views, Open Blue Ocean, Lockable Resources, Credentials, and New View. The main content area has a heading 'Welcome to Jenkins!' and a message 'Please [create new jobs](#) to get started.' Below this, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle).

10. On the Create Item page, enter the name of your choice, select Pipeline, and click Ok.

The screenshot shows the 'Enter an item name' dialog. The input field contains 'sample-demo'. Below the input field, there's a note '» Required field'. A list of project types is shown:

- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Bitbucket Team/Project**: Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**: Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.

11. Select the Pipeline tab. From the Try Sample Pipeline drop-down menu, select [Github + Maven](#). The code is automatically populated. Click Save.

General Build Triggers Advanced Project Options Pipeline

Advanced...

## Pipeline

Definition Pipeline script

Script

```
1 node {
2     def mvnHome
3     stage('Preparation') { // for display purposes
4         // Get some code from a GitHub repository
5         git 'https://github.com/jglick/simple-maven-project-with-tests.git'
6         // Get the Maven tool.
7         // ** NOTE: This 'M3' Maven tool must be configured
8         // ** in the global configuration.
9         mvnHome = tool 'M3'
10    }
11    stage('Build') {
12        // Run the maven build
13        withEnv(["MVN_HOME=$mvnHome"]) {
14            if (isUnix()) {
15                sh '$MVN_HOME/bin/mvn' -Dmaven.test.failure.ignore clean package'
16            } else {
17                bat("%MVN_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package)
18            }
19        }
20    }
21}
```

GitHub + Maven

Use Groovy Sandbox

[Pipeline Syntax](#)

[Save](#) [Apply](#)

12. Click **Build Now** to trigger the development through the preparation, build, and testing phase. It can take several minutes to complete the whole build process and display the results of the build.

 Jenkins

Jenkins > sample-demo >

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Build Now](#)

[Delete Pipeline](#)

[Configure](#)

[Full Stage View](#)

[Open Blue Ocean](#)

[Rename](#)

[Pipeline Syntax](#)

**Build History** [trend](#)

build	last run	changes
#1	May 27, 2020 3:53 PM	No Changes

[Atom feed for all](#) [Atom feed for failures](#)

## Pipeline sample-demo

Last Successful Artifacts

 [simple-maven-project-with-tests-1.0-SNAPSHOT.jar](#) 1.71 KB [view](#)

 [Recent Changes](#)

### Stage View

Average stage times:  
(Average full run time: ~7s)

Preparation	Build	Results
2s	4s	69ms
2s	4s	69ms

 [Latest Test Result \(no failures\)](#)

### Permalinks

- [Last build \(#1\), 1 min 23 sec ago](#)
- [Last stable build \(#1\), 1 min 23 sec ago](#)
- [Last successful build \(#1\), 1 min 23 sec ago](#)
- [Last completed build \(#1\), 1 min 23 sec ago](#)

13. Whenever there are any code changes, the pipeline can be rebuilt to patch the new version of software enabling continuous integration and continuous delivery. Click [Recent Changes](#) to track the changes from the previous version.

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Build Now](#)

[Delete Pipeline](#)

[Configure](#)

[Full Stage View](#)

[Open Blue Ocean](#)

[Rename](#)

[Pipeline Syntax](#)

**Pipeline sample-demo**

Last Successful Artifacts  
simple-maven-project-with-tests-1.0-SNAPSHOT.jar 1.71 KB [view](#)

Recent Changes

**Stage View**

	Preparation	Build	Results
Average stage times: (Average full run time: ~6s)	2s	4s	86ms
#2 May 27, 2020 3:56 PM No Changes	1s	4s	104ms
#1 May 27, 08:53 No Changes	2s	4s	69ms

[Latest Test Result \(no failures\)](#)

**Permalinks**

- [Last build \(#2\), 19 sec ago](#)
- [Last stable build \(#2\), 19 sec ago](#)
- [Last successful build \(#2\), 19 sec ago](#)
- [Last completed build \(#2\), 19 sec ago](#)

Next: Videos and Demos.

## Configure Multi-tenancy on Red Hat OpenShift with NetApp ONTAP

### Configuring Multitenancy on Red Hat OpenShift with NetApp: Red Hat OpenShift with NetApp

Many organizations that run multiple applications or workloads on containers tend to deploy one Red Hat OpenShift cluster per application/workload. This allows the organizations to implement strict isolation for the application/workload, optimize performance, and reduce security vulnerabilities. However, deploying a separate Red Hat OpenShift cluster for each application poses its own set of problems. It increases operational overhead having to monitor and manage each cluster on its own, increases cost owing to dedicated resources for different applications and hinders efficient scalability.

To overcome these problems, one can consider running all the applications/workloads in a single Red Hat OpenShift cluster. But in such an architecture, resource isolation and application security vulnerabilities pose themselves as one of the major challenges. Any security vulnerability in one workload could naturally spill over into another workload, thus increasing the impact zone. In addition, any abrupt uncontrolled resource utilization by one application can affect the performance of another application, because there is no resource allocation policy by default.

Therefore, organizations look out for solutions that pick up the best in both worlds, for example, by allowing them to run all their workloads in a single cluster and yet offering the benefits of a dedicated cluster for each workload.

One such effective solution is to configure multitenancy on Red Hat OpenShift. Multitenancy is an architecture that allows multiple tenants to coexist on the same cluster with proper isolation of resources, security and so on. In this context, a tenant can be viewed as a subset of the cluster resources that are configured to be used by a particular group of users for an exclusive purpose. Configuring multitenancy on a Red Hat OpenShift cluster provides the following advantages:

- A reduction in CapEx and OpEx by allowing cluster resources to be shared
- Lower operational and management overhead
- Securing the workloads from cross-contamination of security breaches
- Protection of workloads from unexpected performance degradation due to resource contention

For a fully realized multitenant OpenShift cluster, quotas and restrictions must be configured for cluster resources belonging to different resource buckets: compute, storage, networking, security, and so on. Although we cover certain aspects of all the resource buckets in this solution, we focus on best-practices for isolating and securing the data served or consumed by multiple workloads on the same Red Hat OpenShift cluster by configuring multitenancy on storage resources that are dynamically allocated by Astra Trident backed by NetApp ONTAP.

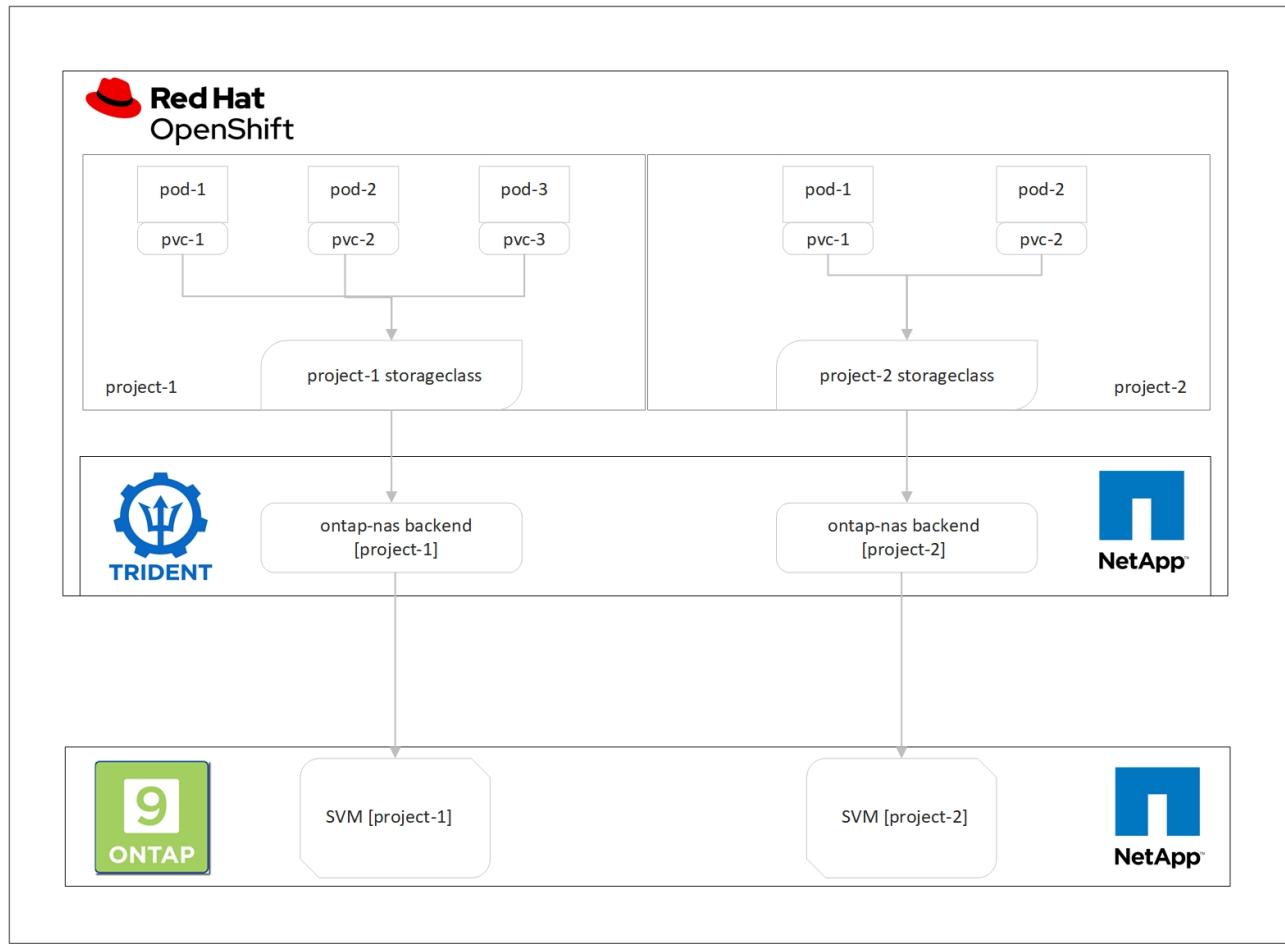
[Next: Architecture.](#)

## Architecture

Although Red Hat OpenShift and Astra Trident backed by NetApp ONTAP do not provide isolation between workloads by default, they offer a wide range of features that can be used to configure multi-tenancy. To better understand designing a multi-tenant solution on a Red Hat OpenShift cluster with Astra Trident backed by NetApp ONTAP, let us consider an example with a set of requirements and outline the configuration around it.

Let us assume that an organization runs two of its workloads on a Red Hat OpenShift cluster as part of two projects that two different teams are working on. The data for these workloads reside on PVCs that are dynamically provisioned by Astra Trident on a NetApp ONTAP NAS backend. The organization has a requirement to design a multi-tenant solution for these two workloads and isolate the resources used for these projects to make sure that security and performance is maintained, primarily focused on the data that serves those applications.

The following figure depicts the multi-tenant solution on a Red Hat OpenShift cluster with Astra Trident backed by NetApp ONTAP.



## Technology requirements

1. NetApp ONTAP storage cluster
2. Red Hat OpenShift cluster
3. Astra Trident

## Red Hat OpenShift – Cluster resources

From the Red Hat OpenShift cluster point of view, the top-level resource to start with is the project. An OpenShift project can be viewed as a cluster resource that divides the whole OpenShift cluster into multiple virtual clusters. Therefore, isolation at project level provides a base for configuring multi-tenancy.

Next up is to configure RBAC in the cluster. The best practice is to have all the developers working on a single project/workload configured into a single user group in the Identity Provider (IdP). Red Hat OpenShift allows IdP integration and user group synchronization thus allowing the users and groups from the IdP to be imported into the cluster. This helps the cluster administrators to segregate access of the cluster resources dedicated to a project to user group/s working on that project, hence restricting unauthorized access to any cluster resources. To learn more about IdP integration with Red Hat OpenShift, refer the documentation [here](#).

## NetApp ONTAP

It is important to isolate the shared storage serving as persistent storage provider for Red Hat OpenShift cluster to ensure the volumes created on the storage for each project appear to the hosts as if they are created

on separate storage. To do this, create as many SVMs (storage virtual machines) on NetApp ONTAP as there are projects/workloads and dedicate each SVM to a workload.

## Astra Trident

After we have different SVMs for different projects created on NetApp ONTAP, we need to map each SVM to a different Trident backend.

The backend configuration on Trident drives the allocation of persistent storage to OpenShift cluster resources and it requires the details of the SVM to be mapped to, protocol driver for the backend at the minimum. Optionally, it allows us to define how the volumes are provisioned on the storage and to set limits for the size of volumes or usage of aggregates etc. Details of defining the Trident backend for NetApp ONTAP can be found [here](#).

## Red Hat OpenShift – storage resources

After configuring the Trident backends, next step is to configure StorageClasses. Configure as many storage classes as there are backends, providing each storage class access to spin up volumes only on one backend. We can map the StorageClass to a particular Trident backend by using storagePools parameter while defining the storage class. The details to define a storage class can be found [here](#). Thus, there will be one-to-one mapping from StorageClass to Trident backend which points back to one SVM. This ensures that all storage claims via the StorageClass assigned to that project will be served by the SVM dedicated to that project only.

But since storage classes are not namespaced resources, how do we ensure that storage claims to storage class of one project by pods in another namespace/project gets rejected? The answer is to use ResourceQuotas. ResourceQuotas are objects that control the total usage of resources per project. It can limit the number as well as the total amount of resources that can be consumed by objects in the project. Almost all the resources of a project can be limited using ResourceQuotas and using this efficiently can help organizations cut cost and outages due to overprovisioning or overconsumption of resources. Refer to the documentation [here](#) for more information.

For this use-case, we need to limit the pods in a particular project from claiming storage from storage classes that are not dedicated to their project. To do that, we need to limit the persistent volume claims for other storage classes by setting `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` to 0. In addition, a cluster administrator must ensure that the developers in a project should not have access to modify the ResourceQuotas.

[Next: Configuration.](#)

## Configuration

For any multitenant solution, no user can have access to more cluster resources than is required. So, the entire set of resources that are to be configured as part of the multitenancy configuration is divided between cluster-admin, storage-admin, and developers working on each project.

The following table outlines the different tasks to be performed by different users:

Role	Tasks
<b>Cluster-admin</b>	Create projects for different applications/workloads
	Create ClusterRoles and RoleBindings for storage-admin
	Create Roles and RoleBindings for developers assigning access to specific projects
	[Optional] Configure projects to schedule pods on specific nodes
<b>Storage-admin</b>	Create SVMs on NetApp ONTAP
	Create Trident backends
	Create StorageClasses
	Create storage ResourceQuotas
<b>Developers</b>	Validate access to create/patch PVCs/pods in assigned project
	Validate access to create/patch PVCs/pods in another project
	Validate access to view/edit Projects, ResourceQuotas, and StorageClasses

[Next: Prerequisites.](#)

## Configuration

### Pre-requisites

- NetApp ONTAP cluster.
- Red Hat OpenShift cluster.
- Trident installed on the cluster.
- Admin workstation with tridentctl and oc tools installed and added to \$PATH.
- Admin access to ONTAP.
- Cluster-admin access to OpenShift cluster.
- Cluster is integrated with Identity Provider.
- Identity provider is configured to efficiently distinguish between users in different teams.

[Next: Cluster Administrator Tasks.](#)

### Configuration: cluster-admin tasks

The following tasks are performed by the Red Hat OpenShift cluster-admin:

1. Log into Red Hat OpenShift cluster as the cluster-admin.
2. Create two projects corresponding to different projects.

```
oc create namespace project-1
oc create namespace project-2
```

### 3. Create the developer role for project-1.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-1
  name: developer-project-1
rules:
- verbs:
  - '*'
  apiGroups:
  - apps
  - batch
  - autoscaling
  - extensions
  - networking.k8s.io
  - policy
  - apps.openshift.io
  - build.openshift.io
  - image.openshift.io
  - ingress.operator.openshift.io
  - route.openshift.io
  - snapshot.storage.k8s.io
  - template.openshift.io
resources:
  - '*'
- verbs:
  - '*'
  apiGroups:
  - ''
resources:
  - bindings
  - configmaps
  - endpoints
  - events
  - persistentvolumeclaims
  - pods
  - pods/log
  - pods/attach
  - podtemplates
  - replicationcontrollers
```

```

    - services
    - limitranges
    - namespaces
    - componentstatuses
    - nodes
- verbs:
    - '*'
apiGroups:
- trident.netapp.io
resources:
- tridentsnapshots
EOF

```



The role definition provided in this section is just an example. Developer role must be defined based on the end-user requirements.

4. Similarly, create developer roles for project-2.
5. All OpenShift and NetApp storage resources are usually managed by a storage admin. Access for storage administrators is controlled by the trident operator role that is created when Trident is installed. In addition to this, the storage admin also requires access to ResourceQuotas to control how storage is consumed.
6. Create a role for managing ResourceQuotas in all projects in the cluster to attach it to storage admin:

```

cat << EOF | oc create -f -
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: resource-quotas-role
rules:
- verbs:
  - '*'
apiGroups:
- ''
resources:
- resourcequotas
- verbs:
  - '*'
apiGroups:
- quota.openshift.io
resources:
- '*'
EOF

```

7. Make sure that the cluster is integrated with the organization's identity provider and that user groups are sync'd with cluster groups. The following example shows that the identity provider has been integrated with the cluster and sync'd with the user groups.

```
$ oc get groups
NAME                      USERS
ocp-netapp-storage-admins ocp-netapp-storage-admin
ocp-project-1              ocp-project-1-user
ocp-project-2              ocp-project-2-user
```

#### 8. Configure ClusterRoleBindings for storage admins.

```
cat << EOF | oc create -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-trident-operator
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-operator
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-resource-quotas-cr
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: resource-quotas-role
EOF
```



For storage admins, two roles must be bound – trident-operator and resource-quotas roles.

#### 9. Create RoleBindings for developers binding the developer-project-1 role to the corresponding group (ocp-project-1) in project-1.

```

cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-1
EOF

```

10. Similarly, create RoleBindings for developers binding the developer roles to the corresponding user group in project-2.

[Next: Storage Administrator Tasks.](#)

### Configuration: Storage-admin tasks

The following resources must be configured by a storage administrator:

1. Log into the NetApp ONTAP cluster as admin.
2. Navigate to [Storage → Storage VMs](#) and click [Add](#). Create two SVMs, one for project-1 and the other for project-2, providing the required details. Also create an vsadmin account to manage the SVM and its resources.

# Add Storage VM

X

STORAGE VM NAME

project-1-svm

## Access Protocol

SMB/CIFS, NFS

iSCSI

Enable SMB/CIFS

Enable NFS

Allow NFS client access

Add at least one rule to allow NFS clients to access volumes in this storage VM. [?](#)

EXPORT POLICY

Default

RULES

Rule Index	Clients	Access Protocols	Read-Only R...	Read/Wr...
	10.61.181.0/24	Any	Any	Any

[+ Add](#)

DEFAULT LANGUAGE [?](#)

c.utf\_8



NETWORK INTERFACE

Use multiple network interfaces when client traffic is high.

K8s-Ontap-01

IP ADDRESS

SUBNET MASK

GATEWAY

BROADCAST DOMAIN

10.61.181.224

24

Add optional  
gateway

Default-4



3. Login to the Red Hat OpenShift cluster as the storage administrator.
4. Create the backend for project-1 and map it to the SVM dedicated to the project. NetApp recommends using the SVM's vsadmin account to connect the backend to SVM instead of using the ONTAP cluster administrator.

```

cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_1",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.224",
    "svm": "project-1-svm",
    "username": "vsadmin",
    "password": "NetApp123"
}
EOF

```



We are using the ontap-nas driver for this example. Use the appropriate driver when creating the backend based on the use-case.



We assume that Trident is installed in trident project.

5. Similarly create the Trident backend for project-2 and map it to the SVM dedicated to project-2.
6. Next, create the storage classes. Create the storage class for project-1 and configure it to use the storage pools from backend dedicated to project-1 by setting the storagePools parameter.

```

cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-1-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_1:.*"
EOF

```

7. Likewise, create a storage class for project-2 and configure it to use the storage pools from backend dedicated to project-2.
8. Create a ResourceQuota to restrict resources in project-1 requesting storage from storageclasses dedicated to other projects.

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-1-sc-rq
  namespace: project-1
spec:
  hard:
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

9. Similarly, create a ResourceQuota to restrict resources in project-2 requesting storage from storageclasses dedicated to other projects.

[Next: Validation.](#)

## Validation

To validate the multitenant architecture that was configured in the previous steps, complete the following steps:

### Validate access to create PVCs/pods in assigned project

1. Log in as ocp-project-1-user, developer in project-1.
2. Check access to create a new project.

```
oc create ns sub-project-1
```

3. Create a PVC in project-1 using the storageclass that is assigned to project-1.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. Check the PV associated with the PVC.

```
oc get pv
```

5. Validate that the PV and its volume is created in an SVM dedicated to project-1 on NetApp ONTAP.

```
volume show -vserver project-1-svm
```

6. Create a pod in project-1 and mount the PVC created in previous step.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  volumes:
    - name: test-pvc-project-1
      persistentVolumeClaim:
        claimName: test-pvc-project-1
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
  volumeMounts:
    - mountPath: "/usr/share/nginx/html"
      name: test-pvc-project-1
EOF
```

7. Check if the pod is running and whether it mounted the volume.

```
oc describe pods test-pvc-pod -n project-1
```

#### **Validate access to create PVCs/pods in another project or use resources dedicated to another project**

1. Log in as ocp-project-1-user, developer in project-1.
2. Create a PVC in project-1 using the storageclass that is assigned to project-2.

```

cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1-sc-2
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-2-sc
EOF

```

### 3. Create a PVC in project-2.

```

cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-2-sc-1
  namespace: project-2
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF

```

### 4. Make sure that PVCs `test-pvc-project-1-sc-2` and `test-pvc-project-2-sc-1` were not created.

```

oc get pvc -n project-1
oc get pvc -n project-2

```

### 5. Create a pod in project-2.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
EOF
```

### Validate access to view/edit Projects, ResourceQuotas, and StorageClasses

1. Log in as ocp-project-1-user, developer in project-1.
2. Check access to create new projects.

```
oc create ns sub-project-1
```

3. Validate access to view projects.

```
oc get ns
```

4. Check if the user can view or edit ResourceQuotas in project-1.

```
oc get resourcequotas -n project-1
oc edit resourcequotas project-1-sc-rq -n project-1
```

5. Validate that the user has access to view the storageclasses.

```
oc get sc
```

6. Check access to describe the storageclasses.
7. Validate the user's access to edit the storageclasses.

```
oc edit sc project-1-sc
```

[Next: Scaling.](#)

#### **Scaling: Adding more projects**

In a multitenant configuration, adding new projects with storage resources requires additional configuration to make sure that multitenancy is not violated. For adding more projects in a multitenant cluster, complete the following steps:

1. Log into the NetApp ONTAP cluster as a storage admin.
2. Navigate to `Storage → Storage VMs` and click `Add`. Create a new SVM dedicated to project-3. Also create a vsadmin account to manage the SVM and its resources.

# Add Storage VM

X

STORAGE VM NAME

project-3-svm

## Access Protocol

SMB/CIFS, NFS

iSCSI

Enable SMB/CIFS

Enable NFS

Allow NFS client access

Add at least one rule to allow NFS clients to access volumes in this storage VM. [?](#)

EXPORT POLICY

Default

RULES

Rule Index	Clients	Access Protocols	Read-Only R...	Read/Wr...
	10.61.181.0/24	Any	Any	Any

[+ Add](#)

DEFAULT LANGUAGE [?](#)

c.utf\_8



NETWORK INTERFACE

Use multiple network interfaces when client traffic is high.

K8s-Ontap-01

IP ADDRESS

10.61.181.228

SUBNET MASK

24

GATEWAY

Add optional  
gateway

BROADCAST DOMAIN

Default-4



3. Log into the Red Hat OpenShift cluster as cluster admin.

4. Create a new project.

```
oc create ns project-3
```

5. Make sure that the user group for project-3 is created on IdP and sync'd with the OpenShift cluster.

```
oc get groups
```

## 6. Create the developer role for project-3.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-3
  name: developer-project-3
rules:
- verbs:
  - '*'
  apiGroups:
  - apps
  - batch
  - autoscaling
  - extensions
  - networking.k8s.io
  - policy
  - apps.openshift.io
  - build.openshift.io
  - image.openshift.io
  - ingress.operator.openshift.io
  - route.openshift.io
  - snapshot.storage.k8s.io
  - template.openshift.io
resources:
  - '*'
- verbs:
  - '*'
  apiGroups:
  - ''
resources:
  - bindings
  - configmaps
  - endpoints
  - events
  - persistentvolumeclaims
  - pods
  - pods/log
  - pods/attach
  - podtemplates
  - replicationcontrollers
  - services
```

```

- limitranges
- namespaces
- componentstatuses
- nodes
- verbs:
  - '*'
apiGroups:
- trident.netapp.io
resources:
- tridentsnapshots
EOF

```



The role definition provided in this section is just an example. The developer role must be defined based on the end-user requirements.

7. Create RoleBinding for developers in project-3 binding the developer-project-3 role to the corresponding group (ocp-project-3) in project-3.

```

cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-3-developer
  namespace: project-3
subjects:
- kind: Group
  apiGroup: rbac.authorization.k8s.io
  name: ocp-project-3
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-3
EOF

```

8. Login to the Red Hat OpenShift cluster as storage admin
9. Create a Trident backend and map it to the SVM dedicated to project-3. NetApp recommends using the SVM's vsadmin account to connect the backend to the SVM instead of using the ONTAP cluster administrator.

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_3",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.228",
    "svm": "project-3-svm",
    "username": "vsadmin",
    "password": "NetApp!23"
}
EOF
```



We are using the ontap-nas driver for this example. Use the appropriate driver for creating the backend based on the use-case.



We assume that Trident is installed in the trident project.

10. Create the storage class for project-3 and configure it to use the storage pools from backend dedicated to project-3.

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-3-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_3:.*"
EOF
```

11. Create a ResourceQuota to restrict resources in project-3 requesting storage from storageclasses dedicated to other projects.

```

cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-3-sc-rq
  namespace: project-3
spec:
  hard:
    project-1-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF

```

12. Patch the ResourceQuotas in other projects to restrict resources in those projects from accessing storage from the storageclass dedicated to project-3.

```

oc patch resourcequotas project-1-sc-rq -n project-1 --patch
'{"spec":{"hard":{"project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
oc patch resourcequotas project-2-sc-rq -n project-2 --patch
'{"spec":{"hard":{"project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'

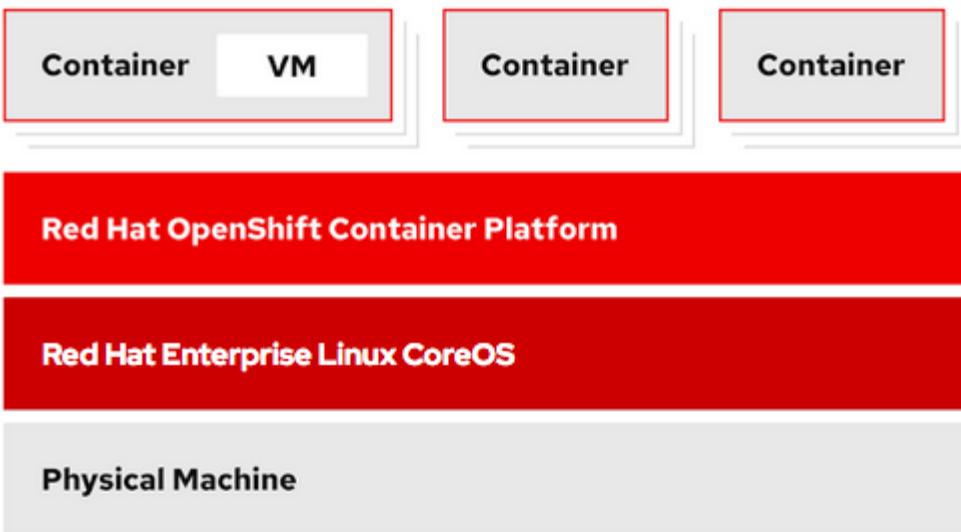
```

## **Red Hat OpenShift Virtualization with NetApp ONTAP**

### **Red Hat OpenShift Virtualization with NetApp ONTAP**

Depending on the specific use-case, both containers and virtual machines (VMs) are considered to offer an optimal platform for different types of applications. Therefore, many organizations run some of their workloads on containers and some on VMs. Often, this leads organizations to face additional challenges by having to manage separate platforms: a hypervisor for VMs and a container orchestrator for applications.

To address this challenge, Red Hat introduced OpenShift Virtualization (formerly known as Container Native Virtualization) starting from OpenShift version 4.6. The OpenShift Virtualization feature enables you to run and manage virtual machines alongside containers on the same OpenShift Container Platform installation, providing hybrid management capability to automate deployment and management of VMs through operators. In addition to creating VMs in OpenShift, with OpenShift Virtualization, Red Hat also supports importing VMs from VMware vSphere, Red Hat Virtualization, and Red Hat OpenStack Platform deployments.



Certain features like live VM migration, VM disk cloning, VM snapshots and so on are also supported by OpenShift Virtualization with assistance from Astra Trident when backed by NetApp ONTAP. Examples of each of these workflows are discussed later in this document in their respective sections.

To learn more about Red Hat OpenShift Virtualization, see the documentation [here](#).

[Next: Deployment Prerequisites.](#)

## Deployment

### Deploy Red Hat OpenShift Virtualization with NetApp ONTAP

#### Prerequisites:

- A Red Hat OpenShift cluster (later than version 4.6) installed on bare-metal infrastructure with RHCOS worker nodes
- The OpenShift cluster must be installed via installer provisioned infrastructure (IPI)
- Deploy Machine Health Checks to maintain HA for VMs
- A NetApp ONTAP cluster
- Astra Trident installed on the OpenShift cluster
- A Trident backend configured with an SVM on ONTAP cluster
- A StorageClass configured on the OpenShift cluster with Astra Trident as the provisioner
- Cluster-admin access to Red Hat OpenShift cluster
- Admin access to NetApp ONTAP cluster
- An admin workstation with tridentctl and oc tools installed and added to \$PATH

Because OpenShift Virtualization is managed by an operator installed on the OpenShift cluster, it imposes additional overhead on memory, CPU, and storage, which must be accounted for while planning the hardware requirements for the cluster. See the documentation [here](#) for more details.

Optionally, you can also specify a subset of the OpenShift cluster nodes to host the OpenShift Virtualization operators, controllers, and VMs by configuring node placement rules. To configure node placement rules for OpenShift Virtualization, follow the documentation [here](#).

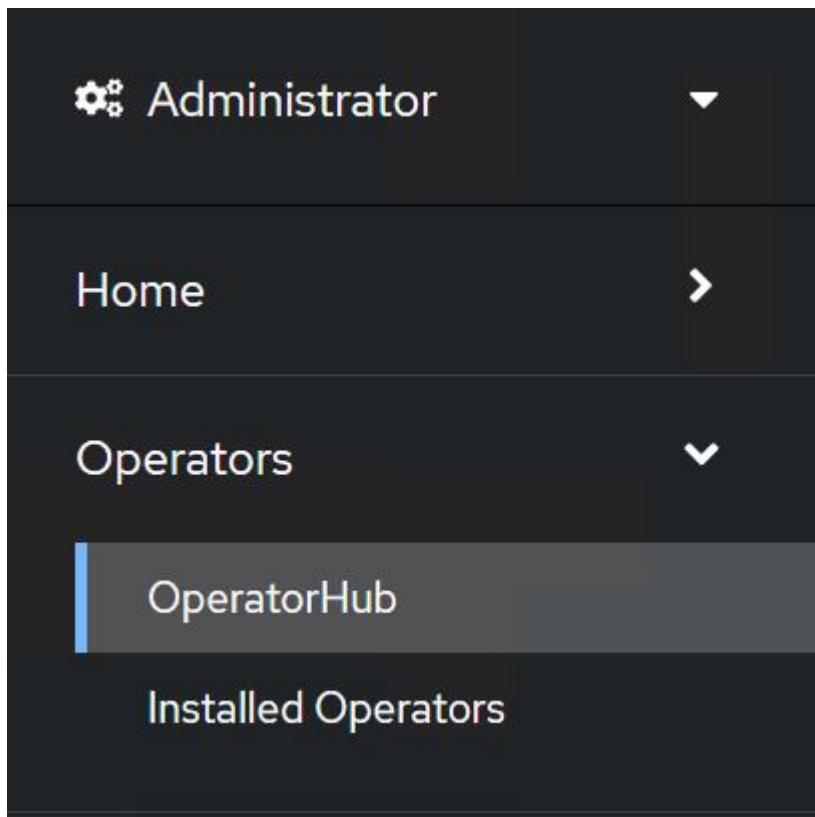
For the storage backing OpenShift Virtualization, NetApp recommends having a dedicated StorageClass that requests storage from a particular Trident backend, which in turn is backed by a dedicated SVM. This maintains a level of multitenancy with regard to the data being served for VM-based workloads on the OpenShift cluster.

Next: Deploy via operator.

## Deploy Red Hat OpenShift Virtualization with NetApp ONTAP

To install OpenShift Virtualization, complete the following steps:

1. Log into the Red Hat OpenShift bare-metal cluster with cluster-admin access.
2. Select Administrator from the Perspective drop down.
3. Navigate to [Operators → OperatorHub](#) and search for OpenShift Virtualization.



4. Select the OpenShift Virtualization tile and click Install.



## OpenShift Virtualization

2.6.2 provided by Red Hat



Install

### Latest version

2.6.2

### Capability level

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

### Provider type

Red Hat

### Provider

Red Hat

## Requirements

Your cluster must be installed on bare metal infrastructure with Red Hat Enterprise Linux CoreOS workers.

## Details

**OpenShift Virtualization** extends Red Hat OpenShift Container Platform, allowing you to host and manage virtualized workloads on the same platform as container-based workloads. From the OpenShift Container Platform web console, you can import a VMware virtual machine from vSphere, create new or clone existing VMs, perform live migrations between nodes, and more. You can use OpenShift Virtualization to manage both Linux and Windows VMs.

The technology behind OpenShift Virtualization is developed in the [KubeVirt](#) open source community. The KubeVirt project extends [Kubernetes](#) by adding additional virtualization resource types through [Custom Resource Definitions](#) (CRDs). Administrators can use Custom Resource Definitions to manage [VirtualMachine](#) resources alongside all other resources that Kubernetes provides.

5. On the Install Operator screen, leave all default parameters and click Install.

Update channel \*

2.1  
 2.2  
 2.3  
 2.4  
 stable

Installation mode \*

All namespaces on the cluster (default)  
This mode is not supported by this Operator  
 A specific namespace on the cluster  
Operator will be available in a single Namespace only.

Installed Namespace \*

Operator recommended Namespace: openshift-cnv

Namespace creation  
Namespace `openshift-cnv` does not exist and will be created.

Select a Namespace

Approval strategy \*

Automatic  
 Manual

**HC** OpenShift Virtualization Deployment Required

Provides APIs

Represents the deployment of OpenShift Virtualization

**Install** **Cancel**

6. Wait for the operator installation to complete.



**OpenShift Virtualization**  
2.6.2 provided by Red Hat

A progress bar at the bottom indicates the operator is being installed.

## Installing Operator

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace openshift-cnv](#)

7. After the operator has installed, click Create HyperConverged.



**OpenShift Virtualization**  
2.6.2 provided by Red Hat

A green checkmark icon is present on the right side of the card.

## Installed operator - operand required

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

**HC** HyperConverged ! Required

Creates and maintains an OpenShift Virtualization Deployment

[Create HyperConverged](#)

[View installed Operators in Namespace openshift-cnv](#)

8. On the Create HyperConverged screen, click Create, accepting all default parameters. This step starts the installation of OpenShift Virtualization.

Name \*

kubevirt-hyperconverged

Labels

app=frontend

Infra

infra HyperConvergedConfig influences the pod configuration (currently only placement) for all the infra components needed on the virtualization enabled cluster but not necessarily directly on each node running VMs/VMIs.

Workloads

workloads HyperConvergedConfig influences the pod configuration (currently only placement) of components which need to be running on a node where virtualization workloads should be able to run. Changes to Workloads HyperConvergedConfig can be applied only without existing workload.

Bare Metal Platform



true

BareMetalPlatform indicates whether the infrastructure is baremetal.

Feature Gates

featureGates is a map of feature gate flags. Setting a flag to `true` will enable the feature. Setting `false` or removing the feature gate, disables the feature.

Local Storage Class Name

LocalStorageClassName the name of the local storage class.

Create

Cancel

- After all the pods move to the Running state in the openshift-cnv namespace and the OpenShift Virtualization operator is in the Succeeded state, the operator is ready to use. VMs can now be created on the OpenShift cluster.

Project: openshift-cnv ▾

## Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name	Managed Namespaces	Status	Last updated	Provided APIs
 <a href="#">OpenShift Virtualization</a> 2.6.2 provided by Red Hat	 <a href="#">openshift-cnv</a>	<span>✓ Succeeded</span> Up to date	May 18, 8:02 pm	<a href="#">OpenShift Virtualization Deployment</a> <a href="#">HostPathProvisioner deployment</a>

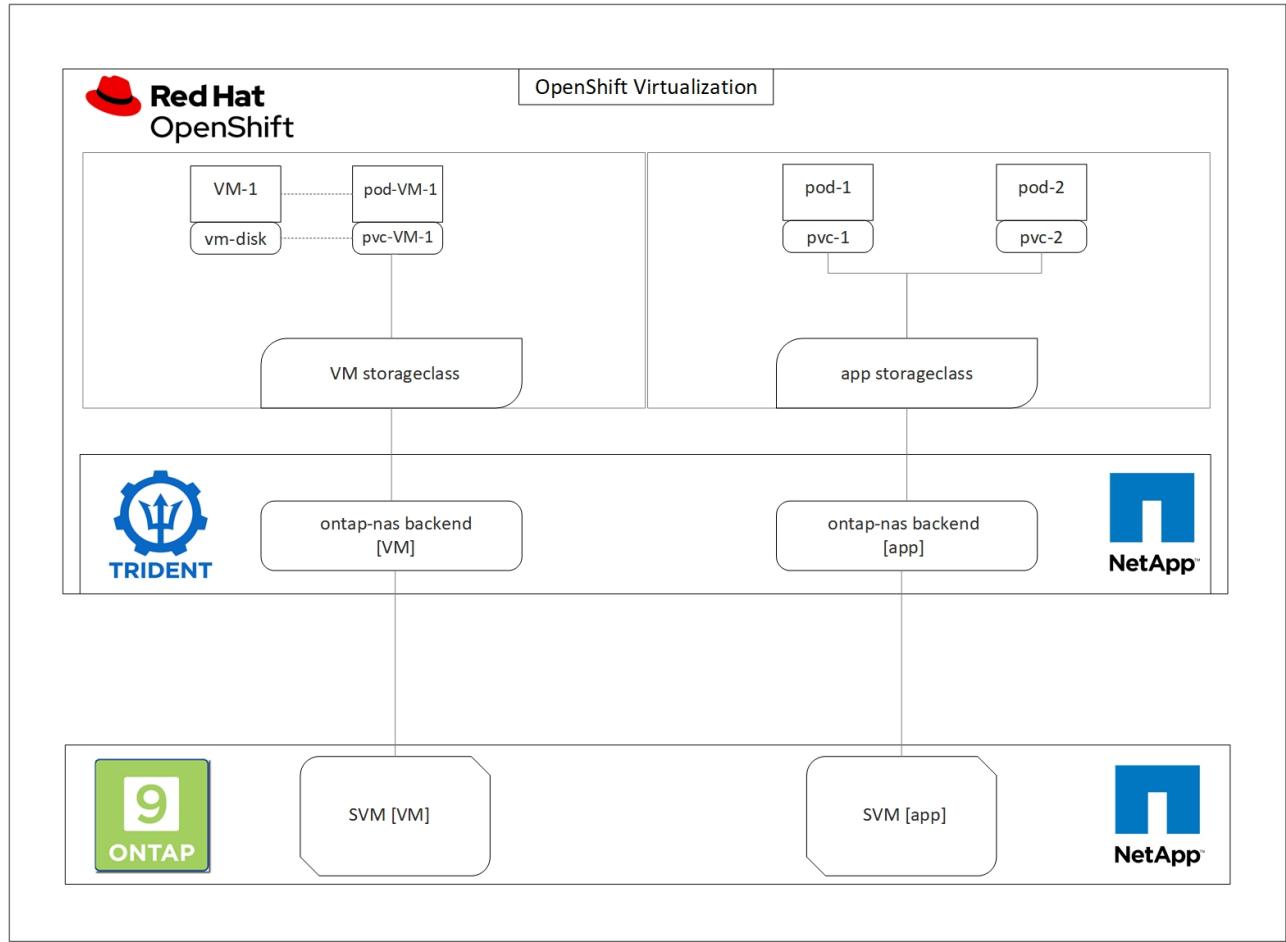
[Next: Workflows: Create VM.](#)

## Workflows

### Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

## Create VM

VMs are stateful deployments that require volumes to host the operating system and data. With CNV, because the VMs are run as pods, the VMs are backed by PVs hosted on NetApp ONTAP through Trident. These volumes are attached as disks and store the entire filesystem including the boot source of the VM.



To create a virtual machine on the OpenShift cluster, complete the following steps:

1. Navigate to `Workloads > Virtualization > Virtual Machines` and click `Create > With Wizard`.
2. Select the desired the operating system and click 'Next'.
3. If the selected operating system has no boot source configured, you must configure it. For Boot Source, select whether you want to import the OS image from an URL or from a registry, and provide the corresponding details. Expand Advanced and select the Trident-backed StorageClass. Then click Next.

## Boot source

This template does not have a boot source. Provide a custom boot source for this **CentOS 8.0+** VM virtual machine.

### Boot source type \*

Import via URL (creates PVC)

### Import URL \*

<https://access.cdn.redhat.com/content/origin/files/sha256/58/588167f828001e57688ec4b9b31c11a59d532489f527488ebc89ac5e952...>

Example: For RHEL, visit the [RHEL download page](#) (requires login) and copy the download link URL of the KVM guest image

Mount this as a CD-ROM boot source ?

### Persistent Volume Claim size \*

5 GiB ▾

Ensure your PVC size covers the requirements of the uncompressed image and any other space requirements. More storage can be added later.

### Advanced

### Storage class \*

basic (default)

### Access mode \*

Single User (RWO)

### Volume mode \*

Filesystem

4. If the selected operating system already has a boot source configured, the previous step can be skipped.
5. In the Review and Create pane, select the project you want to create the VM in and furnish the VM details. Make sure that the boot source is selected to be Clone and boot from CD-ROM with the appropriate PVC assigned for the selected OS.

- 1 Select template
- 2 Review and create

**Review and create**  
You are creating a virtual machine from the Red Hat Enterprise Linux 8.0+ VM template.

**Project \***

PR default

**Virtual Machine Name \* ⓘ**

rhel8-light-bat

**Flavor \***

Small:1CPU | 2 GiB Memory

<b>Storage</b>	<b>Workload profile ⓘ</b>
40 GiB	server

**Boot source**

Clone and boot from CD-ROM

PVC rhel8

ⓘ A new disk has been added to support the CD-ROM boot source. Edit this disk by customizing the virtual machine.

▼ Disk details

rootdisk-install - Blank - 20GiB - virtio - default Storage class

Start this virtual machine after creation

**Create virtual machine**   **Customize virtual machine**   **Back**   **Cancel**

6. If you wish to customize the virtual machine, click Customize Virtual Machine and modify the required parameters.
7. Click Create Virtual Machine to create the virtual machine; this spins up a corresponding pod in the background.

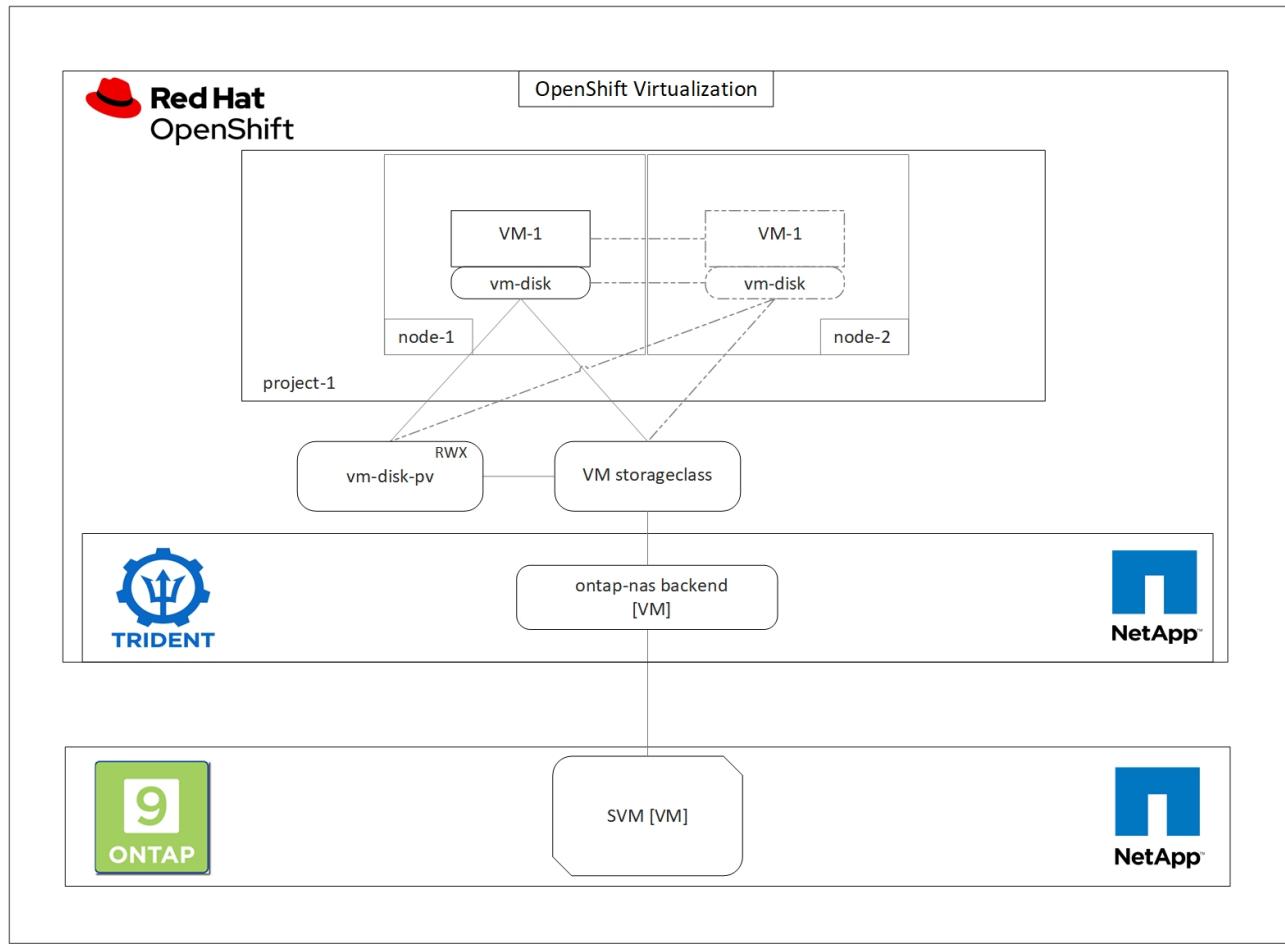
When a boot source is configured for a template or an operating system from an URL or from a registry, it creates a PVC in the `openshift-virtualization-os-images` project and downloads the KVM guest image to the PVC. You must make sure that template PVCs have enough provisioned space to accommodate the KVM guest image for the corresponding OS. These PVCs are then cloned and attached as rootdisks to virtual machines when they are created using the respective templates in any project.

Next: [Workflows: VM Live Migration](#).

## Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

### VM Live Migration

Live Migration is a process of migrating a VM instance from one node to another in an OpenShift cluster with no downtime. For live migration to work in an OpenShift cluster, VMs must be bound to PVCs with shared ReadWriteMany access mode. Astra Trident backend configured with an SVM on a NetApp ONTAP cluster that is enabled for NFS protocol supports shared ReadWriteMany access for PVCs. Therefore, the VMs with PVCs that are requested from StorageClasses provisioned by Trident from NFS-enabled SVM can be migrated with no downtime.



To create a VM bound to PVCs with shared ReadWriteMany access:

1. Navigate to `Workloads > Virtualization > Virtual Machines` and click `Create > With Wizard`.
2. Select the desired operating system and click `Next`. Let us assume the selected OS already had a boot source configured with it.
3. In the `Review and Create` pane, select the project you want to create the VM in and furnish the VM details. Make sure that the boot source is selected to be `Clone and boot from CD-ROM` with the appropriate PVC assigned for the selected OS.
4. Click `Customize Virtual Machine` and then click `Storage`.
5. Click on the ellipsis next to `rootdisk`, make sure that the storageclass provisioned using Trident is selected. Expand `Advanced` and select `Shared Access (RWX)` for `Access Mode`. Then click `Save`.

## Edit Disk

Type: Disk

Interface \*

virtio

Storage Class

basic (default)

▼ Advanced

Volume Mode

Filesystem

Volume Mode is set by Source PVC

Access Mode

Shared Access (RWX) - Not recommended for basic storage class

**ⓘ Access and Volume modes should follow storage feature matrix**

[Learn more ↗](#)

Cancel Save

6. Click Review and confirm and then click Create Virtual Machine.

To manually migrate a VM to another node in the OpenShift cluster, complete the following steps.

1. Navigate to [Workloads > Virtualization > Virtual Machines](#).

2. For the VM you wish to migrate, click the ellipsis, and then click Migrate the Virtual Machine.

3. Click Migrate when the message pops up to confirm.



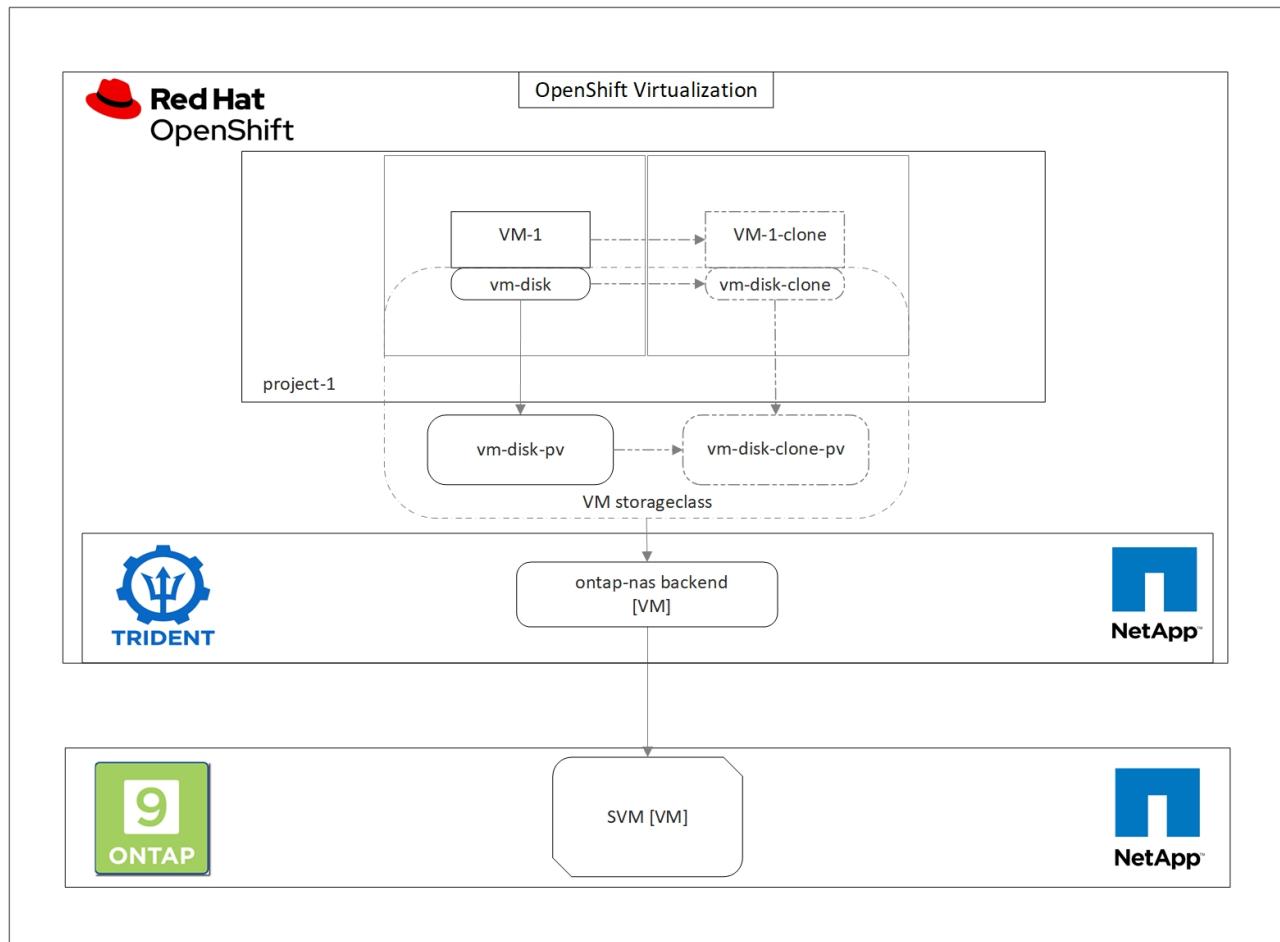
A VM instance in an OpenShift cluster automatically migrates to another node when the original node is placed into maintenance mode if the evictionStrategy is set to LiveMigrate.

Next: [Workflows: VM Cloning](#).

## Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

### VM cloning

Cloning an existing VM in OpenShift is achieved with the support of Astra Trident's Volume CSI cloning feature. CSI volume cloning allows for creation of a new PVC using an existing PVC as the data source by duplicating its PV. After the new PVC is created, it functions as a separate entity and without any link to or dependency on the source PVC.



There are certain restrictions with CSI volume cloning to consider:

1. Source PVC and destination PVC must be in the same project.
2. Cloning is supported within the same storage class.
3. Cloning can be performed only when source and destination volumes use the same VolumeMode setting;

for example, a block volume can only be cloned to another block volume.

VMs in an OpenShift cluster can be cloned in two ways:

1. By shutting down the source VM
2. By keeping the source VM live

### **By Shutting down the source VM**

Cloning an existing VM by shutting down the VM is a native OpenShift feature that is implemented with support from Astra Trident. Complete the following steps to clone a VM.

1. Navigate to `Workloads > Virtualization > Virtual Machines` and click the ellipsis next to the virtual machine you wish to clone.
2. Click Clone Virtual Machine and provide the details for the new VM.

# Clone Virtual Machine

Name *	<input type="text" value="rhel8-short-frog-clone"/>											
Description	<input type="text"/>											
Namespace *	<input type="text" value="default"/>											
<input checked="" type="checkbox"/> Start virtual machine on clone												
Configuration	<table><tr><td>Operating System</td></tr><tr><td>Red Hat Enterprise Linux 8.0 or higher</td></tr><tr><td>Flavor</td></tr><tr><td>Small: 1 CPU   2 GiB Memory</td></tr><tr><td>Workload Profile</td></tr><tr><td>server</td></tr><tr><td>NICs</td></tr><tr><td>default - virtio</td></tr><tr><td>Disks</td></tr><tr><td>cloudinitdisk - cloud-init disk</td></tr><tr><td>rootdisk - 20Gi - basic</td></tr></table>	Operating System	Red Hat Enterprise Linux 8.0 or higher	Flavor	Small: 1 CPU   2 GiB Memory	Workload Profile	server	NICs	default - virtio	Disks	cloudinitdisk - cloud-init disk	rootdisk - 20Gi - basic
Operating System												
Red Hat Enterprise Linux 8.0 or higher												
Flavor												
Small: 1 CPU   2 GiB Memory												
Workload Profile												
server												
NICs												
default - virtio												
Disks												
cloudinitdisk - cloud-init disk												
rootdisk - 20Gi - basic												



The VM rhel8-short-frog is still running. It will be powered off while cloning.

[Cancel](#)

[Clone Virtual Machine](#)

3. Click Clone Virtual Machine; this shuts down the source VM and initiates the creation of the clone VM.
4. After this step is completed, you can access and verify the content of the cloned VM.

## By keeping the source VM live

An existing VM can also be cloned by cloning the existing PVC of the source VM and then creating a new VM using the cloned PVC. This method does not require you to shut down the source VM. Complete the following steps to clone a VM without shutting it down.

1. Navigate to Storage → PersistentVolumeClaims` and click the ellipsis next to the PVC that is attached to the source VM.
2. Click Clone PVC and furnish the details for the new PVC.

## Clone

Name \*

rhel8-short-frog-rootdisk-28dvb-clone

Access Mode \*

Single User (RWO)  Shared Access (RWX)  Read Only (ROX)

Size \*

20

GiB



PVC details

Namespace	Requested capacity	Access mode
NS default	20 GiB	Shared Access (RWX)
Storage Class	Used capacity	Volume mode
SC basic	2.2 GiB	Filesystem

Cancel

Clone

3. Then click Clone. This creates a PVC for the new VM.
4. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With YAML.
5. In the `spec > template > spec > volumes` section, attach the cloned PVC instead of the container disk. Provide all other details for the new VM according to your requirements.

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dwb-clone
```

6. Click Create to create the new VM.
7. After the VM is created successfully, access and verify that the new VM is a clone of the source VM.

Next: [Workflows: Create VM from a Snapshot](#).

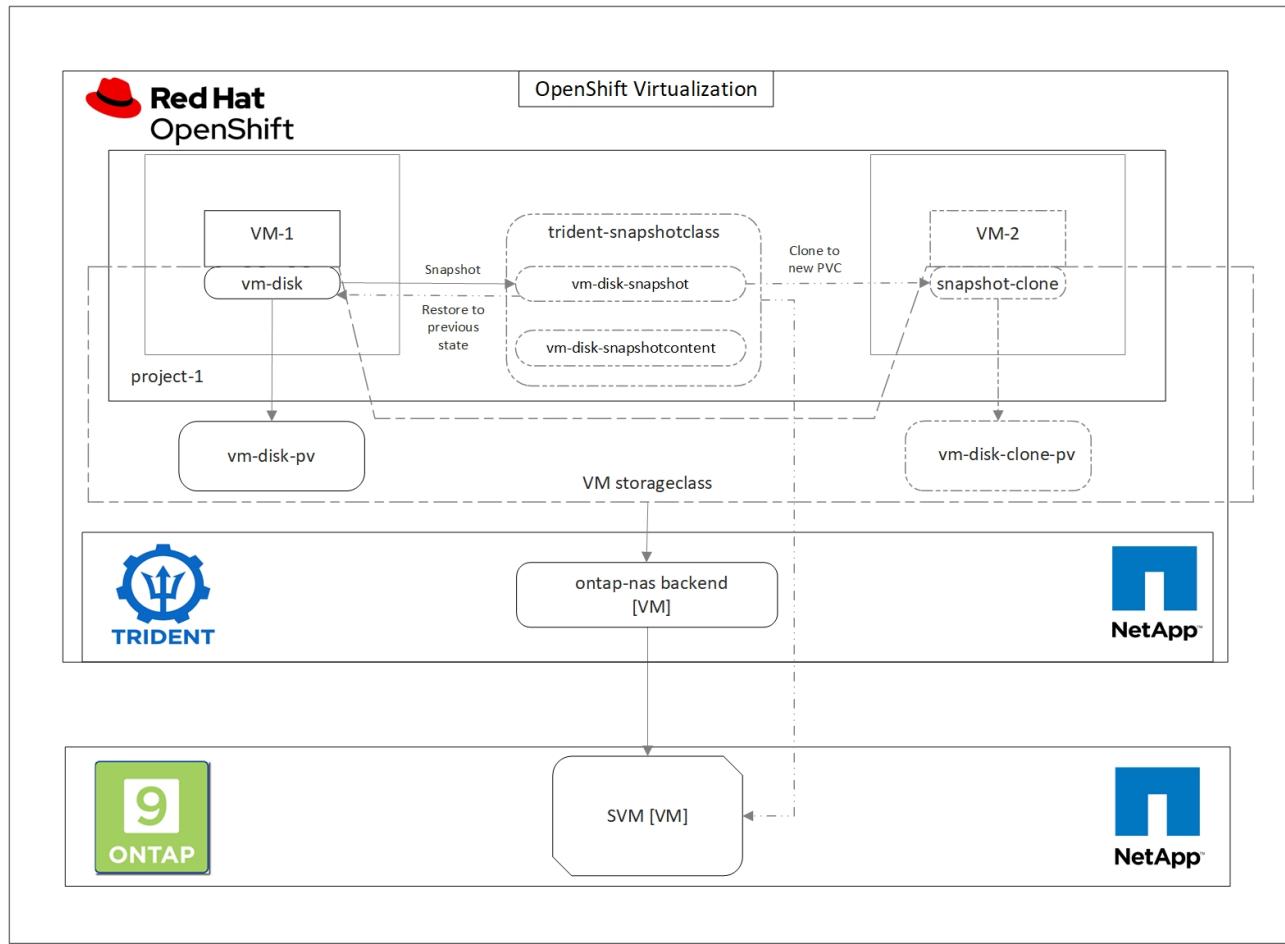
## Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

### Create VM from a Snapshot

With Astra Trident and Red Hat OpenShift, users can take a snapshot of a persistent volume on Storage Classes provisioned by it. With this feature, users can take a point-in-time copy of a volume and use it to create a new volume or restore the same volume back to a previous state. This enables or supports a variety of use-cases, from rollback to clones to data restore.

For Snapshot operations in OpenShift, the resources VolumeSnapshotClass, VolumeSnapshot, and VolumeSnapshotContent must be defined.

- A VolumeSnapshotContent is the actual snapshot taken from a volume in the cluster. It is cluster-wide resource analogous to PersistentVolume for storage.
- A VolumeSnapshot is a request for creating the snapshot of a volume. It is analogous to a PersistentVolumeClaim.
- VolumeSnapshotClass lets the administrator specify different attributes for a VolumeSnapshot. It allows you to have different attributes for different snapshots taken from the same volume.



To create Snapshot of a VM, complete the following steps:

1. Create a VolumeSnapshotClass that can then be used to create a VolumeSnapshot. Navigate to [Storage > VolumeSnapshotClasses](#) and click Create VolumeSnapshotClass.
2. Enter the name of the Snapshot Class, enter csi.trident.netapp.io for the driver, and click Create.

[View shortcuts](#)

```
1 apiVersion: snapshot.storage.k8s.io/v1
2 kind: VolumeSnapshotClass
3 metadata:
4   name: trident-snapshot-class
5 driver: csi.trident.netapp.io
6 deletionPolicy: Delete
7
```

[Create](#)

[Cancel](#)

[Download](#)

3. Identify the PVC that is attached to the source VM and then create a Snapshot of that PVC. Navigate to [Storage > VolumeSnapshots](#) and click Create VolumeSnapshots.
4. Select the PVC that you want to create the Snapshot for, enter the name of the Snapshot or accept the default, and select the appropriate VolumeSnapshotClass. Then click Create.

## Create VolumeSnapshot

[Edit YAML](#)

**PersistentVolumeClaim \***

[PVC](#) rhel8-short-frog-rootdisk-28dvh

**Name \***

rhel8-short-frog-rootdisk-28dvh-snapshot

**Snapshot Class \***

[VSC](#) trident-snapshot-class

[Create](#)

[Cancel](#)

5. This creates the snapshot of the PVC at that point in time.

### Create a new VM from the snapshot

1. First, restore the Snapshot into a new PVC. Navigate to [Storage > VolumeSnapshots](#), click the ellipsis next to the Snapshot that you wish to restore, and click on 'Restore as new PVC'.
2. Enter the details of the new PVC and click Restore. This creates a new PVC.

## Restore as new PVC

When restore action for snapshot **rhel8-short-frog-rootdisk-28dvb-snapshot** is finished a new crash-consistent PVC copy will be created.

Name \*

Storage Class \*

 basic

Access Mode \*

Single User (RWO)  Shared Access (RWX)  Read Only (ROX)

Size \*



VolumeSnapshot details

Created at

 May 21, 12:46 am

Namespace

 default

Status

 Ready

API version

snapshot.storage.k8s.io/v1

Size

20 GiB

3. Next, create a new VM from this PVC. Navigate to [Workloads > Virtualization > Virtual Machines](#) and click [Create → With YAML](#).

4. In the `spec > template > spec > volumes` section, specify the new PVC created from Snapshot instead of the container disk. Provide all other details for the new VM according to your requirements.

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvc-snapshot-restore
```

5. Click Create to create the new VM.
6. After the VM is created successfully, access and verify that the new VM has the same state as that of the VM whose PVC was used to create the snapshot at the time when the snapshot was created.

## Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

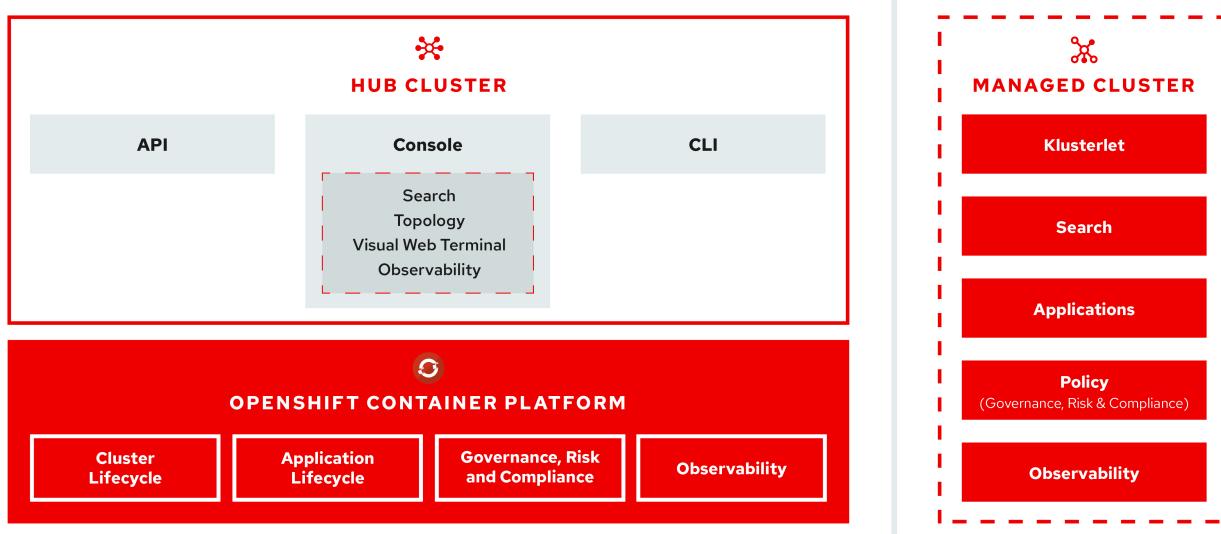
### Advanced Cluster Management for Kubernetes: Red Hat OpenShift with NetApp

As a containerized application transitions from development to production, many organizations require multiple Red Hat OpenShift clusters to support the testing and deployment of that application. In conjunction with this, organizations usually host multiple applications or workloads on OpenShift clusters. Therefore, each organization ends up managing a set of clusters, and OpenShift administrators must thus face the added challenge of managing and maintaining multiple clusters across a range of environments that span multiple on-premises data centers and public clouds. To address these challenges, Red Hat introduced Advanced Cluster Management for Kubernetes.

Red Hat Advanced Cluster Management for Kubernetes allows the users to:

1. Create, import, and manage multiple clusters across data centers and public clouds.
2. Deploy and manage applications or workloads on multiple clusters from a single console.
3. Monitor and analyse health and status of different cluster resources.
4. Monitor and enforce security compliance across multiple clusters.

Red Hat Advanced Cluster Management for Kubernetes is installed as an add-on to a Red Hat OpenShift cluster, and it uses this cluster as a central controller for all its operations. This cluster is known as hub cluster, and it exposes a management plane for the users to connect to Advanced Cluster Management. All the other OpenShift clusters that are either imported or created via the Advanced Cluster Management console are managed by the hub cluster and are called managed clusters. It installs an agent called Klusterlet on the managed clusters to connect them to the hub cluster and serve the requests for different activities related to cluster lifecycle management, application lifecycle management, observability, and security compliance.



For more information, see the documentation [here](#).

[Next: Deployment Prerequisites.](#)

## Deployment

### Deploy Advanced Cluster Management for Kubernetes: Red Hat OpenShift with NetApp

#### Prerequisites

1. A Red Hat OpenShift cluster (greater than version 4.5) for hub cluster
2. Red Hat OpenShift clusters (greater than version 4.4.3) for managed clusters
3. Cluster-admin access to Red Hat OpenShift cluster
4. A Red Hat subscription for Advanced Cluster Management for Kubernetes

Advanced Cluster Management is an add-on on for the OpenShift cluster, so there are certain requirements and restrictions on the hardware resources based on the features used across the hub and managed clusters. You need to take these issues into account when sizing the clusters. See the documentation [here](#) for more details.

Optionally, if the hub cluster has dedicated nodes for hosting infrastructure components and you would like to install Advanced Cluster Management resources only on those nodes, you need to add tolerations and selectors to those nodes accordingly. For more details, see the documentation [here](#).

[Next: Installation.](#)

### Deploy Advanced Cluster Management for Kubernetes: Red Hat OpenShift with NetApp

To install Advanced Cluster Management for Kubernetes on an OpenShift cluster, complete the following steps:

1. Choose an OpenShift cluster as the hub cluster and log into it with cluster-admin privileges.
2. Navigate to `Operators → Operators Hub` and search for `Advanced Cluster Management for Kubernetes`.

3. Select the **Advanced Cluster Management for Kubernetes** and click **Install**.

**Advanced Cluster Management for Kubernetes**

2.2.3 provided by Red Hat

**Install**

**Latest version**  
2.2.3

**Capability level**

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

**Provider type**  
Red Hat

**Provider**  
Red Hat

**Infrastructure features**  
Disconnected

**How to Install**

Use of this Red Hat product requires a licensing and subscription agreement.

4. On the **Install Operator** screen, provide the necessary details (NetApp recommends retaining the default parameters) and click **Install**.

## Install Operator

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

### Update channel \*

- release-2.0
- release-2.1
- release-2.2

### Installation mode \*

- All namespaces on the cluster (default)  
This mode is not supported by this Operator
- A specific namespace on the cluster  
Operator will be available in a single Namespace only.

### Installed Namespace \*

- Operator recommended Namespace:  open-cluster-management

 Namespace creation

Namespace open-cluster-management does not exist and will be created.

- Select a Namespace

### Approval strategy \*

- Automatic
- Manual

**Install**

**Cancel**

5. Wait for the operator installation to complete.



**Advanced Cluster Management for Kubernetes**  
2.2.3 provided by Red Hat

### Installing Operator

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace open-cluster-management](#)

6. After the operator is installed, click **Create MultiClusterHub**.



Advanced Cluster Management for Kubernetes

2.2.3 provided by Red Hat



## Installed operator - operand required

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

**MCH** MultiClusterHub ! Required

Advanced provisioning and management of OpenShift and Kubernetes clusters

[Create MultiClusterHub](#)

[View installed Operators in Namespace open-cluster-management](#)

7. On the [Create MultiClusterHub](#) screen, click [Create](#) after furnishing the details. This initiates the installation of a multi-cluster hub.

Project: open-cluster-management ▾

Advanced Cluster Management for Kubernetes > Create MultiClusterHub

### Create MultiClusterHub

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

Configure via:  Form view  YAML view

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

MultiClusterHub  
provided by Red Hat

MultiClusterHub defines the configuration for an instance of the MultiCluster Hub

Name \*

multiclusetherub

Labels

app=frontend

» Advanced configuration

[Create](#)

[Cancel](#)

8. After all the pods move to the [Running](#) state in the open-cluster-management namespace and the operator moves to the [Succeeded](#) state, Advanced Cluster Management for Kubernetes is installed.

Project: open-cluster-management ▾

## Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name	Managed Namespaces	Status	Provided APIs	⋮
 Advanced Cluster Management for Kubernetes 2.2.3 provided by Red Hat	NS open-cluster-management	<span>✓ Succeeded</span> Up to date	MultiClusterHub ClusterManager ClusterDeployment ClusterState <a href="#">View 25 more...</a>	

9. It takes some time to complete the hub installation, and, after it is done, the MultiCluster hub moves to **Running** state.

Installed Operators > Operator details

 Advanced Cluster Management for Kubernetes  
2.2.3 provided by Red Hat

Actions ▾

Details YAML Subscription Events All instances **MultiClusterHub** ClusterManager ClusterDeployment ClusterSt...  
[progress bar]

### MultiClusterHubs

Create MultiClusterHub

Name	Kind	Status	Labels	⋮
MCH multicloudclusterhub	MultiClusterHub	Phase: <span>✓ Running</span>	No labels	

10. It creates a route in the open-cluster-management namespace, connect to the URL in the route to access the Advanced Cluster Management console.

Project: open-cluster-management ▾

## Routes

Create Route

Filter ▾ Name mul

Name mul X Clear all filters

Name	Status	Location	Service	⋮
RT multicloud-console	<span>✓ Accepted</span>	<a href="https://multicloud-console.apps.ocp-vmware2.cie.netapp.com">https://multicloud-console.apps.ocp-vmware2.cie.netapp.com</a>	S management-ingress	

[Next: Features - Cluster Lifecycle Management.](#)

## Features

### Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

#### Cluster Lifecycle Management

To manage different OpenShift clusters, you can either create or import them into Advanced Cluster Management.

1. First navigate to [Automate Infrastructures > Clusters](#).
2. To create a new OpenShift cluster, complete the following steps:
  - a. Create a provider connection: Navigate to [Provider Connections](#) and click on [Add a connection](#), provide all the details corresponding to the selected provider type and click on [Add](#).

Select a provider and enter basic information

Provider \* ⓘ

aws Amazon Web Services

Connection name \* ⓘ

nik-hcl-aws

Namespace \* ⓘ

default

Configure your provider connection

Base DNS domain ⓘ

cie.netapp.com

AWS access key ID \* ⓘ

AKIATCFBZDOIASDSAH

AWS secret access key \* ⓘ

.....

Red Hat OpenShift pull secret \* ⓘ

FuS3pNbktVaHplNFc2MkZsbmtBVGN6TktmUlZXcHcxOW9teEZwQ0lYZ1d3cjJobGxJeDBQNoxlZE0yeGM5Q0ZwZk5RR2JUanlxNnNUM2lRbOFJbUFjNCIBYlpEWVZEOHitNkxTMDZPUVpoWFRHcGwtREIDQ2RSYlJRaTlxblDLT2oyQ3pVeUJfNlwicENSa2YyOUSyLWZGSFVfNA=","email":"Nikhil.kulkarni@netapp.com"},"registry.redhat.io":

SSH private key \* ⓘ

-----BEGIN OPENSSH PRIVATE KEY-----  
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbasdadssadmw9uZQAAAAAAAAAABAAAAMwAAAAtzc2gtZWQyNTUxOQAAACCLcwLgAvSIHAEp+DevIRNzaG2zkNreMIZ/UHyf0UWvAAAAAJh/wa6xf8Gu

SSH public key \* ⓘ

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIltzAuAC746agdh21cB4/4N6/VE3NobbOQ2t4zVn9QfJ/RRa8A root@nik-rhel8

- b. To create a new cluster, navigate to [Clusters](#) and click [Add a cluster > Create a cluster](#). Provide the details for the cluster and the corresponding provider, and click [Create](#).

**Configuration**

Cluster name \* [?](#)

**Distribution**

Select the type of Kubernetes distribution to use for your cluster.

Red Hat OpenShift

Select an infrastructure provider to host your Red Hat OpenShift cluster.

<input checked="" type="checkbox"/> AWS Amazon Web Services	<input type="checkbox"/> Google Cloud	<input type="checkbox"/> Microsoft Azure
<input type="checkbox"/> VMware vSphere	<input type="checkbox"/> Bare Metal	

Release image \* [?](#)

Provider connection \* [?](#)

[Add a connection](#)

- c. After the cluster is created, it appears in the cluster list with the status **Ready**.
3. To import an existing cluster, complete the following steps:
- a. Navigate to **Clusters** and click **Add a cluster > Import an existing cluster**.
  - b. Enter the name of the cluster and click **Save import and generate code**. A command to add the existing cluster is displayed.
  - c. Click **Copy command** and run the command on the cluster to be added to the hub cluster. This initiates the installation of the necessary agents on the cluster, and, after this process is complete, the cluster appears in the cluster list with status **Ready**.

Name \*

Additional labels

Once you click on "Save import and generate code", the information you entered will be used to generate the code and cannot be modified anymore. If you wish to change any information, you will have to delete and re-import this cluster.

Code generated successfully  Import saved

Run a command

**1. Copy this command**  
Click the button to have the command automatically copied to your clipboard.  
**Copy command** 

**2. Run this command with kubectl configured for your targeted cluster to start the import**  
Log in to the existing cluster in your terminal and run the command.

[View cluster](#)

[Import another](#)

4. After you create and import multiple clusters, you can monitor and manage them from a single console.

[Next: Features - Application Lifecycle Management.](#)

## Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

### Application lifecycle management

To create an application and manage it across a set of clusters,

1. Navigate to [Manage Applications](#) from the sidebar and click [Create application](#). Provide the details of the application you would like to create and click [Save](#).

Applications /

## Create an application

YAML: Off

[Cancel](#)

[Save](#)

Name\* ⓘ

Namespace\* ⓘ

Repository location for resources

Repository types

Select the type of repository where resources that you want to deploy are located

Git

URL\* ⓘ

Branch ⓘ

Path ⓘ

- After the application components are installed, the application appears in the list.

## Applications

Overview

Advanced configuration

⟳ Refresh every 15s

Last update: 7:36:23 PM

[Create application](#)

Search

Name	Namespace	Clusters	Resource	Time window	Created	
demo-app	default	Local	Git		8 days ago	

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

- The application can now be monitored and managed from the console.

Next: Features - governance and risk.

## Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

### Governance and Risk

This feature allows you to define the compliance policies for different clusters and make sure that the clusters adhere to it. You can configure the policies to either inform or remediate any deviations or violations of the rules.

1. Navigate to [Governance and Risk](#) from the sidebar.
2. To create compliance policies, click [Create Policy](#), enter the details of the policy standards, and select the clusters that should adhere to this policy. If you want to automatically remediate the violations of this policy, select the checkbox [Enforce if supported](#) and click [Create](#).

# Create policy i



YAML: Off

**Name \***

policy-complianceoperator

**Namespace \*** i

default

**Specifications \*** i

1x ComplianceOperator

**Cluster selector** i

1x local-cluster: "true"

**Standards** i

1x NIST-CSF

**Categories** i

1x PR.IP Information Protection Processes and Procedures

**Controls** i

1x PR.IP-1 Baseline Configuration

 Enforce if supported i Disable policy i

3. After all the required policies are configured, any policy or cluster violations can be monitored and remediated from Advanced Cluster Management.

## Governance and risk ⓘ

Filter

Refresh every 10s

Last update: 12:54:01 PM

Create policy

Summary 1 | Standards ▾

NIST-CSF

No violations found  
Based on the industry standards, there are no cluster or policy violations.

Policies Cluster violations

Find policies

Policy name	Namespace	Remediation	Cluster violations	Standards	Categories	Controls	Created
policy-complianceoperator	default	inform	0/1	NIST-CSF	PR.IP Information Protection Processes and Procedures	PR.IP-1 Baseline Configuration	32 minutes ago

1 - 1 of 1 ▾ << < 1 of 1 > >>

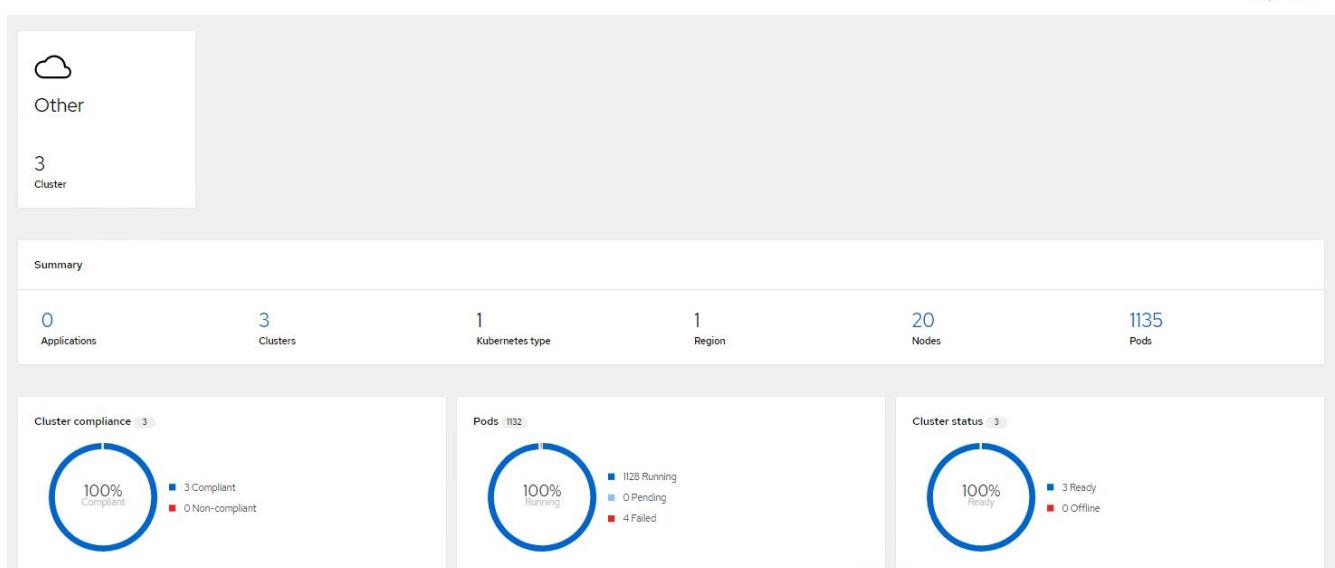
Next: Features - Observability.

## Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

### Observability

Advanced Cluster Management for Kubernetes provides a way to monitor the nodes, pods, and applications and workloads across all the clusters.

1. Navigate to [Observe Environments > Overview](#).

[Overview](#)
[+ Add provider connection](#)
[⟳ Refresh every 1m](#)  
 Last update: 12:36:03 AM


2. All pods and workloads across all clusters are monitored and sorted based on a variety of filters. Click [Pods](#) to view the corresponding data.

[Search](#)
[Saved searches ⏺](#) | [Open new search tab](#)
[3 Related cluster](#)
[673 Related secret](#)
[20 Related node](#)
[8 Related persistentvolumeclaim](#)
[8 Related persistentvolume](#)
[1 Related provisioning](#)
[2 Related searchcollector](#)
[3 Related iampolicycontroller](#)
[Show all \(38\)](#)
[Pod \(1135\)](#)
**Name**

14bbd46d68f3ddd50b9328ce6854a36807ef784dac2bded9cc20638fbpd582

⋮
**Namespace**

openshift-marketplace

**Cluster**

local-cluster

**Status**

Completed

**Restarts**

0

**Host IP**

10.61.186.27

**Pod IP**

10.129.2.215

**Created**

4 days ago

**Labels**

controller-uid=dd259738-2cce-40e2-85d3-6ccf56904ba8

3. All nodes across the clusters are monitored and analyzed based on a variety of data points. Click [Nodes](#) to get more insight into the corresponding details.

## Search

Saved searches | Open new search tab

3 Related cluster	1k Related pod	12 Related service																																
Show all (3)																																		
<b>▼ Node (20)</b> <table border="1"> <thead> <tr> <th>Name</th> <th>Cluster</th> <th>Role</th> <th>Architecture</th> <th>OS image</th> <th>CPU</th> <th>Created</th> <th>Labels</th> </tr> </thead> <tbody> <tr> <td>ocp-master-1-ocp-bare-metal.cie.netapp.com</td> <td>ocp-bare-metal</td> <td>master; worker</td> <td>amd64</td> <td>Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)</td> <td>48</td> <td>a month ago</td> <td>beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more</td> </tr> <tr> <td>ocp-master-2-ocp-bare-metal.cie.netapp.com</td> <td>ocp-bare-metal</td> <td>master; worker</td> <td>amd64</td> <td>Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)</td> <td>48</td> <td>a month ago</td> <td>beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more</td> </tr> <tr> <td>ocp-master-3-ocp-bare-metal.cie.netapp.com</td> <td>ocp-bare-metal</td> <td>master; worker</td> <td>amd64</td> <td>Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)</td> <td>48</td> <td>a month ago</td> <td>beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more</td> </tr> </tbody> </table>			Name	Cluster	Role	Architecture	OS image	CPU	Created	Labels	ocp-master-1-ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more	ocp-master-2-ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more	ocp-master-3-ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more
Name	Cluster	Role	Architecture	OS image	CPU	Created	Labels																											
ocp-master-1-ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more																											
ocp-master-2-ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more																											
ocp-master-3-ocp-bare-metal.cie.netapp.com	ocp-bare-metal	master; worker	amd64	Red Hat Enterprise Linux CoreOS 47.83.202103292105-0 (Octpa)	48	a month ago	beta.kubernetes.io/arch=amd64 beta.kubernetes.io/os=linux kubernetes.io/arch=amd64   5 more																											

4. All clusters are monitored and organized based on different cluster resources and parameters. Click **Clusters** to view cluster details.

## Search

Saved searches | Open new search tab

3k Related secret	787 Related pod	15 Related persistentvolumeclaim	17 Related node	1 Related application																														
15 Related persistentvolume	1 Related searchcollector	8 Related clusterclaim	3 Related resourcequota	5 Related identity																														
Show all (159)																																		
<b>▼ Cluster (2)</b> <table border="1"> <thead> <tr> <th>Name</th> <th>Available</th> <th>Hub accepted</th> <th>Joined</th> <th>Nodes</th> <th>Kubernetes version</th> <th>CPU</th> <th>Memory</th> <th>Console URL</th> <th>Labels</th> </tr> </thead> <tbody> <tr> <td>local-cluster</td> <td>True</td> <td>True</td> <td>True</td> <td>8</td> <td>v1.20.0+c8905da</td> <td>84</td> <td>418501Mi</td> <td>Launch</td> <td>cloud=VSphere clusterID=148632d9-69d5-4ae4-98ee-8dff886463c3 installer.name=multiclusterhub   4 more</td> </tr> <tr> <td>ocp-vmw</td> <td>True</td> <td>True</td> <td>True</td> <td>9</td> <td>v1.20.0+df9c838</td> <td>28</td> <td>111981Mi</td> <td>Launch</td> <td>cloud=VSphere clusterID=9d76ac4e-4aae-4d45-a2e8-11b6b54282fe name=ocp-vmw   1 more</td> </tr> </tbody> </table>					Name	Available	Hub accepted	Joined	Nodes	Kubernetes version	CPU	Memory	Console URL	Labels	local-cluster	True	True	True	8	v1.20.0+c8905da	84	418501Mi	Launch	cloud=VSphere clusterID=148632d9-69d5-4ae4-98ee-8dff886463c3 installer.name=multiclusterhub   4 more	ocp-vmw	True	True	True	9	v1.20.0+df9c838	28	111981Mi	Launch	cloud=VSphere clusterID=9d76ac4e-4aae-4d45-a2e8-11b6b54282fe name=ocp-vmw   1 more
Name	Available	Hub accepted	Joined	Nodes	Kubernetes version	CPU	Memory	Console URL	Labels																									
local-cluster	True	True	True	8	v1.20.0+c8905da	84	418501Mi	Launch	cloud=VSphere clusterID=148632d9-69d5-4ae4-98ee-8dff886463c3 installer.name=multiclusterhub   4 more																									
ocp-vmw	True	True	True	9	v1.20.0+df9c838	28	111981Mi	Launch	cloud=VSphere clusterID=9d76ac4e-4aae-4d45-a2e8-11b6b54282fe name=ocp-vmw   1 more																									

Next: Features - Create Resources.

## Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

### Create resources on multiple clusters

Advanced Cluster Management for Kubernetes allows users to create resources on one or more managed clusters simultaneously from the console. As an example, if you have OpenShift clusters at different sites backed with different NetApp ONTAP clusters, and want to provision PVC's at both sites, you can click the + sign on the top bar. Then select the clusters on which you want to create the PVC, paste the resource YAML, and click **Create**.

## Create resource

Cancel

Create

Clusters | Select the clusters where the resource(s) will be deployed.

2 ×

local-cluster, ▾  
ocp-vmw

Resource configuration | Enter the configuration manifest for the resource(s).

YAML

```
1 kind: PersistentVolumeClaim
2 apiVersion: v1
3 metadata:
4   name: demo-pvc
5 spec:
6   accessModes:
7     - ReadWriteOnce
8   resources:
9     requests:
10    storage: 1Gi
11  storageClassName: ocp-trident
```

## Videos and Demos: Red Hat OpenShift with NetApp

The following video demonstrate some of the capabilities documented in this document:

- [Video: Workload Migration - Red Hat OpenShift with NetApp](#)
- [Video: Installing OpenShift Virtualization - Red Hat OpenShift with NetApp](#)
- [Video: Deploying a Virtual Machine with OpenShift Virtualization - Red Hat OpenShift with NetApp](#)
- [Video: NetApp HCI for Red Hat OpenShift on Red Hat Virtualization Deployment](#)

[Next: Additional Information: Red Hat OpenShift with NetApp.](#)

## Additional Information: Red Hat OpenShift with NetApp

To learn more about the information described in this document, review the following websites:

- [NetApp Documentation](#)  
<https://docs.netapp.com/>
- [Astra Trident Documentation](#)  
<https://netapp-trident.readthedocs.io/en/stable-v21.04/>
- [NetApp Astra Control Center Documentation](#)  
<https://docs.netapp.com/us-en/astra-control-center/>
- [Red Hat OpenShift Documentation](#)  
[https://access.redhat.com/documentation/en-us/openshift\\_container\\_platform/4.7/](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.7/)

- Red Hat OpenStack Platform Documentation  
[https://access.redhat.com/documentation/en-us/red\\_hat\\_openstack\\_platform/16.1/](https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/16.1/)
- Red Hat Virtualization Documentation  
[https://access.redhat.com/documentation/en-us/red\\_hat\\_virtualization/4.4/](https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.4/)
- VMware vSphere Documentation  
<https://docs.vmware.com/>

## Google Anthos

### WP-7337: Anthos on Bare Metal

Alan Cowles and Nikhil M Kulkarni, NetApp

NetApp and Google Cloud have had a strong relationship for several years now, with NetApp first introducing cloud data services for Google Cloud with Cloud Volumes ONTAP and the Cloud Volumes Service. This relationship was then expanded by validating the NetApp HCI platform for use with Google Cloud Anthos on-premises, a hypervisor-based hybrid multi-cloud Kubernetes solution deployed on VMware vSphere. NetApp then passed Anthos Ready qualification for NetApp Trident, ONTAP, and the NFS protocol to provide dynamic persistent storage for containers.

Anthos can now be directly install on bare metal servers in a customer's environment, which adds an additional option for customers to extend Google Cloud into their local data centers without a hypervisor. Additionally, by leveraging the capabilities of NetApp ONTAP storage operating system and NetApp Trident, you can extend your platform's capabilities by integrating persistent storage for containers.

This combination allows you to realize the full potential of your servers, storage, and networking combined with the support, service levels, monthly billing, and on-demand flexibility that Google Cloud provides. Because you are using your own hardware, network, and storage, you have direct control over application scale, security, and network latency, as well as having the benefit of managed and containerized applications with Anthos on bare metal.

[Next: Solution overview.](#)

### Solution overview

#### NetApp ONTAP on NetApp AFF/FAS

NetApp AFF is a robust all-flash storage platform that provides low-latency performance, integrated data protection, multiprotocol support, and nondisruptive operations. Powered by NetApp ONTAP data management software, NetApp AFF ensures nondisruptive operations, from maintenance to upgrades to complete replacement of your storage system.

NetApp ONTAP is a powerful storage-software tool with capabilities such as an intuitive GUI, REST APIs with automation integration, AI-informed predictive analytics and corrective action, nondisruptive hardware upgrades, and cross-storage import.

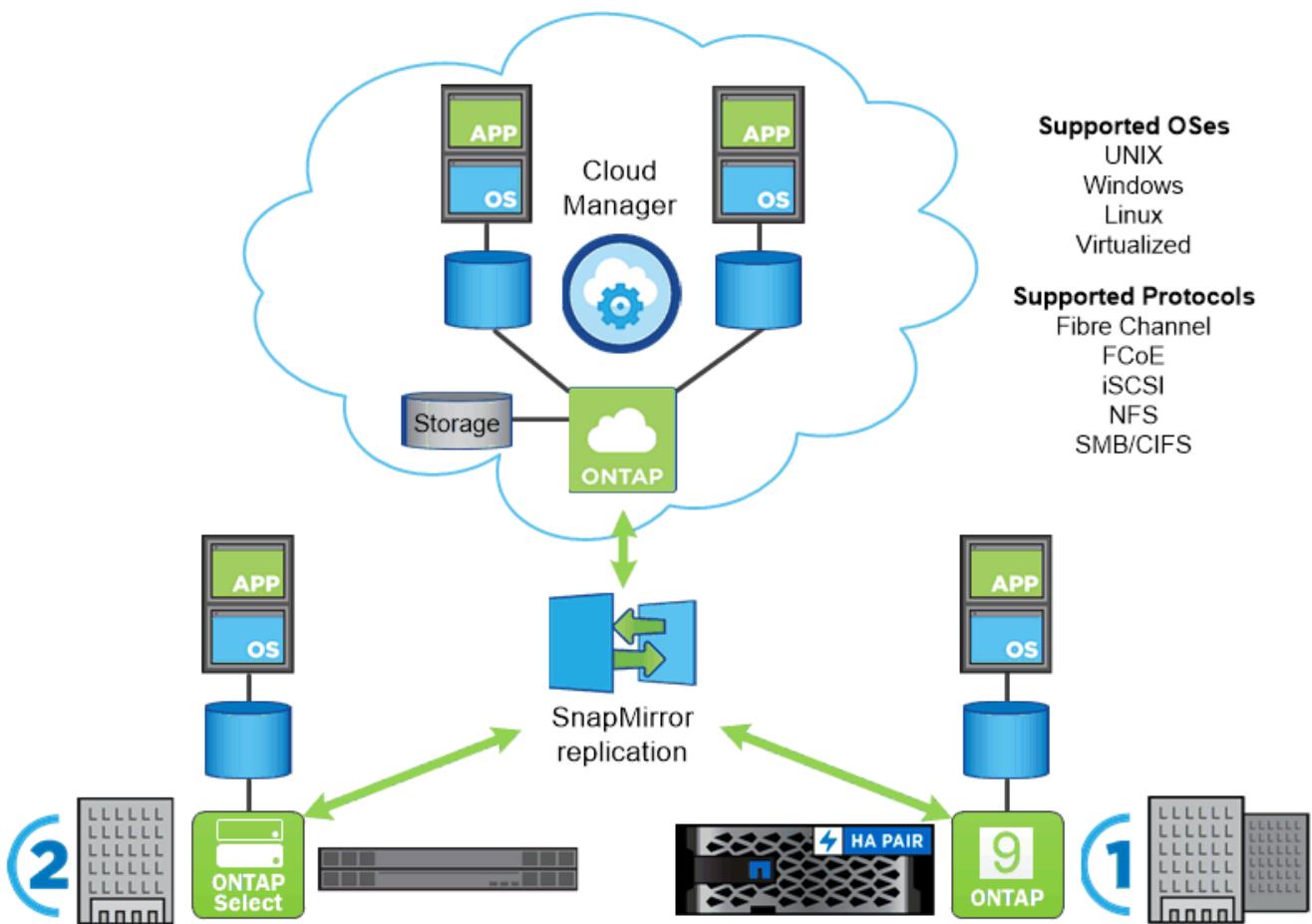
ONTAP provides the following features:

- A unified storage system with simultaneous data access and management of NFS, CIFS, iSCSI, FC,

FCoE, and FC-NVMe protocols.

- Different deployment models include on-premises on all-flash, hybrid, and all-HDD hardware configurations; VM-based storage platforms on a supported hypervisor such as ONTAP Select; and in the cloud as Cloud Volumes ONTAP.
- Increased data storage efficiency on ONTAP systems with support for automatic data tiering, inline data compression, deduplication, and compaction.
- Workload-based, QoS-controlled storage.
- Seamless integration with a public cloud for tiering and protection of data. ONTAP also provides robust data protection capabilities that sets it apart in any environment:
  - **NetApp Snapshot copies.** A fast, point-in-time backup of data using a minimal amount of disk space with no additional performance overhead.
  - **NetApp SnapMirror.** Mirrors the Snapshot copies of data from one storage system to another. ONTAP supports mirroring data to other physical platforms and cloud-native services as well.
  - **NetApp SnapLock.** Efficiently administration of non-rewritable data by writing it to special volumes that cannot be overwritten or erased for a designated period.
  - **NetApp SnapVault.** Backs up data from multiple storage systems to a central Snapshot copy that serves as a backup to all designated systems.
  - **NetApp SyncMirror.** Provides real-time, RAID-level mirroring of data to two different plexes of disks that are connected physically to the same controller.
  - **NetApp SnapRestore.** Provides fast restoration of backed-up data on demand from Snapshot copies.
  - **NetApp FlexClone.** Provides instantaneous provisioning of a fully readable and writeable copy of a NetApp volume based on a Snapshot copy. For more information about ONTAP, see the [ONTAP 9 Documentation Center](#).

NetApp ONTAP is available on-premises, virtualized, or in the cloud.



*Across the NetApp data fabric, you can count on a common set of features and fast, efficient replication across platforms. You can use the same interface and the same data management tools.*

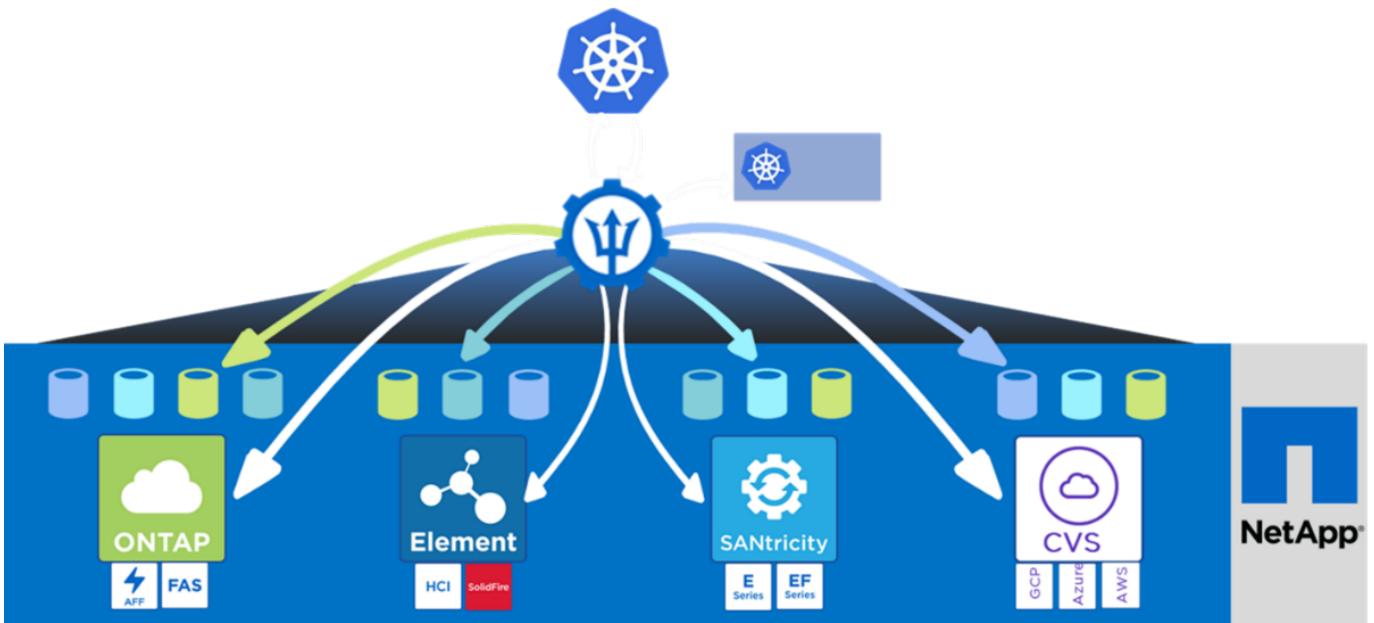
#### NetApp Trident

NetApp Trident is an open-source and fully supported storage orchestrator for containers and Kubernetes distributions, including Google Cloud Anthos. It works with the entire NetApp storage portfolio, including NetApp ONTAP software. Trident is fully CSI-compliant, and it accelerates the DevOps workflow by allowing you to provision and manage storage from your NetApp storage systems, without intervention from a storage administrator. Trident is deployed as an operator that communicates directly with the Kubernetes API endpoint to serve containers' storage requests in the form of persistent volume claims (PVCs) by creating and managing volumes on the NetApp storage system.

Persistent volumes (PVs) are provisioned based on storage classes defined in the Kubernetes environment. They use storage backends created by a storage administrator (which can be customized based on project needs) and storage system models to allow for any number of advanced storage features, such as compression, specific disk types, or QoS levels that guarantee performance.

For more information about NetApp Trident, see the [Trident](#) page.

Trident orchestrates storage from each system and service in the NetApp portfolio.



### Google Cloud's Anthos

Google Cloud's Anthos is a cloud-based Kubernetes data center solution that enables organizations to construct and manage modern hybrid-cloud infrastructures while adopting agile workflows focused on application development. Anthos on bare metal extends the capability of Anthos to run on-premises directly on physical servers without a hypervisor layer and interoperate with Anthos GKE clusters in Google Cloud.

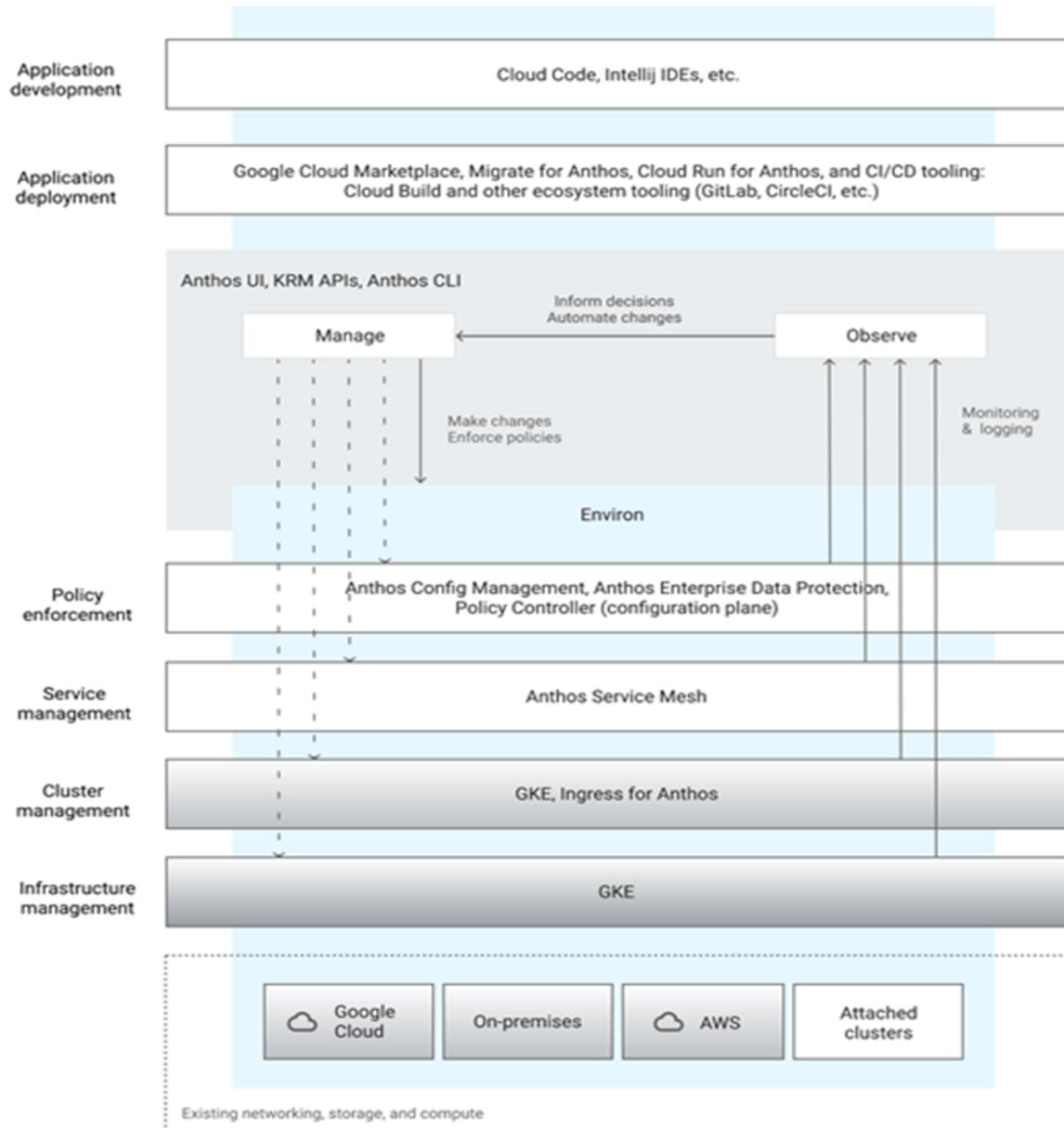
Adopting containers, service mesh, and other transformational technologies enables organizations to experience consistent application development cycles and production-ready workloads in local and cloud-based environments.

Anthos provides the following features:

- **Anthos configuration management.** Automates the policy and security of hybrid Kubernetes deployments.
- **Anthos Service Mesh.** Enhances application observability, security, and control with an Istio-powered service mesh.
- **Google Cloud Marketplace for Kubernetes applications.** A catalog of curated container applications available for easy deployment.
- **Migrate for Anthos.** Automatic migration of physical services and VMs from on-premises to the cloud. Figure 3 depicts the Anthos solution and how a deployment in an on-premises data center interconnects with infrastructure in the cloud.

For more information about Anthos, see the [Anthos website](#).

The following figure presents Google Cloud's Anthos architecture.



## Anthos on bare metal

Anthos on bare metal is an extension of GKE that is deployed in a customer's private data center. An organization can deploy the same applications designed to run in containers in Google Cloud in Anthos clusters on-premises. Anthos on bare metal runs directly on physical servers with the user's choice of underlying Linux operating system and provides customers with a full-fledged hybrid cloud environment with the capability to run at the core or edge of their data centers.

Anthos on bare metal offers the following benefits:

- **Hardware agnostic.** Customers can run Anthos on their choice of optimized hardware platform in their existing data centers.
- **Cost savings.** You can realize significant cost savings by using your own physical resources for application deployments instead of provisioning resources in the Google Cloud environment.
- **Develop then publish.** You can use on-premises deployments while applications are in development, which allows for the testing of applications in the privacy of your local data center before you make them publicly available in the cloud.

- **Better performance.** Intensive applications that demand low latency and the highest levels of performance can be run closer to the hardware.
- **Security requirements.** Customers with increased security concerns or sensitive data sets that cannot be stored in the public cloud are able to run their applications from the security of their own data centers, thereby meeting organizational requirements.
- **Management and operations.** Anthos on bare metal comes with a wide range of facilities that increase operational efficiency such as built-in networking, lifecycle management, diagnostics, health checks, logging, and monitoring.

[Next: Solution requirements.](#)

## Solution requirements

### Hardware requirements

#### Compute: bring your own server

The hardware-agnostic capabilities of Anthos on bare metal allow you to select a compute platform optimized for your use-case. Therefore, you can match your existing infrastructure and reduce capital expenditure.

The following table lists the minimum number of compute hardware components that are required to implement this solution, although the hardware models used can vary based on customer requirements.

Usage	Hardware and model	Quantity
Admin nodes	Cisco UCS B200	3
Worker nodes	HP Proliant DL360	4

#### Storage: NetApp ONTAP

The following table lists the minimum number of storage hardware components needed to implement the solution, although the hardware models used can vary based on customer requirements.

Hardware	Model	Quantity
NetApp AFF	NetApp AFF A300	2 (1 HA pair)

### Software requirements

The software versions identified in the following table were used by NetApp and our partners to validate the solution with NetApp, although the software components used can vary based on customer requirements.

Software	Purpose	Version
Ubuntu	OS on 3 Admins	20.04
	OS on Worker4	20.04
	OS on Worker3	18.04
CentOS	OS on Worker2	8.2
Red Hat Enterprise Linux	OS on Worker1	8.1
Anthos on bare metal	Container Orchestration	1.6.0

Software	Purpose	Version
NetApp ONTAP	Storage OS	9.7P8
NetApp Trident	Container Storage Management	20.10



This multi-OS environment shows the interoperability with supported OS versions of the Anthos on bare metal solution. We anticipate that customers will standardize on one or a subset of operating systems for deployment.

For Anthos on bare metal hardware and software requirements, see the [Anthos on bare metal documentation page](#).

[Next: Deployment summary.](#)

## Deployment summary

For the initial validation of this solution, NetApp partnered with World Wide Technology (WWT) to establish an environment at WWT's Advanced Technology Center (ATC). Anthos was deployed on a bare metal infrastructure using the `bmctl` tool provided by Google Cloud. The following section details the deployment used for validation purposes.

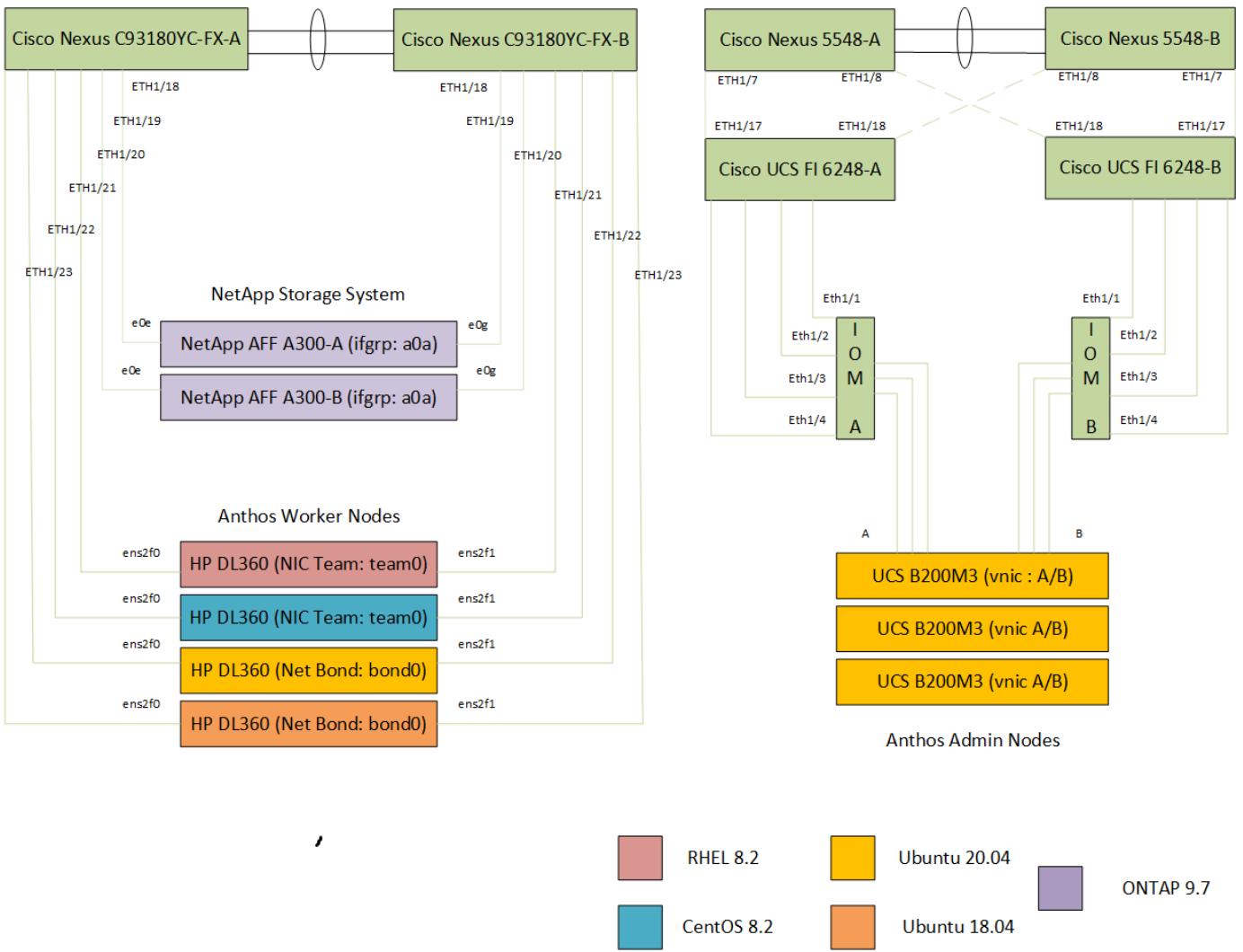
The Anthos on bare metal with NetApp solution was built as a highly available hybrid cluster with three Anthos control-plane nodes and four Anthos worker nodes.

The control-plane nodes used were Cisco UCS B200M3 blade servers hosted in a chassis and configured with a single virtual network interface card (vNIC) on each, which allowed for A/B failover at the Cisco UCS platform level for fault tolerance. The Cisco UCS chassis connected upstream to a pair of Cisco UCS 6248 fabric interconnects providing disparate paths for the separation of traffic along fabric A and fabric B. Those fabric interconnects connected upstream to a pair of Cisco Nexus 5548 data center switches that tied back to the core network at WWT.

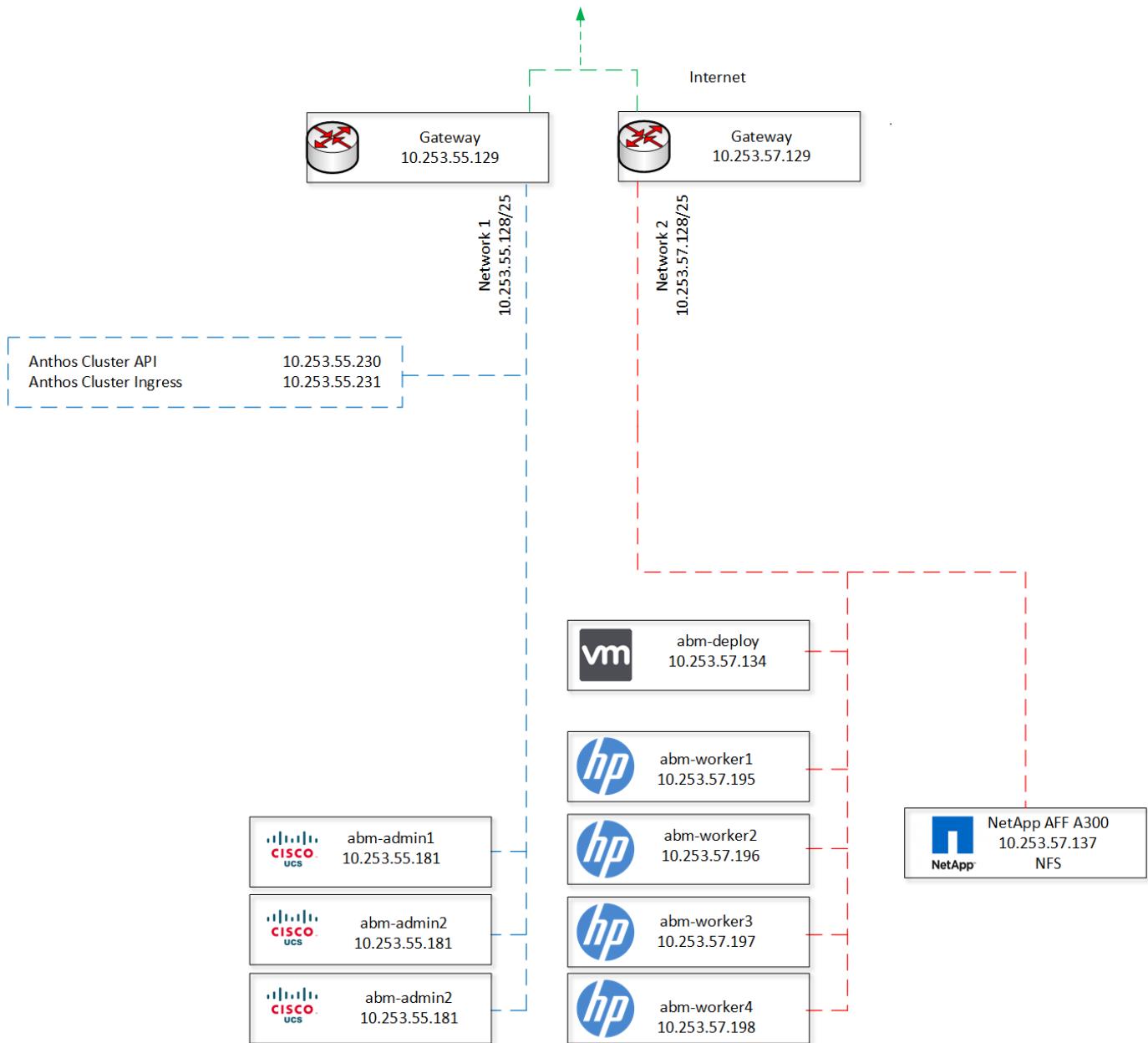
The worker nodes were HP Proliant DL360 nodes, each running one of the supported Linux distributions for Anthos on bare metal: Red Hat Enterprise Linux 8.2, CentOS 8.2, Ubuntu 20.04 LTS, or Ubuntu 18.04 LTS. The Red Hat Enterprise Linux 8 and CentOS 8 nodes were configured with NIC teams running in LACP mode and cabled to two Nexus 9k C93180YC-FX switches for fault tolerance. The Ubuntu servers were configured for network bonding in LACP mode and cabled to the same pair of Nexus 9k switches for fault tolerance.

The NetApp AFF A300 storage system running ONTAP 9.7 software was installed and connected physically to the same pair of Nexus 9k switches as the Anthos worker nodes. These network uplinks were aggregated into an interface group (a0a), and the appropriate data network VLAN was tagged to allow the worker nodes to interact with the storage system. A storage virtual machine (SVM) was created with data LIFs supporting the NFS protocol and dedicated to storage operations for Trident to provide persistent storage to the containers deployed in the Anthos on bare metal cluster. These persistent volumes were provided by NetApp Trident 20.10, the latest release of the fully supported NetApp open-source storage orchestrator for Kubernetes.

The following figure depicts a physical cabling diagram of the solution to the top of rack data center switches.



The next figure presents a logical view of the solution as deployed and validated on the hardware in the lab at the NetApp partner WWT.



Next: Solution validation.

## Solution validation

The current deployment of this solution was put through two rigorous validation processes using tools provided by the Google Cloud team. These validations include a subset of the following tests:

- Partner validation of the Anthos-ready platform:
  - Confirm that all Anthos on bare metal platform services are installed and running.
  - Scale down the physical Anthos on bare metal cluster from four worker nodes to three and then back to four.
  - Create and delete a custom namespace.
  - Create a deployment of the Nginx web server, scaling that deployment by increasing the number of replicas.

- Create an ingress for the Nginx application and verify connectivity by curling the index.html.
- Successfully clean up all test suite activities and return the cluster to a pretest state.
- Partner validation of Anthos-ready storage:
  - Create a deployment with a persistent volume claim.
  - Use NetApp Trident to provision and attach the requested persistent volume from NetApp ONTAP.
  - Validate the detach and reattach capability of persistent volumes.
  - Validate multi-attach read-only access of persistent volumes from other pods on the node.
  - Validate the offline volume resize operation.
  - Verify that the persistent volume survives a cluster-scaling operation.

[Next: Conclusion.](#)

## Conclusion

Anthos on bare metal with NetApp provides a robust platform to run container-based workloads efficiently by allowing for the customization of deployed infrastructure. Customers can use the server infrastructure and supported operating system of their choice or even deploy the solution within their existing infrastructure. The power and flexibility of these environments increases greatly through the integration of NetApp ONTAP and NetApp Trident, supporting stateful application workloads by efficiently provisioning and managing persistent storage for containers. By extending the potential of Google Cloud into their data center powered by NetApp, a customer can realize the benefits of a fully supported, highly available, easily scalable, and fully managed Kubernetes solution for development and production of their application workloads.

[Next: Where to find additional information.](#)

## Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp ONTAP Documentation Center  
<https://docs.netapp.com/ontap-9/index.jsp>
- NetApp Trident  
<https://netapp-trident.readthedocs.io/en/stable-v20.10/>
- Google Cloud's Anthos  
<https://cloud.google.com/anthos>
- Anthos on bare metal  
<https://cloud.google.com/anthos/gke/docs/bare-metal>

## NVA-1141: NetApp HCI with Anthos, design and deployment

Alan Cowles

The program solutions described in this document are designed and thoroughly tested to minimize deployment

risks and accelerate time to market.

This document is for NetApp and partner solutions engineers and customer strategic decision makers. It describes the architecture design considerations that were used to determine the specific equipment, cabling, and configurations required to support the validated workload.

NetApp HCI with Anthos is a verified, best-practice hybrid cloud architecture for the deployment of an on-premises Google Kubernetes Engine (GKE) environment in a reliable and dependable manner. This NetApp Verified Architecture reference document serves as both a design guide and a deployment validation of the Anthos solution on NetApp HCI. The architecture described in this document has been validated by subject matter experts at NetApp and Google to provide the advantage of running Anthos on NetApp HCI within your own enterprise data-center environment.

NetApp HCI, is the industry's first and leading disaggregated hybrid cloud infrastructure, providing the widely recognized benefits of hyperconverged solutions. Benefits include lower TCO and ease of acquisition, deployment, and management for virtualized workloads, while also allowing enterprise customers to independently scale compute and storage resources as needed. NetApp HCI with Anthos provides an on-premises, cloud-like experience for the deployment of containerized workloads managed by Anthos GKE on-premises. This solution provides simplified management, detailed metrics, and a range of additional functionalities that enable the easy movement of workloads deployed both on-site and in the cloud.

## Features

With NetApp HCI for Anthos, you can deploy a fully integrated, production-grade Anthos GKE environment in your on-premises data center, which allows you to take advantage of the following features:

- NetApp HCI compute and storage nodes
  - Enterprise-grade hyperconverged infrastructure designed for hybrid cloud workloads
  - NetApp Element storage software
  - Intel-based server compute nodes, including options for Nvidia GPUs
- VMware vSphere 6.7U3
  - Enterprise hypervisor solution for deployment and management of virtual infrastructures
- Anthos GKE in Google Cloud and On-Prem
  - Deploy Anthos GKE instances in Google Cloud or on NetApp HCI

The NetApp Verified Architecture program gives customers reference configurations and sizing guidance for specific workloads and use cases.

[Next: Solution Components](#)

## Solution components

The solution described in this document builds on the solid foundation of NetApp HCI, VMware vSphere, and the Anthos hybrid-cloud Kubernetes data center solution.

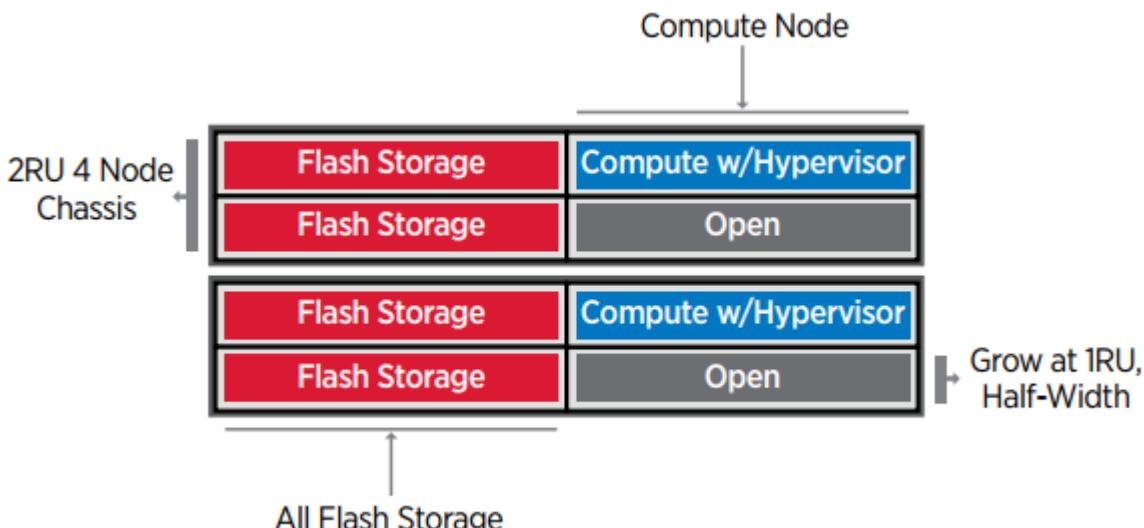
### NetApp HCI

By providing an agile turnkey infrastructure platform, NetApp HCI enables you to run enterprise-class virtualized and containerized workloads in an accelerated manner. At its core, NetApp HCI is designed to provide predictable performance, linear scalability of both compute and storage resources, and a simple deployment and management experience.

- **Predictable.** One of the biggest challenges in a multitenant environment is delivering consistent, predictable performance for all your workloads. Running multiple enterprise-grade workloads can result in resource contention, in which one workload might interfere with the performance of another. NetApp HCI alleviates this concern with storage quality-of-service (QoS) limits that are available natively with NetApp Element software. Element enables the granular control of every application and volume, helps to eliminate noisy neighbors, and satisfies enterprise performance SLAs. NetApp HCI multitenancy capabilities can help eliminate many traditional performance-related problems.
- **Flexible.** Previous generations of hyperconverged infrastructures often required fixed resource ratios, limiting deployments to four-node and eight-node configurations. NetApp HCI is a disaggregated hyperconverged infrastructure that can scale compute and storage resources independently. Independent scaling prevents costly and inefficient overprovisioning, eliminates the 10% to 30% HCI tax from controller VM overhead, and simplifies capacity and performance planning. NetApp HCI is available in mix-and-match small, medium, and large storage and compute configurations. The architectural design choices offered enable you to confidently scale on your terms, making HCI viable for core Tier 1 data center applications and platforms. NetApp HCI is architected in building blocks at either the chassis or the node level. Each chassis can hold four nodes in a mixed configuration of storage or compute nodes.
- **Simple.** A driving imperative within the IT community is to simplify deployment and automate routine tasks, eliminating the risk of user error while freeing up resources to focus on more interesting, higher-value projects. NetApp HCI can help your IT department become more agile and responsive by both simplifying deployment and ongoing management. The NetApp Deployment Engine (NDE) tool eases the configuration and deployment of physical infrastructure, including the installation of the VMware vSphere environment and the integration of the NetApp Element Plug-in for vCenter Server. With NDE, future scaling operations can be performed without difficulty.

## NetApp HCI configuration

NetApp HCI is an enterprise-scale disaggregated hybrid cloud infrastructure (HCI) solution that delivers compute and storage resources in an agile, scalable, and easy-to-manage two-rack unit (2RU) four-node building block. It can also be configured with 1RU compute and server nodes. The NetApp HCI deployment referenced in this guide consists of four NetApp HCI storage nodes and two NetApp HCI compute nodes. The compute nodes are installed as VMware ESXi hypervisors in an HA cluster without the enforcement of VMware DRS anti-affinity rules. This minimum deployment can be easily scaled to fit customer enterprise workload demands by adding additional NetApp HCI storage or compute nodes to expand available storage. The following figure depicts the minimum configuration for NetApp HCI.



The design for NetApp HCI for Anthos consists of the following components in a minimum starting configuration:

- NetApp H-Series all-flash storage nodes running NetApp Element software
- NetApp H-Series compute nodes running VMware vSphere 6.7U3

For more information about compute and storage nodes in NetApp HCI, see the [NetApp HCI Datasheet](#).

## NetApp Element software

NetApp Element software provides modular, scalable performance, with each storage node delivering guaranteed capacity and throughput to the environment. You can also specify per-volume storage QoS policies to support dedicated performance levels for even the most demanding workloads.

### iSCSI login redirection and self-healing capabilities

NetApp Element software uses the iSCSI storage protocol, a standard way to encapsulate SCSI commands on a traditional TCP/IP network. When SCSI standards change or when Ethernet network performance improves, the iSCSI storage protocol benefits without the need for any changes.

Although all storage nodes have a management IP and a storage IP, NetApp Element software advertises a single storage virtual IP address (SVIP address) for all storage traffic in the cluster. As a part of the iSCSI login process, storage can respond that the target volume has been moved to a different address, and therefore it cannot proceed with the negotiation process. The host then reissues the login request to the new address in a process that requires no host-side reconfiguration. This process is known as iSCSI login redirection.

iSCSI login redirection is a key part of the NetApp Element software cluster. When a host login request is received, the node decides which member of the cluster should handle the traffic based on IOPS and the capacity requirements for the volume. Volumes are distributed across the NetApp Element software cluster and are redistributed if a single node is handling too much traffic for its volumes or if a new node is added. Multiple copies of a given volume are allocated across the array. In this manner, if a node failure is followed by volume redistribution, there is no effect on host connectivity beyond a logout and login with redirection to the new location. With iSCSI login redirection, a NetApp Element software cluster is a self-healing, scale-out architecture that is capable of nondisruptive upgrades and operations.

## NetApp Element software cluster QoS

A NetApp Element software cluster allows QoS to be dynamically configured on a per-volume basis. You can use per-volume QoS settings to control storage performance based on SLAs that you define. The following three configurable parameters define the QoS:

- **Minimum IOPS.** The minimum number of sustained IOPS that the NetApp Element software cluster provides to a volume. The minimum IOPS configured for a volume is the guaranteed level of performance for a volume. Per-volume performance does not drop below this level.
- **Maximum IOPS.** The maximum number of sustained IOPS that the NetApp Element software cluster provides to a specific volume.
- **Burst IOPS.** The maximum number of IOPS allowed in a short burst scenario. The burst duration setting is configurable, with a default of 1 minute. If a volume has been running below the maximum IOPS level, burst credits are accumulated. When performance levels become very high and are pushed, short bursts of IOPS beyond the maximum IOPS are allowed on the volume.

## Multitenancy

Secure multitenancy is achieved with the following features:

- **Secure authentication.** The Challenge-Handshake Authentication Protocol (CHAP) is used for secure volume access. The Lightweight Directory Access Protocol (LDAP) is used for secure access to the cluster for management and reporting.
- **Volume access groups (VAGs).** Optionally, VAGs can be used in lieu of authentication, mapping any number of iSCSI initiator-specific iSCSI Qualified Names (IQNs) to one or more volumes. To access a volume in a VAG, the initiator's IQN must be in the allowed IQN list for the group of volumes.
- **Tenant virtual LANs (VLANs).** At the network level, end-to-end network security between iSCSI initiators and the NetApp Element software cluster is facilitated by using VLANs. For any VLAN that is created to isolate a workload or a tenant, NetApp Element Software creates a separate iSCSI target SVIP address that is accessible only through the specific VLAN.
- **VPN routing/forwarding (VFR)-enabled VLANs.** To further support security and scalability in the data center, NetApp Element software allows you to enable any tenant VLAN for VRF-like functionality. This feature adds these two key capabilities:
  - **L3 routing to a tenant SVIP address.** This feature allows you to situate iSCSI initiators on a separate network or VLAN from that of the NetApp Element software cluster.
  - **Overlapping or duplicate IP subnets.** This feature enables you to add a template to tenant environments, allowing each respective tenant VLAN to be assigned IP addresses from the same IP subnet. This capability can be useful for service provider environments where scale and preservation of IP-space are important.

## Enterprise storage efficiencies

The NetApp Element software cluster increases overall storage efficiency and performance. The following features are performed inline, are always on, and require no manual configuration by the user:

- **Deduplication.** The system only stores unique 4K blocks. Any duplicate 4K blocks are automatically associated to an already stored version of the data. Data is on block drives and is mirrored by using Element Helix data protection. This system significantly reduces capacity consumption and write operations within the system.
- **Compression.** Compression is performed inline before data is written to NVRAM. Data is compressed, stored in 4K blocks, and remains compressed in the system. This compression significantly reduces capacity consumption, write operations, and bandwidth consumption across the cluster.
- **Thin provisioning.** This capability provides the right amount of storage at the time that you need it, eliminating capacity consumption that caused by overprovisioned volumes or underutilized volumes.
- **Helix.** The metadata for an individual volume is stored on a metadata drive and is replicated to a secondary metadata drive for redundancy.

**Note:** Element was designed for automation. All the storage features mentioned above can be managed with APIs. These APIs are the only method that the UI uses to control the system whether actions are performed directly through Element or through the vSphere plug-in for Element.

## VMware vSphere

VMware vSphere is the industry leading virtualization solution built on VMware ESXi hypervisors and managed by vCenter Server, which provides advanced functionality often required for enterprise datacenters. When using the NDE with NetApp HCI, a VMware vSphere environment is configured and installed. The following features are available after the environment is deployed:

- **Centralized Management.** Through vSphere, individual hypervisors can be grouped into data centers and combined into clusters, allowing for advanced organization to ease the overall management of resources.
- **VMware HA.** This feature allows virtual guests to restart automatically if their host becomes unavailable. By enabling this feature, virtual guests become fault tolerant, and virtual infrastructures experience minimal disruption when there are physical failures in the environment.
- **VMware Distributed Resource Scheduler (DRS).** VMware vMotion allows for the movement of guests between hosts nondisruptively when certain user-defined thresholds are met. This capability makes the virtual guests in an environment highly available.
- **vSphere Distributed Switch (vDS).** A virtual switch is controlled by the vCenter server, enabling centralized configuration and management of connectivity for each host by creating port groups that map to the physical interfaces on each host.

## **Anthos**

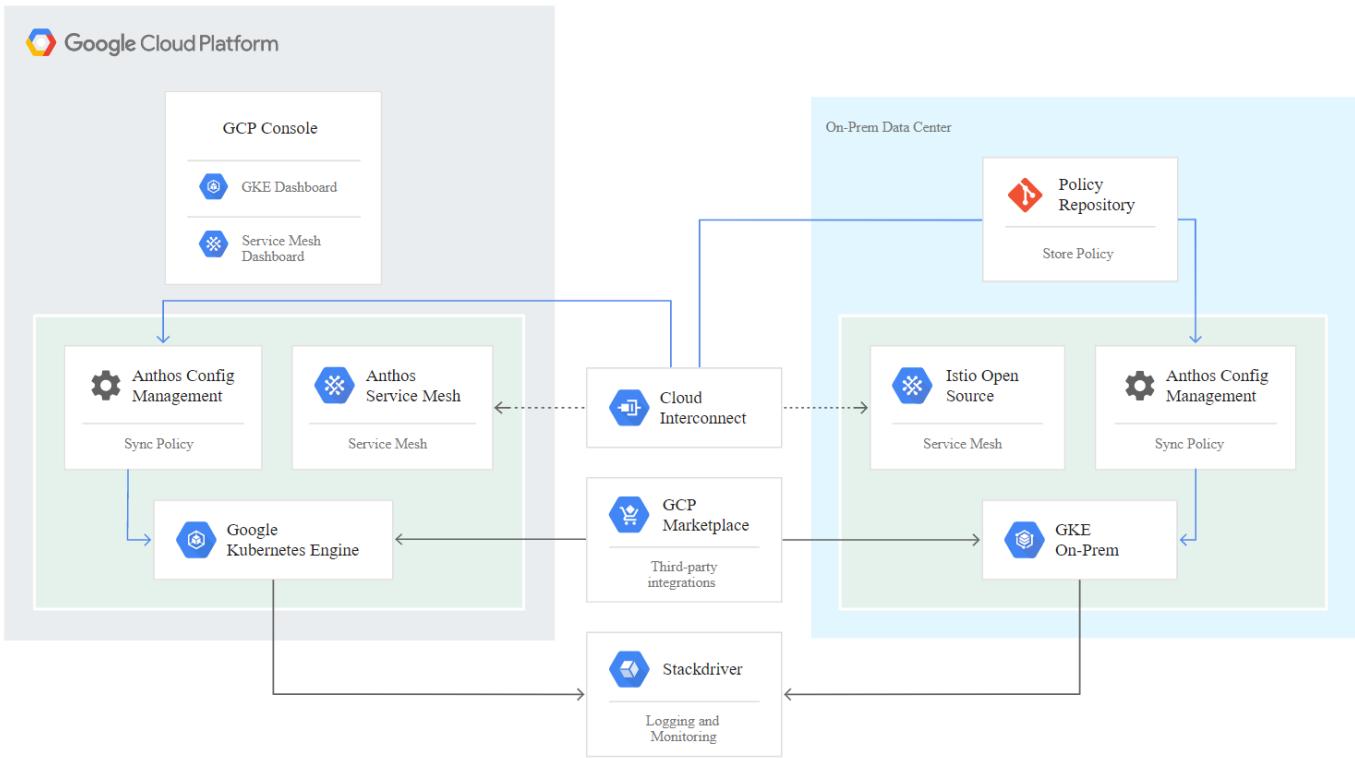
Anthos is a hybrid-cloud Kubernetes data center solution that enables organizations to construct and manage modern hybrid-cloud infrastructures, while adopting agile workflows focused on application development. Anthos on VMware, a solution built on open-source technologies, runs on-premises in a VMware vSphere-based infrastructure, which can connect and interoperate with Anthos GKE in Google Cloud.

Adopting containers, service mesh, and other transformational technologies enables organizations to experience consistent application development cycles and production-ready workloads in local and cloud-based environments. The following figure depicts the Anthos solution and how a deployment in an on-premises data center interconnects with infrastructure in the cloud.

For more information about Anthos, see the Anthos website located [here](#).

Anthos provides the following features:

- **Anthos configuration management.** Automates the policy and security of hybrid Kubernetes deployments.
- **Anthos Service Mesh.** Enhances application observability, security, and control with an Istio-powered service mesh.
- **Google Cloud Marketplace for Kubernetes Applications.** A catalog of curated container applications available for easy deployment.
- **Migrate for Anthos.** Automatic migration of physical services and VMs from on-premises to the cloud.
- **Stackdriver.** Management service offered by Google for logging and monitoring cloud instances.



## Containers and Kubernetes orchestration

Container technology has been available to developers for a long time. However, it has only recently become a core concept in data center architecture and design as more enterprises have adopted application-specific workload requirements.

A traditional development environment requires a dedicated development host deployed on either a bare-metal or virtual server. Such environments require each application to have its own dedicated machine, complete with operating system (OS) and networking connectivity. These machines often must be managed by the enterprise system administration team, who must account for the application versions installed as well as host OS patches. In contrast, containers by design require less overhead to deploy. All that is needed is the packaging of application code and supporting libraries together, because all other services depend on the host OS. Rather than managing a complete virtual machine (VM) environment, developers can instead focus on the application development process.

As container technology began to find appeal in the enterprise landscape, many enterprise features, such as fault tolerance and application scaling, were both requested and expected. In response, Google partnered with the Linux Foundation to form the Cloud Native Computing Foundation (CNCF). Together, they introduced Kubernetes (K8s), an open-source platform for orchestrating and managing containers. Kubernetes was designed by Google to be a successor to both the Omega and Borg container management platforms that had been used in their data centers in the previous decade.

## Anthos GKE

Anthos GKE is a certified distribution of Kubernetes in the Google Cloud. It allows end users to easily deploy managed, production-ready Kubernetes clusters, enabling developers to focus primarily on application development rather than on the management of their environment. Deploying Kubernetes clusters in Anthos GKE offers the following benefits:

- Simplifying deployment of applications.** Anthos GKE allows for rapid development, deployment, and updates of applications and services. By providing simple descriptions of the expected system resources

(compute, memory, and storage) required by the application containers, the Kubernetes Engine automatically provisions and manages the lifecycle of the cluster environment.

- **Ensuring availability of clusters.** The environment is made extremely accessible and easy to manage by using the dashboard built into the Google Cloud console. Anthos GKE clusters are continually monitored by Google Site Reliability Engineers (SREs) to make sure that clusters behave as expected by collecting regular metrics and observing the use of assigned system resources. A user can also leverage available health checks to make sure that their deployed applications are highly available and that they can recover easily should something go awry.
- **Securing clusters in Google Cloud.** An end user can ensure that clusters are secure and accessible by customizing network policies available from Google Cloud's Global Virtual Private Cloud. Public services can be placed behind a single global IP address for load balancing purposes. A single IP can help provide high availability for applications and protect against distributed denial of service (DDOS) and other forms of attacks that might hinder service performance.
- **Easily scaling to meet requirements.** An end user can enable auto-scaling on their cluster to easily counter both planned and unexpected increases in application demands. Auto-scaling helps make sure that system resources are always available by increasing capacity during high-demand windows. It also allows the cluster to return to its previous state and size after peak demand wanes.

## Anthos on VMware

Anthos on VMware is an extension of the Google Kubernetes Engine that is deployed in an end user's private data center. An organization can deploy the same applications designed to run in containers in Google Cloud in Kubernetes clusters on premises. Anthos on VMware offers the following benefits:

- **Cost savings.** End users can realize significant cost savings by utilizing their own physical resources for their application deployments instead of provisioning resources in their Google Cloud environment.
- **Develop, then publish.** On-premises deployments can be used while applications are in development, which allows for testing of applications in the privacy of a local data center before being made publicly available in the cloud.
- **Security requirements.** Customers with increased security concerns or sensitive data sets that cannot be stored in the public cloud are able to run their applications from the security of their own data centers, thereby meeting organizational requirements.

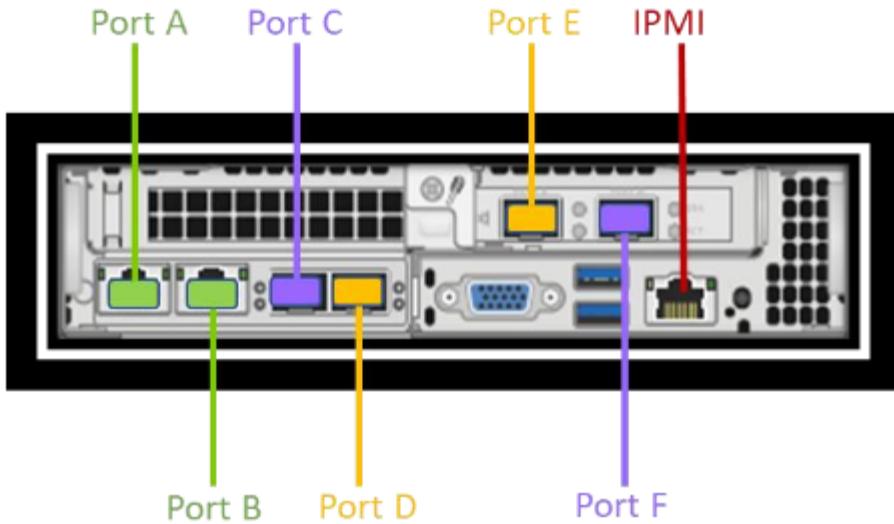
[Next: Design Considerations](#)

## Design considerations

This section describes the design considerations necessary for the successful deployment of the NetApp HCI Anthos solution.

### Port identification

NetApp HCI consists of NetApp H-Series nodes dedicated to either compute or storage. Both node configurations are available with two 1GbE ports (ports A and B) and two 10/25 GbE ports (ports C and D) on board. The compute nodes have additional 10/25GbE ports (ports E and F) available in the first mezzanine slot. Each node also has an additional out-of-band management port that supports Intelligent Platform Management Interface (IPMI) functionality. The following figure identifies each of these ports on the rear of an H410C node.



### Network design

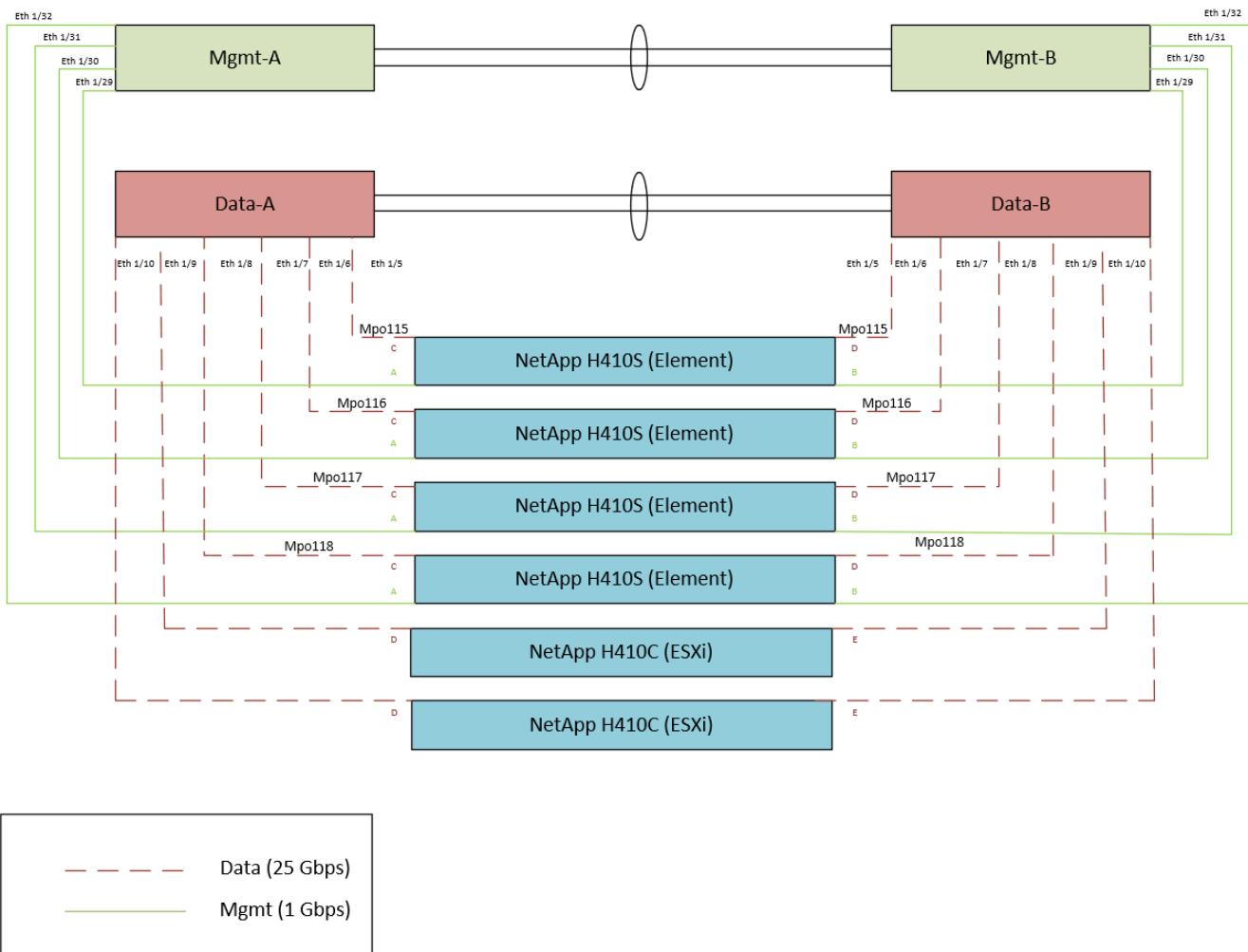
The NetApp HCI with Anthos solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality.

### Cabling storage nodes

The management ports A and B must be active on each storage node to run NDE, configure the NetApp HCI cluster, and provide management accessibility to Element after the solution is deployed. The two 25Gbps ports (C and D) should be connected, one to each data switch, to provide physical fault tolerance. The switch ports should be configured for multi-chassis link aggregation (MLAG) and the data ports on the node should be configured for LACP with jumbo-frames support enabled. The IPMI ports on each node can be used to remotely manage the node after it is installed in a data center. With IPMI, the node can be accessed with a web-browser-based console to run the initial installation, run diagnostics, and reboot or shut down the node if necessary.

### Cabling compute nodes

The 25Gbps ports on the compute nodes are cabled with one onboard port (C) cabled to one data switch, and an additional port from the PCI slot (E) cabled to the second switch to provide physical fault tolerance. These ports should be configured to support jumbo frames. Connectivity for the node is managed by the vDS after VMware vSphere is deployed in the environment. The IPMI ports can also be used to remotely manage the node after it is installed in a data center. With IPMI, the node can be accessed via a web-browser-based console to run diagnostics and to be rebooted or shut down if necessary. The following figure provides a reference for network cabling.



## VLAN requirements

The solution is designed to logically separate network traffic for different purposes by using Virtual Local Area Networks (VLANs). NetApp HCI requires a minimum of three network segments. However, this configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution, as well as the specific VLAN IDs that are used later in the validated architecture deployment.

VLANs	Purpose	VLAN used
Out-of-band management	Management for HCI nodes	16
In-band management	Management for HCI nodes and infrastructure virtual guests	3480
Storage network	Storage network for NetApp Element	3481
vMotion network	Network for VMware vMotion	3482
VM network	Network for virtual guests	1172

## Network infrastructure support resources

The following infrastructure should be in place prior to the deployment of the Anthos on NetApp HCI solution:

- A DHCP server providing addresses for both the in-band management network and the VM network. The DHCP pool must be large enough to support at least 10 VMs for an initial deployment and should be scaled as necessary.
- At least one DNS server providing full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- Outbound internet connectivity for both the in-band management network and the VM network.

## **Best practices**

The details in this document describe a deployment of Anthos on VMware that meets the minimum requirements for deployment. Prior to deploying the solution in a production environment, you should use the information presented in this Best Practice section.

### **Install a second SeeSaw load balancer**

In a production environment, it is a best practice to avoid single points of failure in your environment. For this validation, a single Seesaw bundled load balancer was allocated to the admin and each user cluster deployed. While this works fine for a simple validation, loss of communication with the control plane VIP for a cluster can make a cluster inaccessible or unable to be managed from the admin workstation or the Google Cloud console. By deploying HA Seesaw load balancers, it is possible to make sure disruptions do not happen. The setup procedures and additional requirements to enable this function are not described in detail in this document, however full instructions can be found [here](#).

### **Install a second F5 Big-IP Virtual Edition appliance**

In a production environment, it is a best practice to avoid single points of failure in your environment. For this validation, a single F5 BIG-IP Virtual Edition Load Balancer appliance was used to validate connectivity to the control plane and the ingress VIP addresses for the Anthos on VMware clusters. Although this works fine for a simple validation, loss of communication with the control plane VIP for a cluster can make a cluster inaccessible or unable to be managed from the admin workstation or the Google Cloud console. F5 BIG-IP Virtual Edition supports application-based HA to make sure disruptions do not happen. Although this issue is mentioned briefly, setup procedures for this functionality are not described in detail in this document. However, NetApp recommends investigating this feature further before deploying the NetApp HCI for Anthos solution into production.

### **Enable VMware vSphere DRS and configure anti-affinity rules**

VMware vSphere provides a feature that makes sure that no single node in the cluster runs low on physical resources available to virtual guests. The Distributed Resource Scheduler (DRS) can be configured on vSphere clusters consisting of at least three ESXi nodes. The NetApp HCI minimum configuration described in this deployment guide consists of two compute nodes and is unable to make use of this feature. As a result of this limitation, we were also forced to disable anti-affinity rules for the Anthos on VMware clusters that we deployed.

Anti-affinity rules ensure that all masters or all workers for a specific user cluster run on different nodes so that a single node failure cannot disable an entire user cluster or the pods that it is hosting. The NetApp HCI system is both easily and rapidly scalable and the minimum deployment described in this validation has two open chassis slots for immediate expansion of HCI 410C nodes. Therefore, NetApp suggests adding additional compute nodes into the empty chassis slots prior to deploying the solution into production and enabling DRS with anti-affinity rules.

## Use SnapMirror to copy data remotely for disaster recovery

NetApp Element storage systems can use NetApp SnapMirror technology to replicate storage volumes to systems running the NetApp ONTAP system, including AFF, FAS, and Cloud Volumes ONTAP. You can set up regularly scheduled SnapMirror operations to back up the VMware datastores and restore from a remote site in the event of a disaster. It is also possible to use SnapMirror to back up or migrate the persistent volumes provisioned by Trident and reattach them to Kubernetes clusters deployed in other environments and in the cloud.

[Next: Hardware and Software Requirements](#)

### Hardware and software requirements

This section describes the hardware and software requirements for the NetApp HCI and Anthos solution.

#### Hardware requirements

The following table lists the minimum number of hardware components that are required to implement the solution. The hardware components that are used in specific implementations of the solution might vary based on customer requirements.

Hardware	Model	Quantity
NetApp HCI compute nodes	NetApp H410C	2
NetApp HCI storage nodes	NetApp H410S	2
Data switches	Cisco Nexus 3048	2
Management switches	Mellanox NS2010	2

#### Software requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any implementation of the solution might vary based on customer requirements.

Software	Purpose	Version
NetApp HCI	Infrastructure (compute/storage)	1.8P1
VMware vSphere	Virtualization	6.7U3
Anthos on VMware	Container orchestration	1.7
F5 Big-IP Virtual Edition	Load balancing	15.0.1
NetApp Trident	Storage management	21.04

[Next: Deployment steps.](#)

### Deployment Steps

This section provides detailed protocols for implementing the NetApp HCI solution for Anthos.

This deployment is divided into the following high-level tasks:

1. [Configure management switches](#)
2. [Configure data switches](#)
3. [Deploy NetApp HCI with the NetApp Deployment Engine](#)

4. Configure the vCenter Server
5. Deploy and configure the F5 Big-IP Virtual Edition Appliance
6. Complete Anthos prerequisites
7. Deploy the Anthos admin workstation
8. Deploy the admin cluster
9. Deploy user clusters
10. Enable access to cluster with the GKE console
11. Install and configure NetApp Trident storage provisioner

Next: Configure management switches.

### 1. Configure management switches

Cisco Nexus 3048 switches are used in this deployment procedure to provide 1Gbps connectivity for in- and out-of-band management of the compute and storage nodes. These steps begin after the switches have been racked, powered, and put through the initial setup process. To configure the switches to provide management connectivity to the infrastructure, complete the following steps:

#### Enable advanced features for Cisco Nexus

Run the following commands on each Cisco Nexus 3048 switch to configure advanced features:

1. Enter configuration mode.

```
Switch-01# configure terminal
```

2. Enable VLAN functionality.

```
Switch-01(config)# feature interface-vlan
```

3. Enable LACP.

```
Switch-01(config)# feature lacp
```

4. Enable virtual port channels (vPCs).

```
Switch-01(config)# feature vpc
```

5. Set the global port-channel load-balancing configuration.

```
Switch-01(config)# port-channel load-balance src-dst ip-l4port
```

6. Perform the global spanning-tree configuration.

```
Switch-01(config)# spanning-tree port type network default
Switch-01(config)# spanning-tree port type edge bpduguard default
```

## Configure ports on the switch for in-band management

1. Run the following commands to create VLANs for management purposes.

```
Switch-01(config)# vlan 2
Switch-01(config-vlan)# Name Native_VLAN
Switch-01(config-vlan)# vlan 16
Switch-01(config-vlan)# Name OOB_Network
Switch-01(config-vlan)# vlan 3480
Switch-01(config-vlan)# Name MGMT_Network
Switch-01(config-vlan)# exit
```

2. Configure the ports ETH1/29-32 as VLAN trunk ports that connect to management interfaces on each HCI storage node.

```
Switch-01(config)# int eth 1/29
Switch-01(config-if)# description HCI-STG-01 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 3480
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# int eth 1/30
Switch-01(config-if)# description HCI-STG-02 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 3480
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# int eth 1/31
Switch-01(config-if)# description HCI-STG-03 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 3480
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# int eth 1/32
Switch-01(config-if)# description HCI-STG-04 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 3480
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# exit
```

## Configure ports on the switch for out-of-band management

1. Run the following commands to configure the ports for cabling the IPMI interfaces on each HCI node.

```
Switch-01(config)# int eth 1/13
Switch-01(config-if)# description HCI-CMP-01 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# int eth 1/14
Switch-01(config-if)# description HCI-STG-01 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# int eth 1/15
Switch-01(config-if)# description HCI-STG-03 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# exit
```



In the validated configuration, we cabled odd-node IPMI interfaces to Switch-01, and even-node IPMI interfaces to Switch-02.

## Create a vPC domain to ensure fault tolerance

1. Activate the ports used for the vPC peer-link between the two switches.

```
Switch-01(config)# int eth 1/1
Switch-01(config-if)# description vPC peer-link Switch-02 1/1
Switch-01(config-if)# int eth 1/2
Switch-01(config-if)# description vPC peer-link Switch-02 1/2
Switch-01(config-if)# exit
```

2. Perform the vPC global configuration.

```
Switch-01(config)# vpc domain 1
Switch-01(config-vpc-domain)# role priority 10
Switch-01(config-vpc-domain)# peer-keepalive destination <switch-
02_mgmt_address> source <switch-01_mgmt_address> vrf management
Switch-01(config-vpc-domain)# peer-gateway
Switch-01(config-vpc-domain)# auto recovery
Switch-01(config-vpc-domain)# ip arp synchronize
Switch-01(config-vpc-domain)# int eth 1/1-2
Switch-01(config-vpc-domain)# channel-group 10 mode active
Switch-01(config-vpc-domain)# int Po10
Switch-01(config-if)# description vPC peer-link
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 16,3480
Switch-01(config-if)# spanning-tree port type network
Switch-01(config-if)# vpc peer-link
Switch-01(config-if)# exit
```

[Next: Configure Data Switches](#)

## 2. Configure Data Switches

Mellanox SN2010 switches provide 25Gbps connectivity for the data plane of the compute and storage nodes. To configure the switches to provide data connectivity to the infrastructure, complete the following steps:

### Create MLAG cluster to provide fault tolerance

1. Run the following commands on each Mellanox SN210 switch for general configuration:

- Enter configuration mode.

```
Switch-01 enable
Switch-01 configure terminal
```

- Enable the LACP required for the Inter-Peer Link (IPL).

```
Switch-01 (config) # lacp
```

- Enable the Link Layer Discovery Protocol (LLDP).

```
Switch-01 (config) # lldp
```

- Enable IP routing.

```
Switch-01 (config) # ip routing
```

- e. Enable the MLAG protocol.

```
Switch-01 (config) # protocol mlag
```

- f. Enable global QoS.

```
Switch-01 (config) # dcb priority-flow-control enable force
```

2. For MLAG to function, the switches must be made peers to each other through an IPL. This should consist of two or more physical links for redundancy. The MTU for the IPL is set for jumbo frames (9216), and all VLANs are enabled by default. Run the following commands on each switch in the domain:

- a. Create port channel 10 for the IPL.

```
Switch-01 (config) # interface port-channel 10
Switch-01 (config interface port-channel 10) # description IPL
Switch-01 (config interface port-channel 10) # exit
```

- b. Add interfaces ETH 1/20 and 1/22 to the port channel.

```
Switch-01 (config) # interface ethernet 1/20 channel-group 10 mode
active
Switch-01 (config) # interface ethernet 1/20 description ISL-SWB_01
Switch-01 (config) # interface ethernet 1/22 channel-group 10 mode
active
Switch-01 (config) # interface ethernet 1/22 description ISL-SWB_02
```

- c. Create a VLAN outside of the standard range dedicated to IPL traffic.

```
Switch-01 (config) # vlan 4000
Switch-01 (config vlan 4000) # name IPL VLAN
Switch-01 (config vlan 4000) # exit
```

- d. Define the port channel as the IPL.

```
Switch-01 (config) # interface port-channel 10 ipl 1
Switch-01 (config) # interface port-channel 10 dcb priority-flow-
control mode on force
```

- e. Set an IP for each IPL member (non-routable; it is not advertised outside of the switch).

```
Switch-01 (config) # interface vlan 4000
Switch-01 (config vlan 4000) # ip address 10.0.0.1 255.255.255.0
Switch-01 (config vlan 4000) # ipl 1 peer-address 10.0.0.2
Switch-01 (config vlan 4000) # exit
```

3. Create a unique MLAG domain name for the two switches and assign an MLAG virtual IP (VIP). This IP is used for keep-alive heartbeat messages between the two switches. Run these commands on each switch in the domain:

- a. Create the MLAG domain and set the IP address and subnet.

```
Switch-01 (config) # mlag-vip MLAG-VIP-DOM ip a.b.c.d /24 force
```

- b. Create a virtual MAC address for the system MLAG.

```
Switch-01 (config) # mlag system-mac AA:BB:CC:DD:EE:FF
```

- c. Configure the MLAG domain so that it is active globally.

```
Switch-01 (config) # no mlag shutdown
```



The IP used for the MLAG VIP must be in the same subnet as the switch management network (mgmt0).



The MAC address used can be any unicast MAC address and must be set to the same value on both switches in the MLAG domain.

## Configure ports to connect to storage and compute hosts

1. Create each of the VLANs needed to support the services for NetApp HCI. Run these commands on each switch in the domain:

- a. Create VLANs.

```
Switch-01 (config) # vlan 1172
Switch-01 (config vlan 1172) exit
Switch-01 (config) # vlan 3480-3482
Switch-01 (config vlan 3480-3482) exit
```

- b. Create names for each VLAN for easier accounting.

```
Switch-01 (config) # vlan 1172 name "VM_Network"
Switch-01 (config) # vlan 3480 name "MGMT_Network"
Switch-01 (config) # vlan 3481 name "Storage_Network"
Switch-01 (config) # vlan 3482 name "vMotion_Network"
+
```

2. Create hybrid VLAN ports on ports ETH1/9-10 so that you can tag the appropriate VLANs for the NetApp HCI compute nodes.

- Select the ports you want to work with.

```
Switch-01 (config) # interface ethernet 1/9-1/10
```

- Set the MTU for each port.

```
Switch-01 (config interface ethernet 1/9-1/10) # mtu 9216 force
```

- Modify spanning-tree settings for each port.

```
Switch-01 (config interface ethernet 1/9-1/10) # spanning-tree
bpdufilter enable
Switch-01 (config interface ethernet 1/9-1/10) # spanning-tree port
type edge
Switch-01 (config interface ethernet 1/9-1/10) # spanning-tree
bpduguard enable
```

- Set the switchport mode to hybrid.

```
Switch-01 (config interface ethernet 1/9-1/10 ) # switchport mode
hybrid
Switch-01 (config interface ethernet 1/9-1/10 ) # exit
```

- Create descriptions for each port being modified.

```
Switch-01 (config) # interface ethernet 1/9 description HCI-CMP-01
PortD
Switch-01 (config) # interface ethernet 1/10 description HCI-CMP-02
PortD
```

- Tag the appropriate VLANs for the NetApp HCI environment.

```
Switch-01 (config) # interface ethernet 1/9 switchport hybrid  
allowed-vlan add 1172  
Switch-01 (config) # interface ethernet 1/9 switchport hybrid  
allowed-vlan add 3480-3482  
Switch-01 (config) # interface ethernet 1/10 switchport hybrid  
allowed-vlan add 1172  
Switch-01 (config) # interface ethernet 1/10 switchport hybrid  
allowed-vlan add 3480-3482
```

3. Create MLAG interfaces and hybrid VLAN ports on ports ETH1/5-8 so that you can distribute connectivity between the switches and tag the appropriate VLANs for the NetApp HCI storage nodes.

- a. Select the ports that you want to work with.

```
Switch-01 (config) # interface ethernet 1/5-1/8
```

- b. Set the MTU for each port.

```
Switch-01 (config interface ethernet 1/5-1/8) # mtu 9216 force
```

- c. Modify spanning tree settings for each port.

```
Switch-01 (config interface ethernet 1/5-1/8) # spanning-tree  
bpdufilter enable  
Switch-01 (config interface ethernet 1/5-1/8) # spanning-tree port  
type edge  
Switch-01 (config interface ethernet 1/5-1/8) # spanning-tree  
bpduguard enable
```

- d. Set the switchport mode to hybrid.

```
Switch-01 (config interface ethernet 1/5-1/8 ) # switchport mode  
hybrid  
Switch-01 (config interface ethernet 1/5-1/8 ) # exit
```

- e. Create descriptions for each port being modified.

```
Switch-01 (config) # interface ethernet 1/5 description HCI-STG-01
PortD
Switch-01 (config) # interface ethernet 1/6 description HCI-STG-02
PortD
Switch-01 (config) # interface ethernet 1/7 description HCI-STG-03
PortD
Switch-01 (config) # interface ethernet 1/8 description HCI-STG-04
PortD
```

f. Create and configure the MLAG port channels.

```
Switch-01 (config) # interface mlag-port-channel 115-118
Switch-01 (config) interface mlag-port-channel 115-118) # exit
Switch-01 (config) # interface mlag-port-channel 115-118 no shutdown
Switch-01 (config) # interface mlag-port-channel 115-118 mtu 9216
force
Switch-01 (config) # interface mlag-port-channel 115-118 lacp-
individual enable force
Switch-01 (config) # interface ethernet 1/5-1/8 lacp port-priority 10
Switch-01 (config) # interface ethernet 1/5-1/8 lacp rate fast
Switch-01 (config) # interface ethernet 1/5 mlag-channel-group 115
mode active
Switch-01 (config) # interface ethernet 1/6 mlag-channel-group 116
mode active
Switch-01 (config) # interface ethernet 1/7 mlag-channel-group 117
mode active
Switch-01 (config) # interface ethernet 1/8 mlag-channel-group 118
mode active
```

g. Tag the appropriate VLANs for the storage environment.

```
Switch-01 (config) # interface mlag-port-channel 115-118 switchport
mode hybrid
Switch-01 (config) # interface mlag-port-channel 115 switchport
hybrid allowed-vlan add 1172 Switch-01 (config) # interface mlag-
port-channel 116 switchport hybrid allowed-vlan add 1172
Switch-01 (config) # interface mlag-port-channel 117 switchport
hybrid allowed-vlan add 1172
Switch-01 (config) # interface mlag-port-channel 118 switchport
hybrid allowed-vlan add 1172
Switch-01 (config) # interface mlag-port-channel 115 switchport
hybrid allowed-vlan add 3481
Switch-01 (config) # interface mlag-port-channel 116 switchport
hybrid allowed-vlan add 3481
Switch-01 (config) # interface mlag-port-channel 117 switchport
hybrid allowed-vlan add 3481
Switch-01 (config) # interface mlag-port-channel 118 switchport
hybrid allowed-vlan add 3481
```



The configurations in this section must also be run on the second switch in the MLAG domain. NetApp recommends that the descriptions for each port are updated to reflect the device ports that are cabled and configured on the other switch.

### Create uplink ports for the switches

1. Create an MLAG interface to provide uplinks to both Mellanox SN2010 switches from the core network.

```
Switch-01 (config) # interface mlag port-channel 101
Switch-01 (config interface mlag port-channel) # description Uplink
CORE-SWITCH port PORT
Switch-01 (config interface mlag port-channel) # exit
```

2. Configure the MLAG members.

```
Switch-01 (config) # interface ethernet 1/18 description Uplink to CORE-
SWITCH port PORT
Switch-01 (config) # interface ethernet 1/18 speed 10000 force
Switch-01 (config) # interface mlag-port-channel 101 mtu 9216 force
Switch-01 (config) # interface ethernet 1/18 mlag-channel-group 101 mode
active
```

3. Set the switchport mode to hybrid and allow all VLANs from the core uplink switches.

```
Switch-01 (config) # interface mlag-port-channel switchport mode hybrid  
Switch-01 (config) # interface mlag-port-channel switchport hybrid  
allowed-vlan all
```

4. Verify that the MLAG interface is up.

```
Switch-01 (config) # interface mlag-port-channel 101 no shutdown  
Switch-01 (config) # exit
```

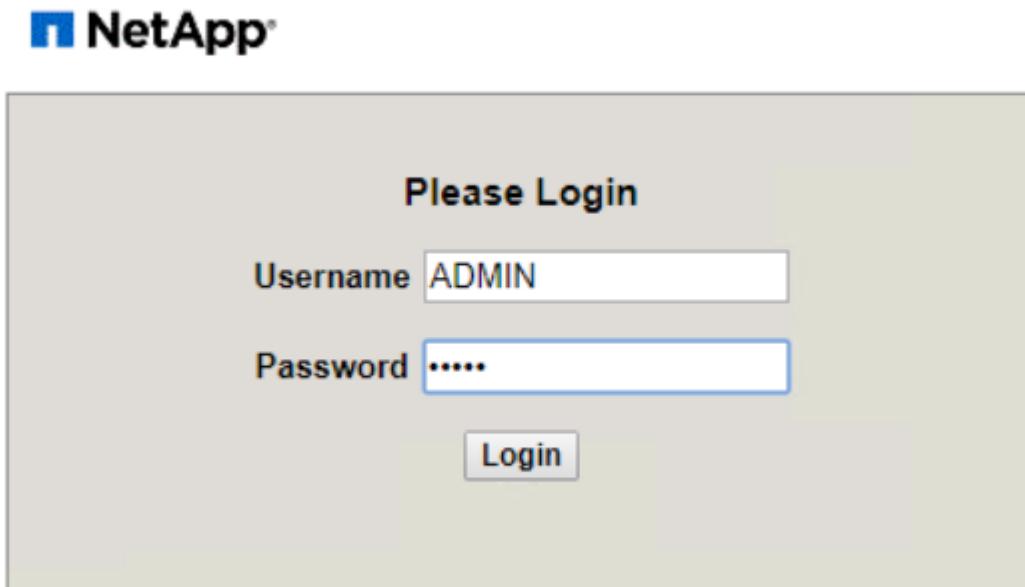
[Next: Deploy NetApp HCI with the NetApp Deployment Engine](#)

### 3. Deploy NetApp HCI with the NetApp Deployment Engine

NDE delivers a simple and streamlined deployment experience for the NetApp HCI solution. A detailed guide to using NDE 1.6 to deploy your NetApp HCI system can be found [here](#).

These steps begin after the nodes have been racked, and cabled, and the IPMI port has been configured on each node using the console. To Deploy the NetApp HCI solution using NDE, complete the following steps:

1. Access the out-of-band management console for one of the storage nodes in the cluster and log in with the default credentials ADMIN/ADMIN.



2. Click the Remote Console Preview image in the center of the screen to download a JNLP file launched by Java Web Start, which launches an interactive console to the system.
3. With the virtual console launched, a user can log in to the HCI storage node using the ADMIN/ADMIN

username and password combination.

4. The Bond1G interface must have an IP, a netmask, and a gateway set statically; its VLAN set to 3480; and DNS servers defined for the environment.

```
Bond10G
  Method          : static
  Link Speed     : 50000
  IPv4 Address   :
  IPv4 Subnet Mask  : 
    --->
  IPv4 Gateway Address : 
    --->
  MTU            : 9000
    --->
  Bond Mode      : LACP  [ActivePassive, ALB, LACP]
    --->
  LACP Rate      : Fast  [Fast, Slow]
    --->
  Status          : UpAndRunning  [Down, Up, UpAndRunning]
    --->
  Virtual Network Tag : 
    --->
  Routes          : Number of routes: 0.
    --->
```



Select an IP that is within the subnet you intend to use for in-band management but not an IP you would like to use in production. NDE reconfigures the node with a production IP after initial access.



This task must only be performed on the first storage node. Afterward, the other nodes in the infrastructure are discovered by the Automatic Private IP Address (APIPA) addresses assigned to each storage interface when left unconfigured.

5. The Bond 10G interface must have its MTU setting changed to enable jumbo frames and its bond mode changed to LACP.

```
Bond10G
      Method          : static
      Link Speed     : 50000
      IPv4 Address   :
      IPv4 Subnet Mask  :
      IPv4 Gateway Address :
      MTU            : 9000
      Bond Mode      : LACP  [ActivePassive, ALB, LACP]
      LACP Rate       : Fast   [Fast, Slow]
      Status          : UpAndRunning  [Down, Up, UpAndRunning]
      Virtual Network Tag :
      Routes          : Number of routes: 0.
```



Configure each of the four storage nodes in the NetApp HCI solution this way. The NDE process is then able to discover all the nodes in the solution and configure them. You do not need to modify the Bond10g interfaces on the two compute nodes.

6. After completion, open a web browser and visit the IP address you configured for the management port to start NetApp HCI configuration with NDE.
7. On the Welcome to NetApp HCI page, click the Get Started button.
8. Check each associated box on the Prerequisites page and click Continue.
9. The next page presents End User Licenses for NetApp HCI and VMware vSphere. If you accept the terms, click I Accept at the end of each agreement and then click Continue.
10. Click Configure a New vSphere Deployment, select vSphere 6.5U2, and enter the Fully Qualified Domain Name (FQDN) of your vCenter Server. Then click Continue.

# vSphere Configuration

You may elect to configure a new vSphere deployment or to join an existing vSphere deployment.

- Configure a new vSphere deployment
- Configure Using vSphere Version 6.7 Update 1
- Configure Using vSphere Version 6.5 Update 2
- Join and extend an existing vSphere deployment

If you have set up a DNS record for your new vCenter server, then configure your server using its fully qualified domain name and DNS server IP address:

- Configure Using a Fully Qualified Domain Name Best Practice!

## vCenter Server Fully Qualified Domain Name

anthos-vc.cie.netapp.com



**Note:** The domain name must resolve to an unused IP address.

## DNS Server IP Address

10.61.184.251



If you have not set up a DNS record for your new vCenter server, you may configure using an IP address that we define:

- Configure Using an IP Address ?

**Note:** Once defined, the IP address cannot be changed.

[Back](#)

[Continue](#)

11. NDE asks for the credentials to be used in the environment. This is used for VMware vSphere, the NetApp Element storage cluster, and the NetApp Mnode, which provides management functionality for the cluster. When you are finished, click Continue.

# Credentials

Define the user name and password that will be used for the storage cluster, vCenter, and the management node.

User Name

admin



Password

\*\*\*\*\*



**Password must contain:**

- ✓ At least 8 characters
- ✓ No more than 20 characters
- ✓ 1 uppercase letter that is not the first character
- ✓ 1 lowercase letter
- ✓ 1 of the following special characters: !@#\$
- ✓ Allowed characters: A-Z a-z 0-9 !@#\$
- ✓ 1 number that is not the last character

Re-enter Password

\*\*\*\*\*



[Back](#)

[Continue](#)

12. NDE then prompts for the network topology used to cable the NetApp HCI environment. The validated solution in this document has been deployed using the two-cable option for the compute nodes, and the four-cable option for the storage nodes. Click Continue.

# Network Topology

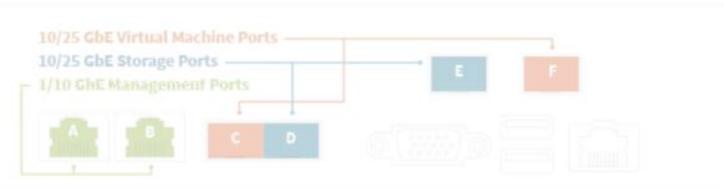
Select a compute node topology and a storage node topology appropriate for your hardware installation.

## Compute Node Topology

### 6 Cable Option

The 6 cable option provides dedicated ports for management (2 x 1/10 GbE), virtual machines (2 x 10/25 GbE) and storage (2 x 10/25 GbE).

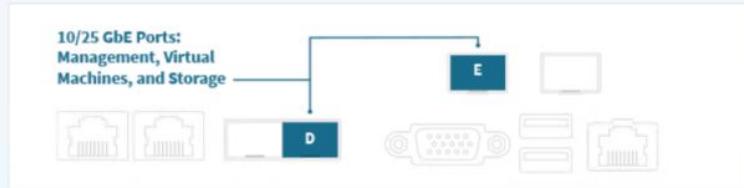
Use vSphere Distributed Switch? [?](#)



(H300E,H410C,H500E,H700E)

### 2 Cable Option

The 2 cable option provides shared management with ports for virtual machines and storage (2 x 10/25 GbE). The 2 cable option uses vSphere Distributed Switch. [?](#)

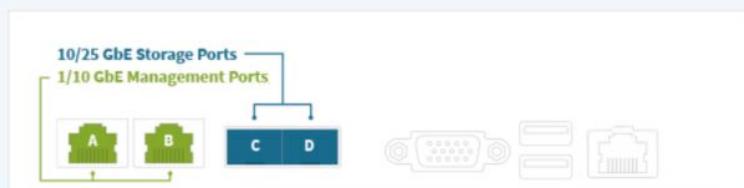


(H300E,H410C,H500E,H700E)

## Storage Node Topology

### 4 Cable Option

The 4 cable option provides dedicated ports for management (2 x 1/10 GbE) and storage (2 x 10/25 GbE).



(H300S,H410S,H500S,H700S)

[Back](#)

[Continue](#)

13. The next page presented by NDE is the inventory of the environment as discovered by the APIPA addressed on the storage network. The storage node that is currently running NDE is already selected with a green check mark. Select the corresponding boxes to add additional nodes to the NetApp HCI environment. Click Continue.

## Inventory

Verify the available nodes and select **at least 2 compute nodes and 4 storage nodes** to include in your installation.

[Refresh Inventory](#)

### Compute Nodes

	Serial Number	Chassis Serial Number / Slot	Node Type	Software Version	Physical CPU Cores	Memory	1 GbE Ports	10 GbE Ports
<input checked="" type="checkbox"/>	HM17CS002729	002170990158 / B	H410C	1.6	8	384 GB	0 of 2 detected	2 of 4 detected
<input checked="" type="checkbox"/>	HM181S002024	002170990158 / A	H410C	1.6	8	384 GB	0 of 2 detected	2 of 4 detected

1 - 2 of 2 results

◀ ▶ 1 ▶ ▷

20 ▾

2 compute nodes selected

### Storage Nodes

	Serial Number	Chassis Serial Number / Slot	Node Type	Raw Capacity	Element Version	Drive Count	1 GbE Ports	10 GbE Ports
<input checked="" type="checkbox"/>	221814003506	221814003436 / C	H500S	5.76 TB	11.3.1.5	6 of 6 detected	2 of 2 detected	2 of 2 detected
<input checked="" type="checkbox"/>	221818004613	221814003436 / D	H500S	5.76 TB	11.3.1.5	6 of 6 detected	2 of 2 detected	2 of 2 detected
<input checked="" type="checkbox"/> ?	221826005865	002170990158 / C	H500S	5.76 TB	11.3.1.5	6 of 6 detected	2 of 2 detected	2 of 2 detected
<input checked="" type="checkbox"/>	221826005866	002170990158 / D	H500S	5.76 TB	11.3.1.5	6 of 6 detected	2 of 2 detected	2 of 2 detected

1 - 4 of 4 results

◀ ▶ 1 ▶ ▷

20 ▾

4 storage nodes selected

[Back](#)

[Continue](#)



If there are any nodes missing from the inventory screen, wait a few minutes and click Refresh Inventory. If the node still fails to appear, additional investigation of environment networking might be required.

14. You must next configure the permanent network settings for the NetApp HCI deployment. The first page configures infrastructure services (DNS and NTP), vCenter networking, and Mnode networking.

# Network Settings

Provide the network settings that will be used for your installation.

Live network validation is: **On**

## Infrastructure Services

DNS Server IP Address 1	DNS Server IP Address 2 (Optional)
<input type="text" value="10.61.184.251"/>	<input type="text" value="10.61.184.252"/>
NTP Server Address 1	NTP Server Address 2 (Optional)
<input type="text" value="10.61.184.251"/>	<input type="text" value="10.61.184.252"/>

To save time, launch the easy form to enter fewer network settings.

## vCenter Networking

VLAN ID	Subnet	Default Gateway	FQDN	IP Address
<input type="text" value="3480"/>	<input type="text" value="172.21.224.0/24"/>	<input type="text" value="172.21.224.1"/>	anthos-vc.cie.netapp.com	<input type="text" value="172.21.224.10"/>

## Management Node Networking

Management Network		iSCSI Network
VLAN ID	Subnet	VLAN ID
<input type="text" value="3480"/>	<input type="text" value="172.21.224.0/24"/>	<input type="text" value="3481"/>
Subnet	Default Gateway	Subnet
<input type="text" value="172.21.224.0/24"/>	<input type="text" value="172.21.224.1"/>	<input type="text" value="172.21.225.0/24"/>
Default Gateway	Management IP Address	Storage (iSCSI) IP Address
<input type="text" value="anthos-mnode"/>	<input type="text" value="172.21.224.50"/>	<input type="text" value="172.21.225.50"/>

15. The next page allows you to configure each node in the environment. For the compute nodes, it allows you to configure the host name, management network, vMotion network, and storage network. For the storage nodes, name the storage cluster and configure the management and storage networks being used for each node. Click Continue.

## Compute Node Networking

		Management Network	vmotion Network	iSCSI A Network	iSCSI B Network
VLAN ID	VLAN ID	VLAN ID	VLAN ID	VLAN ID	VLAN ID
3480	3482	3481	3481	3481	3481
Subnet ?	Subnet ?	Subnet ?	Subnet ?	Subnet ?	Subnet ?
172.21.224.0/24	172.21.226.0/24	172.21.225.0/24	172.21.225.0/24	172.21.225.0/24	172.21.225.0/24
Default Gateway	Default Gateway (Optional)				
172.21.224.1					
Serial Number	Hostname	Management IP Address	vmotion IP Address	iSCSI A - IP Address	iSCSI B - IP Address
HM17CS002729	Anthos-ESXi-01	172.21.224.11	172.21.226.11	172.21.225.11	172.21.225.111
HM181S002024	Anthos-ESXi-02	172.21.224.12	172.21.226.12	172.21.225.12	172.21.225.112

## Storage Node Networking

### Storage Cluster Name

Anthos-Store

Note: The storage cluster name cannot be changed after deployment.

Management Network		iSCSI Network	
VLAN ID	VLAN ID	Subnet ?	Subnet ?
3480	3481	172.21.224.0/24	172.21.225.0/24
Subnet ?	Subnet ?	Default Gateway	Default Gateway (Optional)
172.21.224.0/24	172.21.225.0/24	172.21.224.1	
Management Virtual IP (MVIP) ?	Storage Virtual IP (SVIP) ?	Management IP Address	Storage (iSCSI) IP Address
172.21.224.20	172.21.225.20	172.21.224.21	172.21.225.21
		172.21.224.22	172.21.225.22
		172.21.224.23	172.21.225.23
		172.21.224.24	172.21.225.24

[Back](#)

Live network validation is:  [On](#) [?](#)

[Continue](#)

16. On the next page, review all the settings that have been defined for the environment by expanding each section, and, if necessary, click Edit to make corrections. There is also a check box on this page that enables or disables the Mnode from sending real-time health and diagnostics information to NetApp Active IQ. If all the information is correct, click Start Deployment.



If you want to enable Active IQ, verify that your management network can reach the internet. If NDE is unable to reach Active IQ, the deployment can fail.

17. A summary page appears along with a progress bar for each component of the NetApp HCI solution, as well as the overall solution. When complete, you are presented with an option to launch the vSphere client and begin working with your environment.

# Your setup is complete.

[Launch vSphere Client](#)

Configure Network	Complete	✓
Set up NetApp Cluster	Complete	✓
Set up ESXi	Complete	✓
Set up vCenter	Complete	✓
Configure Management Node	Complete	✓
Finalize Configuration	Complete	✓

Overall Progress

100%

 [Export all setup information to CSV file](#)

[Next: Configure the vCenter Server](#)

#### 4. Configure the vCenter Server

NDE deploys the solution with vCenter server and integrates the solution with the Element cluster by provisioning the Mnode VM and installing the NetApp Element Plug-in for vCenter.



Note that NDE deploys vSphere 6.7U1. You can upgrade the Virtual Appliance and individual ESXi hosts by following the instructions from VMware [here](#).

After deployment, you must make a few modifications to the environment, including the creation of additional vDS portgroups, datastores, and resource groups for the deployment of the Anthos on VMware solution.

Complete the following steps to configure your vCenter Server:

1. Log into the VMware vCenter server using the `Administrator@vsphere.local` account and the password chosen for the admin user during NDE configuration.



2. Right-click **NetApp-HCI-Cluster-01** created by NDE and select the option to create a new resource pool. Name this pool **Infrastructure-Resource-Pool** and accept the defaults by clicking OK. This resource pool is used in a later configuration step.

# New Resource Pool

NetApp-HCI-Cluster-01



Name	Infrastructure Resource		
<b>CPU</b>			
Shares	Normal	4000	
Reservation	0	MHz	▼
Max reservation: 54,128 MHz			
Reservation Type	<input checked="" type="checkbox"/> Expandable		
Limit	Unlimited	MHz	▼
Max limit: 58,128 MHz			
<b>Memory</b>			
Shares	Normal	163840	
Reservation	0	MB	▼
Max reservation: 751,064 MB			
Reservation Type	<input checked="" type="checkbox"/> Expandable		
Limit	Unlimited	MB	▼
Max limit: 756,820 MB			

CANCEL

OK



The reservations in this resource pool can be modified based on the resources available in the environment. NetApp HCI is deployed as an all-in-one solution. Therefore, NetApp recommends reserving the resources necessary to provide availability for the infrastructure services by placing them into this resource pool and adjusting the resources appropriately. Infrastructure services include vCenter Server, NetApp Mnnode, and F5 Big-IP Load Balancer.

3. Repeat this step to create another resource pool for VMs deployed by Anthos. Name this pool Anthos-Resource-Pool, and click the OK button to accept the default values. Adjust the resource availability based on the specific environment in which you are deploying the solution. This resource pool is used in a later deployment step.
4. To configure Element volumes to be used as vSphere datastores, click the dropdown menu and select NetApp Element Management from the list.
5. A Getting Started screen appears with details about your Element cluster.



6. Click Management, and the vSphere client presents a list of datastores. Click Create Datastore to create one datastore to host VMs and another to host ISOs for future guest installs.
7. Next click the Network menu item in the left panel. This displays a screen with information about the vDS deployed by NDE.
8. Several virtual port groups are defined by the initial configuration. NetApp recommends leaving these alone to support the infrastructure, and additional port groups should be created for user-deployed virtual guests. Right-click the NetApp HCI VDS 01 vDS in the left panel, and then select Distributed Port Group followed by the New Distributed Port Group option from the expanded menu.
9. Create a new distributed port group called **Management\_Network**. Then click Next.
10. On the next screen, select the VLAN type as VLAN, and set the VLAN ID to 3480 for management purposes. Click Next, and, after reviewing the options on the summary page, click Next again to complete the creation of the distributed port group.
11. Repeat these steps to create distributed port groups for the **VM\_Network** (VLAN 1172) as well as any other networks that might be used in the NetApp HCI environment.



Additional networks can be defined to segment any additional deployed VMs. Examples of this use could be for a dedicated HA network for additional F5 Big-IP appliances if provisioned. Such configurations are in addition to the environment deployed in this validated solution and are considered out of scope for this NVA document.

[Next: Deploy and Configure the F5 Big-IP Virtual Edition Appliance](#)

## 5. Deploy and Configure the F5 Big-IP Virtual Edition Appliance

Anthos enables native integration with F5 Big-IP load balancers to expose services from each pod to the world.

This solution makes use of the virtual appliance deployed in VMware vSphere as deployed by NDE. Networking for the F5 Big-IP virtual appliance can be configured in a two-armed or three-armed configuration based on your network environment. The deployment in this document is based on the two-armed configuration. Additional details for configuring the virtual appliance for use with Anthos can be found [here](#).

To deploy the F5 Big-IP Virtual Edition appliance, complete the following steps:

1. Download the virtual application Open Virtual Appliance (OVA) file from F5 [here](#).



To download the appliance, a user must register with F5. They provide a 30-day demo license for the Big-IP Virtual Edition Load Balancer. NetApp recommends a permanent 10Gbps license for the production deployment of an appliance.

2. Right-click the infrastructure resource pool and select Deploy OVF Template. A wizard launches that allows you to select the OVA file that you just downloaded in Step 1. Click Next.

## Deploy OVF Template

### 1 Select an OVF template

- 2 Select a name and folder
- 3 Select a compute resource
- 4 Review details
- 5 Select storage
- 6 Ready to complete

### Select an OVF template

Select an OVF template from remote URL or local file system

Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

URL

http | https://remoteserver-address/filetodeploy.ovf | .ova

Local file

BIGIP-15.0.1-0....ALL-vmware.ova

3. Click Next to continue through each step and accept the default values for each screen presented until you reach the storage selection screen. Select the VM\_Datastore that was created earlier, and then click Next.
4. The next screen presented by the wizard allows you to customize the virtual networks for use in the environment. Select VM\_Network for the External field and select Management\_Network for the Management field. Internal and HA are used for advanced configurations for the F5 Big-IP appliance and

are not configured. These parameters can be left alone, or they can be configured to connect to non-infrastructure, distributed port groups. Click Next.

5. Review the summary screen for the appliance, and, if all the information is correct, click Finish to start the deployment.
6. After the virtual appliance is deployed, right-click it and power it up. It should receive a DHCP address on the management network. The appliance is Linux-based, and it has VMware Tools deployed, so that you can view the DHCP address it receives in the vSphere client.
7. Open a web browser and connect to the appliance at the IP address from the previous step. The default login is admin/admin, and, after the first login, the appliance immediately prompts you to change the admin password. It then returns you to a screen where you must log in with the new credentials.

The screenshot shows the F5 BIG-IP Configuration Utility login interface. On the left, there is a sidebar with the following fields:

- Hostname**: bigip1
- IP Address**: 172.21.224.20
- Username**: admin
- Password**: (redacted)

On the right, the main panel displays the following text:

Welcome to the BIG-IP Configuration Utility.  
Log in with your username and password using the fields on the left.

At the bottom of the main panel, there is a copyright notice:

(c) Copyright 1996-2019, F5 Networks, Inc., Seattle, Washington. All rights reserved.  
[F5 Networks, Inc. Legal Notices](#)

8. The first screen prompts the you to complete the Setup Utility. Begin the utility by clicking Next.
9. The next screen prompts you for activation of the appliance license. Click Activate to begin. When prompted on the next page, paste either the 30-day evaluation license key you received when you registered for the download or the permanent license you acquired when you purchased the appliance. Click Next.



For the device to perform activation, the network defined on the management interface must be able to reach the internet.

10. On the next screen, the End User License Agreement (EULA) is presented. If the terms in the license are acceptable, click Accept.
11. The next screen counts the elapsed time as it verifies the configuration changes that have been made so far. Click Continue to resume with the initial configuration.
12. The Configuration Change window closes, and the Setup Utility displays the Resource Provisioning menu. This window lists the features that are currently licensed and the current resource allocations for the virtual appliance and each running service.
13. Clicking the Platform menu option on the left enables additional modification of the platform. Modifications include setting the management IP address configured with DHCP, setting the host name and the time zone the appliance is installed in, and securing the appliance from SSH accessibility.
14. Next click the Network menu, which enables you to configure standard networking features. Click Next to begin the Standard Network Configuration wizard.
15. The first page of the wizard configures redundancy; leave the defaults and click Next. The next page enables you to configure an internal interface on the load balancer. Interface 1.1 maps to the vmnic labeled Internal in the OVF deployment wizard.

#### [Big-IP Configuration]



The fields in this page for Self IP Address, Netmask, and Floating IP address can be filled with a non-routable IP address for use as a placeholder. They can also be filled with an internal network that has been configured as a distributed port group for virtual guests if you are deploying the three-armed configuration. They must be completed to continue with the wizard.

16. The next page enables you to configure an external network that is used to map services to the pods deployed in Kubernetes. Select a static IP from the VM\_Network range, the appropriate subnet mask, and a floating IP from that same range. Interface 1.2 maps to the vmnic labeled External in the OVF deployment wizard.

#### [Big-IP Configuration]

17. On the next page, you can configure an internal-HA network if you are deploying multiple virtual appliances in the environment. To proceed, you must fill the Self-IP Address and the Netmask fields, and you must select interface 1.3 as the VLAN Interface, which maps to the HA network defined by the OVF template wizard.

18. The next page enables you to configure the NTP servers. Then click Next to continue to the DNS setup.

The DNS servers and domain search list should already be populated by the DHCP server. Click Next to accept the defaults and continue.

19. For the remainder of the wizard, click Next to continue through the advanced peering setup, the configuration of which is beyond the scope of this document. Then click Finish to exit the wizard.
  20. Create individual partitions for the Anthos admin cluster and each user cluster deployed in the environment. Click System in the menu on the left, navigate to Users, and click Partition List.
- 
21. The displayed screen only shows the current common partition. Click Create on the right to create the first additional partition and name it **Anthos-Admin**. Then click Repeat, name the partition **Anthos-Cluster1**, and click the Repeat button again to name the next partition **Anthos-Cluster2**. Finally click Finished to complete the wizard. The Partition list screen returns with all the partitions now listed.

## Next: Complete Anthos Prerequisites

### Complete Anthos prerequisites

Now that the physical environment is set up, you can begin Anthos deployment. This starts with several prerequisites that you must meet to deploy the solution and access it afterward. Each of these steps are discussed in depth in the Anthos [GKE On-Prem Guide](#).

To prepare your environment for the deployment of Anthos on VMware, complete the following steps:

1. Create a Google Cloud project following the instructions available [here](#).



Your organization might already have a project in place intended for this purpose. Check with your cloud administration team to see if a project exists and is already configured for access to Anthos on VMware. All projects intended for use with Anthos must be whitelisted by Google. This includes the primary user account, additional team members, and the access service account created in a later step.

2. Create a deployment workstation from which to manage the installation of Anthos on VMware. The deployment workstation can be Linux, MacOS, or Windows. For the purposes of this validated deployment, Red Hat Enterprise Linux 7 was used.



This workstation can be hosted either internal or external to the NetApp HCI deployment. The only requirement is that it must be able to successfully communicate with the deployed VMware vCenter Server and the internet to function correctly.

3. Install [Google Cloud SDK](#) for interactions with Google Cloud. It can be downloaded as an archive of binaries for manual install or installed by either the apt-get (Ubuntu/Debian) or yum (RHEL) package managers.

```
[user@rhel7 ~]$ sudo yum install google-cloud-sdk
Failed to set locale, defaulting to C
Loaded plugins: langpacks, product-id, search-disabled-repos,
subscription-manager
Resolving Dependencies
--> Running transaction check
--> Package google-cloud-sdk.noarch 0:270.0.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
=====
=====
Package           Arch      Version       Repository
Size
=====
=====
=====
Installing:
google-cloud-sdk      noarch    270.0.0-1   google-cloud-
sdk                36 M

Transaction Summary
=====
=====
Install 1 Package

Total download size: 36 M
Installed size: 174 M
Is this ok [y/d/N]: y
Downloading packages:
6d81c821884ae40244c746f6044fc1bcd801143a0d9c8da06767036b8d090a24-google-
cloud-sdk-270.0.0-1.noar | 36 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : google-cloud-sdk-270.0.0-1.noarch
1/1
  Verifying  : google-cloud-sdk-270.0.0-1.noarch
1/1

Installed:
  google-cloud-sdk.noarch 0:270.0.0-1

Complete!
```



The gcloud binary must be at least version 265.0.0. You can update a manual install with a gcloud components update. However, if SDK was installed by a package manager, future updates must also be performed using that same package manager.

- With the workstation configured, log in to Google Cloud with your credentials. To do so, enter the login command from the deployment workstation and retrieve a link that can be copied and pasted into a browser to allow interactive sign-in to Google services. After you have logged in, the web page presents a code that you can copy and paste back into the deployment workstation when prompted.

```
[user@rhel7 ~]$ gcloud auth login  
Go to the following link in your browser:
```

```
https://accounts.google.com/o/oauth2/auth?code_challenge=-  
7oPNSySHr_Sd2Zz4K83koIeGTLVcdbjc8omr6zCbAI&prompt=select_account&code_ch  
allenge_method=S256&access_type=offline&redirect_uri=urn%3Aietf%3Awg%3Ao  
auth%3A2.0%3Aoob&response_type=code&client_id=32655940559.apps.googleuse  
rcontent.com&scope=https%3A%3F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.em  
ail+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-  
platform+https%3A%6F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https  
%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis  
.com%2Fauth%2Faccounts.reauth
```

```
Enter verification code: 6/swGAh52VVgB-  
TRS5LVrSvP79ZdDlb9V6ObyUGqoY67a3zp9NPciIKsM  
You are now logged in as [user@netapp.com].  
Your current project is [anthos-dev]. You can change this setting by  
running:  
$ gcloud config set project PROJECT_ID
```

- Enable several APIs so that your environment can communicate with Google Cloud. The pods deployed in your clusters must be able to access <https://www.googleapis.com> and <https://gkeconnect.googleapis.com> to function as expected. Therefore, the VM\_Network that the worker nodes are attached to must have internet access. To enable the necessary APIs, run the following command from the deployment workstation:

```
[user@rhel7 ~]$ gcloud services enable --project anthos-dev \  
cloudresourcemanager.googleapis.com \  
container.googleapis.com \  
gkeconnect.googleapis.com \  
gkehub.googleapis.com \  
serviceusage.googleapis.com \  
stackdriver.googleapis.com \  
monitoring.googleapis.com \  
logging.googleapis.com
```

6. Create a working directory called anthos-install, and change into that directory.

```
[user@rhel7 ~]$ mkdir anthos-install && cd anthos-install  
[user@rhel7 anthos-install]$
```

7. Before you can install Anthos on VMware, you must create four service accounts, each with a specific purpose in interacting with Google Cloud. The following table lists the accounts and their purposes.

Account Name	Purpose
component-access-sa	Used to download the Anthos binaries from Cloud Storage.
connect-register-sa	Used to register Anthos clusters to the Google Cloud console.
connect-agent-sa	Used to maintain the connection between user clusters and the Google Cloud.
logging-monitoring-sa	Used to write logging and monitoring data to Stackdriver.



Each account is assigned an email address that references your approved Google Cloud project name. The following examples all list the project Anthos-Dev, which was used during the NetApp validation. Make sure to substitute your appropriate project name in syntax examples where necessary.

```

[user@rhel7 anthos-install]$ gcloud iam service-accounts create
component-access-sa \
    --display-name "Component Access Service Account" \
    --project anthos-dev
[user@rhel7 anthos-install]$ gcloud iam service-accounts keys create
component-access-key.json \
    --iam-account component-access-sa@anthos-dev.iam.gserviceaccount.com

[user@rhel7 anthos-install]$ gcloud iam service-accounts create connect-
register-sa \
    --project anthos-dev
[user@rhel7 anthos-install]$ gcloud iam service-accounts keys create
connect-register-key.json \
    --iam-account connect-register-sa@anthos-dev.iam.gserviceaccount.com

[user@rhel7 anthos-install]$ gcloud iam service-accounts create connect-
agent-sa \
    --project anthos-dev
[user@rhel7 anthos-install]$ gcloud iam service-accounts keys create
connect-agent-key.json \
    --iam-account connect-agent-sa@anthos-dev.iam.gserviceaccount.com

[user@rhel7 anthos-install]$ gcloud iam service-accounts create logging-
monitoring-sa \
    --project anthos-dev
[user@rhel7 anthos-install]$ gcloud iam service-accounts keys create
logging-monitoring-key.json \
    --iam-account logging-monitoring-sa@anthos-
dev.iam.gserviceaccount.com

```

8. The final step needed to prepare your environment to deploy Anthos is to limit certain privileges to your service accounts. You need the associated email address for each service account listed in Step 7.
  - a. Using the component-access-sa account, assign the roles for `serviceusage.serviceUsageViewer`, `iam.serviceAccountCreator`, and `iam.roleViewer`.

```
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding  
anthos-dev\  
    --member "serviceAccount:component-access-sa@anthos-  
dev.iam.gserviceaccount.com" \  
    --role "roles/serviceusage.serviceUsageViewer"  
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding  
anthos-dev\  
    --member "serviceAccount:component-access-sa@anthos-  
dev.iam.gserviceaccount.com" \  
    --role "roles/iam.serviceAccountCreator"  
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding  
anthos-dev\  
    --member "serviceAccount:component-access-sa@anthos-  
dev.iam.gserviceaccount.com" \  
    --role "roles/iam.roleViewer"
```

- b. Using the connect-register-sa service account, assign the role for [gkehub.admin](#).

```
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding  
anthos-dev \  
    --member "serviceAccount:connect-register-sa@anthos-  
dev.iam.gserviceaccount.com" \  
    --role "roles/gkehub.admin"
```

- c. Using the connect-agent-sa account, assign the role for [gkehub.connect](#).

```
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding  
anthos-dev \  
    --member "serviceAccount:connect-agent-sa@anthos-  
dev.iam.gserviceaccount.com" \  
    --role "roles/gkehub.connect"
```

- d. With the logging-monitoring-sa service account, assign the roles for [stackdriver.resourceMetadata.writer](#), [logging.logWriter](#), [monitoring.metricWriter](#), and [monitoring.dashboardEditor](#).

```
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding
anthos-dev \
    --member "serviceAccount:logging-monitoring-sa@anthos-
dev.iam.gserviceaccount.com" \
    --role "roles/stackdriver.resourceMetadata.writer"
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding
anthos-dev\
    --member "serviceAccount:logging-monitoring-sa@anthos-
dev.iam.gserviceaccount.com" \
    --role "roles/logging.logWriter"
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding
anthos-dev\
    --member "serviceAccount:logging-monitoring-sa@anthos-
dev.iam.gserviceaccount.com" \
    --role "roles/monitoring.metricWriter"
[user@rhel7 anthos-install]$ gcloud projects add-iam-policy-binding
anthos-dev\
    --member "serviceAccount:logging-monitoring-sa@anthos-
dev.iam.gserviceaccount.com" \
    --role "roles/monitoring.dashboardEditor"
```

9. Download the vCenter certificate for the VMWare CA; this is used later to authenticate to the vCenter during installation.

```
[user@rhel7 anthos-install]$ true | openssl s_client -connect anthos-
vc.cie.netapp.com:443 -showcerts 2>/dev/null | sed -ne '/-BEGIN/,/-END/p' > vcenter.pem
```

[Next: Deploy the Anthos admin workstation](#)

## 7. Deploy the Anthos admin workstation

The admin workstation is a vSphere VM deployed within your NetApp HCI environment that is preinstalled with all the tools necessary to administer the Anthos on VMware solution. Follow the instructions in this section to deploy the Anthos admin workstation.

To deploy the Anthos admin workstation, complete the following steps:

1. Download the gkeadm binary into your working directory

```
[user@rhel7 anthos-install]$ gsutil cp gs://gke-on-prem-
release/gkeadm/1.6.1-gke.1/linux/gkeadm ./
[user@rhel7 anthos-install]$ chmod +x gkeadm
```

2. Use the gkeadm tool to create an admin workstation configuration file.

```
[user@rhel7 anthos-install]$ ./gkeadm create config
```

3. Two files are created: `credential.yaml` and `admin-ws-config.yaml`. Fill out each of these files.

- a. `credential.yaml` contains your username and passwords for your VMware vCenter server.

```
kind: CredentialFile
items:
- name: vCenter
  username: "administrator@vsphere.local"
  password: "vSphereAdminPassword"
```

- b. `admin-ws-config.yaml` contains other information about your vSphere environment as well as the physical and networking options for the admin-workstation VM.

```
gcp:
  # Path of the whitelisted service account's JSON key file
  whitelistedServiceAccountKeyPath: "/home/anthos-install/service-
keys/access-key.json"
  # Specify which vCenter resources to use
  vCenter:
    # The credentials and address GKE On-Prem should use to connect to
    vCenter
    credentials:
      address: "anthos-vc.cie.netapp.com"
      datacenter: "NetApp-HCI-Datacenter-01"
      datastore: "VM_Datastore"
      cluster: "NetApp-HCI-Cluster-01"
      network: "VM_Network"
      resourcePool: "Anthos-Resource-Pool"
    # Provide the path to vCenter CA certificate pub key for SSL
    verification
    caCertPath: "/home/anthos-install/vcenter.pem"
  # The URL of the proxy for the jump host
  proxyUrl: ""
  adminWorkstation:
    name: gke-admin-ws-200915-151421
    cpus: 4
    memoryMB: 8192
  #The boot disk size of the admin workstation in GB. It is recommended
  #to use a disk with at least 50 GB to host images decompressed from
  #the bundle.
  diskGB: 50
  # Name for the persistent disk to be mounted to the home directory
```

```
(ending in  
.vmdk).  
# Any directory in the supplied path must be created before  
deployment.  
  dataDiskName: gke-on-prem-admin-workstation-data-disk/gke-admin-ws-  
200915-151421-data-disk.vmdk  
# The size of the data disk in MB.  
  dataDiskMB: 512  
network:  
# The IP allocation mode: 'dhcp' or 'static'  
  ipAllocationMode: "dhcp"  
# # The host config in static IP mode. Do not include if using DHCP  
# hostConfig:  
  #   # The IPv4 static IP address for the admin workstation  
  #   ip: ""  
  #   # The IP address of the default gateway of the subnet in  
which the admin workstation  
  #   # is to be created  
  #   gateway: ""  
  #   # The subnet mask of the network where you want to create  
your admin workstation  
  #   netmask: ""  
  #   # The list of DNS nameservers to be used by the admin  
workstation  
  #   dns:  
  #   - ""  
# The URL of the proxy for the admin workstation  
proxyUrl: ""  
ntpServer: ntp.ubuntu.com
```

#### 4. Create the admin workstation.

```
[user@rhel7 anthos-install]$ ./gkeadm create admin-workstation
The output will be verbose as the workstation is created. In the end you
will be prompted with the IP address to login to the workstation if you
chose DHCP.

...
Getting ... service account...
...
*****
Admin workstation is ready to use.

Admin workstation information saved to /usr/local/google/home/me/my-
admin-workstation
This file is required for future upgrades
SSH into the admin workstation with the following command:
ssh -i /home/user/.ssh/gke-admin-workstation ubuntu@10.63.172.10
*****
```

Next: Deploy the admin and the first user cluster

## 8. Deploy the admin cluster

All Kubernetes clusters deployed as a part of the Anthos solution are deployed from the Anthos admin workstation that you just created. A user logs into the admin workstation using SSH, the public key created in a previous step, and the IP address provided at the end of the VM deployment. An admin cluster controls all actions in an Anthos environment. The admin cluster must be deployed first, and then individual user clusters can be deployed for specific workload needs.



There are specific procedures for deploying clusters that use static IP addresses [here](#), and procedures for environments with DHCP can be found [here](#). In this guide, we use the second set of instructions for ease of deployment.

To deploy the admin cluster, complete the following steps:

1. Log into your admin-workstation using the SSH command prompted at the end of the deployment. After successful authentication, you can list the files in the home directory, which are used to create the admin cluster and additional clusters later on. The directory also includes the copied vCenter cert and the access key for Anthos that was created in earlier steps.

```
[user@rhel7 anthos-install]$ ssh -i ~/.ssh/gke-admin-workstation  
ubuntu@10.63.172.10
```

```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1001-gkeop x86_64)
```

- \* Documentation: <https://help.ubuntu.com>
- \* Management: <https://landscape.canonical.com>
- \* Support: <https://ubuntu.com/advantage>

```
Last login: Fri Jan 29 15:46:35 2021 from 10.249.129.216
```

```
ubuntu@gke-admin-200915-151421:~$ ls  
admin-cluster.yaml  
user-cluster.yaml  
vcenter.pem  
component-access-key.json
```

2. Use scp to copy the remaining keys for your Anthos account over from the workstation you deployed the admin-workstation from.

```
ubuntu@gke-admin-200915-151421:~$ scp user@rhel7:~/anthos-  
install/connect-register-key.json ./  
ubuntu@gke-admin-200915-151421:~$ scp user@rhel7:~/anthos-  
install/connect-agent-key.json ./  
ubuntu@gke-admin-200915-151421:~$ scp user@rhel7:~/anthos-  
install/logging-monitoring-key.json ./
```

3. Edit the admin-cluster.yaml file so that it is specific to the deployed environment. The file is very large, so we will address it by sections.

- a. Most of the information is already filled in by default based on the configuration used to deploy the admin-workstation by gkeadm. This first section confirms the information for the version of Anthos being deployed and the vCenter instance it is deployed on. It also allows you to define a local data disk (VMDK) for Kubernetes object data.

```

apiVersion: v1
kind: AdminCluster
# (Required) Absolute path to a GKE bundle on disk
bundlePath: /var/lib/gke/bundles/gke-onprem-vsphere-1.6.0-gke.7-
full.tgz
# (Required) vCenter configuration
vCenter:
  address: anthos-vc.cie.netapp.com
  datacenter: NetApp-HCI-Datacenter-01
  cluster: NetApp-HCI-Cluster-01
  resourcePool: Anthos-Resource-Pool
  datastore: VM_Datastore
  # Provide the path to vCenter CA certificate pub key for SSL
  verification
  caCertPath: "/home/ubuntu/vcenter.pem"
  # The credentials to connect to vCenter
  credentials:
    username: administrator@vsphere.local
    password: "vSphereAdminPassword"
  # Provide the name for the persistent disk to be used by the
  deployment (ending
  # in .vmdk). Any directory in the supplied path must be created
  before deployment
  dataDisk: "admin-cluster-disk.vmdk"

```

- b. Fill out the networking section next, and select whether you are using static or DHCP mode. If you are using static addresses, you must create an IP-block file based on the instructions linked to above, and add it to the config file.



If static IPs are used in a deployment, the items under the host configuration are global. This includes static IPs for clusters or those used for SeeSaw load balancers, which are configured later.

```

# (Required) Network configuration
network:
# (Required) Hostconfig for static addresseses on Seesaw LB's
hostConfig:
  dnsServers:
    - "10.61.184.251"
    - "10.61.184.252"
  ntpServers:
    - "0.pool.ntp.org"
    - "1.pool.ntp.org"
    - "2.pool.ntp.org"
  searchDomainsForDNS:
    - "cie.netapp.com"
ipMode:
  # (Required) Define what IP mode to use ("dhcp" or "static")
  type: dhcp
  # # (Required when using "static" mode) The absolute or relative
  path to the yaml file
  # # to use for static IP allocation
  # ipBlockFilePath: ""
  # (Required) The Kubernetes service CIDR range for the cluster.
  Must not overlap
  # with the pod CIDR range
  serviceCIDR: 10.96.232.0/24
  # (Required) The Kubernetes pod CIDR range for the cluster. Must
  not overlap with
  # the service CIDR range
  podCIDR: 192.168.0.0/16
vCenter:
  # vSphere network name
  networkName: VM_Network

```

- c. Fill out the load balancer section next. This can vary depending on the type of load balancer being deployed.

**Seesaw example:**

```

loadBalancer:
  # (Required) The VIPs to use for load balancing
  vips:
    # Used to connect to the Kubernetes API
    controlPlaneVIP: "10.63.172.155"
    # # (Optional) Used for admin cluster addons (needed for multi
    cluster features). Must
    # # be the same across clusters

```

```

    # # addonsVIP: "10.63.172.153"
    # (Required) Which load balancer to use "F5BigIP" "Seesaw" or
    "ManualLB". Uncomment
        # the corresponding field below to provide the detailed spec
        kind: Seesaw
        # # (Required when using "ManualLB" kind) Specify pre-defined
        nodeports
        # manualLB:
            # # NodePort for ingress service's http (only needed for user
            cluster)
            #     ingressHTTPNodePort: 0
            #     # NodePort for ingress service's https (only needed for user
            cluster)
            #     ingressHTTPSPNodePort: 0
            #     # NodePort for control plane service
            #     controlPlaneNodePort: 30968
            #     # NodePort for addon service (only needed for admin cluster)
            #     addonsNodePort: 31405
            # # (Required when using "F5BigIP" kind) Specify the already-
            existing partition and
            # # credentials
            # f5BigIP:
            #     address:
            #         credentials:
            #             username:
            #             password:
            #             partition:
            #                 # # (Optional) Specify a pool name if using SNAT
            #                 snatPoolName: ""
            # (Required when using "Seesaw" kind) Specify the Seesaw configs
            seesaw:
                # (Required) The absolute or relative path to the yaml file to use
                for IP allocation
                # for LB VMs. Must contain one or two IPs.
                ipBlockFilePath: "admin-seesaw-block.yaml"
                # (Required) The Virtual Router IDentifier of VRRP for the Seesaw
                group. Must
                    # be between 1-255 and unique in a VLAN.
                    vrid: 100
                    # (Required) The IP announced by the master of Seesaw group
                    masterIP: "10.63.172.151"
                    # (Required) The number CPUs per machine
                    cpus: 1
                    # (Required) Memory size in MB per machine
                    memoryMB: 2048
                    # (Optional) Network that the LB interface of Seesaw runs in

```

```

(default: cluster
  #   network)
  vCenter:
    #   vSphere network name
    networkName: VM_Network
    #   (Optional) Run two LB VMs to achieve high availability
(default: false)
  enableHA: false

```

- d. For a SeeSaw load balancer, you must create an additional external file to supply the static IP information for the load balancer. Create the file [admin-seesaw-block.yaml](#), which was referenced in this configuration section.

```

blocks:
  - netmask: "255.255.255.0"
    gateway: "10.63.172.1"
    ips:
      - ip: "10.63.172.152"
        hostname: "admin-seesaw-vm"

```

#### F5 BigIP Example:

```

# (Required) Load balancer configuration
loadBalancer:
  # (Required) The VIPs to use for load balancing
  vips:
    # Used to connect to the Kubernetes API
    controlPlaneVIP: "10.63.172.155"
    # # (Optional) Used for admin cluster addons (needed for multi
    # cluster features). Must
    # # be the same across clusters
    # # addonsVIP: "10.63.172.153"
    # (Required) Which load balancer to use "F5BigIP" "Seesaw" or
    # "ManualLB". Uncomment
    # the corresponding field below to provide the detailed spec
    kind: F5BigIP
    # # (Required when using "ManualLB" kind) Specify pre-defined
    nodeports
    # manualLB:
    #   # NodePort for ingress service's http (only needed for user
    # cluster)
    #     ingressHTTPNodePort: 0
    #   # NodePort for ingress service's https (only needed for user
    # cluster)
    #     ingressHTTPSNodePort: 0

```

```

#   # NodePort for control plane service
#   controlPlaneNodePort: 30968
#   # NodePort for addon service (only needed for admin cluster)
#   addonsNodePort: 31405
# # (Required when using "F5BigIP" kind) Specify the already-existing partition and
# # credentials
f5BigIP:
  address: "172.21.224.21"
  credentials:
    username: "admin"
    password: "admin-password"
    partition: "Admin-Cluster"
#   # (Optional) Specify a pool name if using SNAT
#   # snatPoolName: ""
# (Required when using "Seesaw" kind) Specify the Seesaw configs
# seesaw:
  # (Required) The absolute or relative path to the yaml file to use for IP allocation
  # for LB VMs. Must contain one or two IPs.
  # ipBlockFilePath: ""
  # (Required) The Virtual Router IDentifier of VRRP for the Seesaw group. Must
    # be between 1-255 and unique in a VLAN.
  # vrid: 0
  # (Required) The IP announced by the master of Seesaw group
  # masterIP: ""
  # (Required) The number CPUs per machine
  # cpus: 4
  # (Required) Memory size in MB per machine
  # memoryMB: 8192
  # (Optional) Network that the LB interface of Seesaw runs in
(default: cluster
  # network)
  # vCenter:
    # vSphere network name
    #     networkName: VM_Network
  # (Optional) Run two LB VMs to achieve high availability
(default: false)
  # enableHA: false

```

- e. The last section of the admin config file contains additional options that can be tuned to fit the specific deployment environment. These include enabling anti-affinity groups if Anthos is being deployed on less than three ESXi servers. You can also configure proxies, private docker registries, and the connections to Stackdriver and Google Cloud for auditing.

```
antiAffinityGroups:
  # Set to false to disable DRS rule creation
  enabled: false
  # (Optional) Specify the proxy configuration
  proxy:
    # The URL of the proxy
    url: ""
    # The domains and IP addresses excluded from proxying
    noProxy: ""
  # # (Optional) Use a private Docker registry to host GKE images
  # privateRegistry:
    #   # Do not include the scheme with your registry address
    #   address: ""
    #   credentials:
    #     username: ""
    #     password: ""
    #   # The absolute or relative path to the CA certificate for this
    #   registry
    #   caCertPath: ""
    # (Required): The absolute or relative path to the GCP service
    # account key for pulling
    # GKE images
    gcrKeyPath: "/home/ubuntu/component-access-key.json"
    # (Optional) Specify which GCP project to connect your logs and
    # metrics to
  stackdriver:
    projectID: "anthos-dev"
    # A GCP region where you would like to store logs and metrics for
    # this cluster.
    clusterLocation: "us-east1"
    enableVPC: false
    # The absolute or relative path to the key file for a GCP service
    # account used to
    # send logs and metrics from the cluster
    serviceAccountKeyPath: "/home/ubuntu/logging-monitoring-key.json"
  # # (Optional) Configure kubernetes apiserver audit logging
  # cloudAuditLogging:
    #   projectid: ""
    #   # A GCP region where you would like to store audit logs for this
    #   cluster.
    #   clusterlocation: ""
    #   # The absolute or relative path to the key file for a GCP service
    #   account used to
    #   # send audit logs from the cluster
    #   serviceaccountkeypath: ""
```



The deployment detailed in this document is a minimum configuration for validation that requires the disabling of anti-affinity rules. NetApp recommends leaving this option set to true in production deployments.



By default, Anthos on VMware uses a pre-existing, Google-owned container image registry that requires no additional setup. If you choose to use a private Docker registry for deployment, then you must configure that registry separately based on instructions found [here](#). This step is beyond the scope of this deployment guide.

- When edits to the admin-cluster.yaml file are complete, be sure to check for proper syntax and spacing.

```
ubuntu@gke-admin-200915-151421:~$ gkectl check-config --config admin-cluster.yaml
```

- After the configuration check has passed and any identified issues have been remedied, you can then stage the deployment of the cluster. Since we have already checked the validation of the config file, we can skip those steps by passing the `--skip-validation-all` flag.

```
ubuntu@gke-admin-200915-151421:~$ gkectl prepare --config admin-cluster.yaml --skip-validation-all
```

- If you are using a SeeSaw load balancer, you must create one before deploying the cluster itself (otherwise skip this step).

```
ubuntu@gke-admin-200915-151421:~$ gkectl create loadbalancer --config admin-cluster.yaml
```

- You can now stand up the admin cluster. This is done with the `gkectl create admin` command, which can use the `--skip-validation-all` flag to speed up deployment.

```
ubuntu@gke-admin-200915-151421:~$ gkectl create admin --config admin-cluster.yaml --skip-validation-all
```

- When the cluster is deployed, it creates the kubeconfig file in the local directory. This file can be used to check the status of the cluster using kubectl or run diagnostics with gkectl.

```
ubuntu@gke-admin-ws-200915-151421:~ $ kubectl get nodes --kubeconfig kubeconfig
NAME                      STATUS   ROLES      AGE
VERSION
gke-admin-master-gkvmp    Ready    master     5m
v1.18.6-gke.6600
gke-admin-node-84b77ff5c7-6zg59  Ready    <none>    5m
v1.18.6-gke.6600
gke-admin-node-84b77ff5c7-8jdmz    Ready    <none>    5m
v1.18.6-gke.6600
ubuntu@gke-admin-ws-200915-151421:~$ gkectl diagnose cluster --kubeconfig kubeconfig
Diagnosing admin cluster "gke-admin-gkvmp"...- Validation Category:
Admin Cluster VCenter
Checking Credentials...SUCCESS
Checking Version...SUCCESS
Checking Datacenter...SUCCESS
Checking Datastore...SUCCESS
Checking Resource pool...SUCCESS
Checking Folder...SUCCESS
Checking Network...SUCCESS- Validation Category: Admin Cluster
Checking cluster object...SUCCESS
Checking machine deployment...SUCCESS
Checking machineset...SUCCESS
Checking machine objects...SUCCESS
Checking kube-system pods...SUCCESS
Checking storage...SUCCESS
Checking resource...System pods on UserMaster cpu resource request report: total 1754m nodeCount 2 min 877m max 877m avg 877m tracked amount in bundle 4000m
System pods on AdminNode cpu resource request report: total 2769m nodeCount 2 min 1252m max 1517m avg 1384m tracked amount in bundle 4000m
System pods on AdminMaster cpu resource request report: total 923m nodeCount 1 min 923m max 923m avg 923m tracked amount in bundle 4000m
System pods on UserMaster memory resource request report: total 4524461824 nodeCount 2 min 2262230912 max 2262230912 avg 2262230912 tracked amount in bundle 8192Mi
System pods on AdminNode memory resource request report: total 6876Mi nodeCount 2 min 2174Mi max 4702Mi avg 3438Mi tracked amount in bundle 16384Mi
System pods on AdminMaster memory resource request report: total 465Mi nodeCount 1 min 465Mi max 465Mi avg 465Mi tracked amount in bundle 16384Mi
SUCCESS
Cluster is healthy.
```

[Next: Deploy user clusters.](#)

## 9. Deploy user clusters

With Anthos, organizations can scale their environments to incorporate multiple user clusters and segregate workloads between teams. A single admin cluster can support up to 20 user clusters, and each user cluster can support up to 250 nodes and 7500 pods.

To configure user clusters for your deployment, complete the following steps:

1. When the anthos-admin workstation is deployed, a file called `user-cluster.yaml` is created that can be used to deploy a number of additional user clusters for running workloads. Start by copying this default file with a new name for each cluster you intend to deploy.

```
ubuntu@gke-admin-ws-200915-151421:~ $ cp config.yaml anthos-cluster01-  
config.yaml
```

2. Edit the `anthos-cluster01-config.yaml` file so that it is specific for the environment that is being deployed.
  - a. In a manner similar to the `admin-config.yaml` used earlier, most of the variables are already filled in or they reference the admin-cluster for the information needed to deploy. This first section confirms the information for the version of Anthos being deployed and the vCenter instance it is being deployed on.

```
apiVersion: v1  
kind: UserCluster  
# (Required) A unique name for this cluster  
name: "anthos-cluster01"  
# (Required) GKE on-prem version (example: 1.3.0-gke.16)  
gkeOnPremVersion: 1.6.0-gke.7  
# # (Optional) vCenter configuration (default: inherit from the admin  
cluster)  
# vCenter:  
#   resourcePool: ""  
#   datastore: ""  
#   # Provide the path to vCenter CA certificate pub key for SSL  
#   verification  
#   caCertPath: ""  
#   # The credentials to connect to vCenter  
#   credentials:  
#     username: ""  
#     password: ""
```

- b. You must fill out the networking section next and select whether you are using static or DHCP mode. If you are using static addresses, you must create an IP-block file to supply addresses similar to the admin-cluster configuration.



The items under the hostConfig section are global for any time static IPs are used in a deployment. This includes both static IPs for the cluster and those used for the SeeSaw load balancers, which are configured later.

```
# (Required) Network configuration; vCenter section is optional and
inherits from
# the admin cluster if not specified
network:
# (Required) Hostconfig for static addresseses on Seesaw LB's
hostConfig:
  dnsServers:
    - "10.61.184.251"
    - "10.61.184.252"
  ntpServers:
    - "0.pool.ntp.org"
    - "1.pool.ntp.org"
    - "2.pool.ntp.org"
  searchDomainsForDNS:
    - "cie.netapp.com"
ipMode:
  # (Required) Define what IP mode to use ("dhcp" or "static")
  type: dhcp
  # # (Required when using "static" mode) The absolute or relative
path to the yaml file
  # # to use for static IP allocation
  # ipBlockFilePath: ""
# (Required) The Kubernetes service CIDR range for the cluster.
Must not overlap
  # with the pod CIDR range
serviceCIDR: 10.96.0.0/12
# (Required) The Kubernetes pod CIDR range for the cluster. Must
not overlap with
  # the service CIDR range
podCIDR: 192.168.0.0/16
vCenter:
  # vSphere network name
  networkName: VM_Network
```

- c. Next fill out the load balancer section. This can vary depending on the type of load balancer being deployed.

#### SeeSaw Example:

```
# (Required) Load balancer configuration
loadBalancer:
```

```

# (Required) The VIPs to use for load balancing
vips:
    # Used to connect to the Kubernetes API
    controlPlaneVIP: "10.63.172.156"
    # Shared by all services for ingress traffic
    ingressVIP: "10.63.172.157"
    # (Required) Which load balancer to use "F5BigIP" "Seesaw" or
    "ManualLB". Uncomment
        # the corresponding field below to provide the detailed spec
    kind: Seesaw
    # # (Required when using "ManualLB" kind) Specify pre-defined
nodeports
    # manualLB:
        # # NodePort for ingress service's http (only needed for user
cluster)
        #     ingressHTTPNodePort: 30243
        # # NodePort for ingress service's https (only needed for user
cluster)
        #     ingressHTTPSPort: 30879
        # # NodePort for control plane service
        #     controlPlaneNodePort: 30562
        # # NodePort for addon service (only needed for admin cluster)
        #     addonsNodePort: 0
        # # (Required when using "F5BigIP" kind) Specify the already-
existing partition and
        # # credentials
# f5BigIP:
    # address:
    # credentials:
        # username:
        # password:
    # partition:
        # # (Optional) Specify a pool name if using SNAT
        #     snatPoolName: ""
# (Required when using "Seesaw" kind) Specify the Seesaw configs
seesaw:
    # (Required) The absolute or relative path to the yaml file to
use for IP allocation
    # for LB VMs. Must contain one or two IPs.
    ipBlockFilePath: "anthos-cluster01-seesaw-block.yaml"
    # (Required) The Virtual Router IDentifier of VRRP for the Seesaw
group. Must
        # be between 1-255 and unique in a VLAN.
    vrid: 101
    # (Required) The IP announced by the master of Seesaw group
    masterIP: "10.63.172.153"

```

```

# (Required) The number CPUs per machine
cpus: 1
# (Required) Memory size in MB per machine
memoryMB: 2048
# (Optional) Network that the LB interface of Seesaw runs in
(default: cluster
  # network)
vCenter:
  # vSphere network name
  networkName: VM_Network
  # (Optional) Run two LB VMs to achieve high availability
(default: false)
  enableHA: false

```

- d. For a SeeSaw load balancer, you must create an additional external file to supply the static IP information for the load balancer. Create the file [anthos-cluster01-seesaw-block.yaml](#) that was referenced in this configuration section.

```

blocks:
- netmask: "255.255.255.0"
  gateway: "10.63.172.1"
  ips:
    - ip: "10.63.172.154"
      hostname: "anthos-cluster01-seesaw-vm"

```

## F5 BigIP Example:

```

loadBalancer:
# (Required) The VIPs to use for load balancing
vips:
  # Used to connect to the Kubernetes API
  controlPlaneVIP: "10.63.172.158"
  # Shared by all services for ingress traffic
  ingressVIP: "10.63.172.159"
# (Required) Which load balancer to use "F5BigIP" "Seesaw" or
"ManualLB". Uncomment
  # the corresponding field below to provide the detailed spec
  kind: F5BigIP
  # # (Required when using "ManualLB" kind) Specify pre-defined
nodeports
  # manualLB:
    #   # NodePort for ingress service's http (only needed for user
cluster)
    #     ingressHTTPNodePort: 30243
    #   # NodePort for ingress service's https (only needed for user

```

```

cluster)
  #   ingressHTTPSNODEPort: 30879
  #   # NodePort for control plane service
  #   controlPlaneNodePort: 30562
  #   # NodePort for addon service (only needed for admin cluster)
  #   addonsNodePort: 0
  # # (Required when using "F5BigIP" kind) Specify the already-existing partition and
  # # credentials
f5BigIP:
  address: "172.21.224.21"
  credentials:
    username: "admin"
    password: "admin-password"
    partition: "Anthos-Cluster-01"
  # # (Optional) Specify a pool name if using SNAT
  # snatPoolName: ""
  # (Required when using "Seesaw" kind) Specify the Seesaw configs
# seesaw:
  # (Required) The absolute or relative path to the yaml file to use for IP allocation
  # for LB VMs. Must contain one or two IPs.
  # ipBlockFilePath: ""
  # (Required) The Virtual Router IDentifier of VRRP for the Seesaw group. Must
    # be between 1-255 and unique in a VLAN.
  # vrid: 0
  # (Required) The IP announced by the master of Seesaw group
  # masterIP: ""
  # (Required) The number CPUs per machine
  # cpus: 4
  # (Required) Memory size in MB per machine
  # memoryMB: 8192
  # (Optional) Network that the LB interface of Seesaw runs in
(default: cluster
  # network)
  # vCenter:
    # vSphere network name
    #   networkName: VM_Network
  # (Optional) Run two LB VMs to achieve high availability
(default: false)
  # enableHA: false

```

- e. The final section describes the resources for the nodes that the cluster is deploying, including creating a node pool that we can use for dynamic scaling later. This section also supplies the service account keys to register the cluster with GKE once deployed.

```
# (Optional) User cluster master nodes must have either 1 or 3
replicas (default:
# 4 CPUs; 16384 MB memory; 1 replica)
masterNode:
  cpus: 4
  memoryMB: 8192
  # How many machines of this type to deploy
  replicas: 1
# (Required) List of node pools. The total un-tainted replicas across
all node pools
# must be greater than or equal to 3
nodePools:
- name: anthos-cluster01
  # # Labels to apply to Kubernetes Node objects
  # labels: {}
  # # Taints to apply to Kubernetes Node objects
  # taints:
  # - key: ""
  #   value: ""
  #   effect: ""
  cpus: 4
  memoryMB: 8192
  # How many machines of this type to deploy
  replicas: 3
# Spread nodes across at least three physical hosts (requires at
least three hosts)
antiAffinityGroups:
  # Set to false to disable DRS rule creation
  enabled: false
# # (Optional): Configure additional authentication
# authentication:
#   # (Optional) Configure OIDC authentication
#   oidc:
#     issuerURL: ""
#     kubectlRedirectURL: ""
#     clientID: ""
#     clientSecret: ""
#     username: ""
#     usernamePrefix: ""
#     group: ""
#     groupPrefix: ""
#     scopes: ""
#     extraParams: ""
#     # Set value to string "true" or "false"
#     deployCloudConsoleProxy: ""
```

```
#      # # The absolute or relative path to the CA file (optional)
#      # caPath: ""
#      # (Optional) Provide an additional serving certificate for the
API server
#      sni:
#          certPath: ""
#          keyPath: ""
#      # (Optional) Configure LDAP authentication (preview feature)
#      ldap:
#          name: ""
#          host: ""
#          # Only support "insecure" for now (optional)
#          connectionType: insecure
#          # # The absolute or relative path to the CA file (optional)
#          # caPath: ""
#          user:
#              baseDN: ""
#              userAttribute: ""
#              memberAttribute: ""
# (Optional) Specify which GCP project to connect your logs and
metrics to
stackdriver:
    projectID: "anthos-dev"
    # A GCP region where you would like to store logs and metrics for
this cluster.
    clusterLocation: "us-east1"
    enableVPC: false
    # The absolute or relative path to the key file for a GCP service
account used to
    # send logs and metrics from the cluster
    serviceAccountKeyPath: "/home/ubuntu/logging-monitoring-key.json"
# (Optional) Specify which GCP project to connect your GKE clusters
to
gkeConnect:
    projectID: "anthos-dev"
    # The absolute or relative path to the key file for a GCP service
account used to
    # register the cluster
    registerServiceAccountKeyPath: "/home/ubuntu/connect-register-
key.json"
    # The absolute or relative path to the key file for a GCP service
account used by
    # the GKE connect agent
    agentServiceAccountKeyPath: "/home/ubuntu/component-access-
key.json"
# (Optional) Specify Cloud Run configuration
```

```

cloudRun:
  enabled: false
# # (Optional/Alpha) Configure the GKE usage metering feature
# usageMetering:
#   bigQueryProjectID: ""
#   # The ID of the BigQuery Dataset in which the usage metering data
#   will be stored
#   bigQueryDatasetID: ""
#   # The absolute or relative path to the key file for a GCP service
#   account used by
#   # gke-usage-metering to report to BigQuery
#   bigQueryServiceAccountKeyPath: ""
#   # Whether or not to enable consumption-based metering
#   enableConsumptionMetering: false
# # (Optional/Alpha) Configure kubernetes apiserver audit logging
# cloudAuditLogging:
#   projectid: ""
#   # A GCP region where you would like to store audit logs for this
#   cluster.
#   clusterlocation: ""
#   # The absolute or relative path to the key file for a GCP service
#   account used to
#   # send audit logs from the cluster
#   serviceaccountkeypath: ""

```

3. After the edits to the configuration file are complete, NetApp recommends that the file be checked for proper syntax and spacing. You can check the config file you just created. This command references the `kubeconfig` file created by the admin-cluster.

```

ubuntu@gke-admin-200915-151421:~$ gkectl check-config --kubeconfig
kubeconfig --config anthos-cluster01-config.yaml

```

4. If you are using a SeeSaw load balancer, you need to create it prior to deploying the user cluster.

```

ubuntu@gke-admin-200915-151421:~$ gkectl create loadbalancer
--kubeconfig kubeconfig --config anthos-cluster-01-config.yaml

```

5. Create the user cluster. Just as we did with the admin cluster, the process can be accelerated by skipping the additional validations because we have already run the checks in the prior step.

```

ubuntu@gke-admin-200915-151421:~$ gkectl create cluster --config anthos-
cluster-01-config.yaml --skip-validation-all

```

6. When the cluster is deployed, it creates the kubeconfig file in the local directory. This file can be used to check the status of the cluster using kubectl or for running diagnostics with gkectl.

```
ubuntu@gke-admin-ws-200915-151421:~$ kubectl get nodes --kubeconfig anthos-cluster01-kubeconfig
NAME           STATUS  ROLES   AGE    VERSION
anthos-cluster01-7b5995cc45-ftrdw  Ready   <none>  5m    v1.18.6-
gke.6600
anthos-cluster01-7b5995cc45-z7q9b  Ready   <none>  5m    v1.18.6-
gke.6600
anthos-cluster01-7b5995cc45-zw6sv  Ready   <none>  6m    v1.18.6-
gke.6600
ubuntu@gke-admin-ws-200915-151421:~/ $ gkectl diagnose cluster
--kubeconfig kubeconfig --cluster-name anthos-cluster01
Diagnosing user cluster "anthos-cluster01"...

- Validation Category: User Cluster VCenter
Checking Credentials...SUCCESS
Checking VSphere CSI Driver...SUCCESS
Checking Version...SUCCESS
Checking Datacenter...SUCCESS
Checking Datastore...SUCCESS
Checking Resource pool...SUCCESS
Checking Folder...SUCCESS
Checking Network...SUCCESS
Checking Datastore...SUCCESS

- Validation Category: User Cluster
Checking onpremusercluster and onpremnodedpool...SUCCESS
Checking cluster object...SUCCESS
Checking machine deployment...SUCCESS
Checking machineset...SUCCESS
Checking machine objects...SUCCESS
Checking control place pods...SUCCESS
Checking gke-connect pods...SUCCESS
Checking config-management-system pods...Warning: No pod is running in namespace "config-management-system"...SUCCESS
Checking kube-system pods...SUCCESS
Checking gke-system pods...SUCCESS
Checking storage...SUCCESS
Checking resource...System pods on UserNode cpu resource request report:
total 3059m nodeCount 3 min 637m max 1224m avg 1019m tracked amount in bundle 4000m
System pods on UserNode memory resource request report: total 6464Mi nodeCount 3 min 1670Mi max 2945Mi avg 2259331754 tracked amount in bundle 8192Mi
SUCCESS
Cluster is healthy.
```

Next: [Enable access to the cluster with the GKE console](#).

## 10. Enable access to the cluster with the GKE console

After clusters are deployed and registered with Google Cloud, they must be logged into with the Google Cloud console to be managed and to receive additional cluster details. The official procedure to gain access to Anthos user clusters after they are deployed is detailed [here](#).



The project and the specific user must be whitelisted to access on-premises clusters in the Google Cloud console and use Anthos on VMware services. If you are unable to see the clusters after they are deployed, you might need to open a support ticket with Google.

The non-whitelisted view looks like this:

The following figures provides a view of clusters.

To enable access to your user clusters using the GKE console, complete the following steps:

1. Create a `node-reader.yaml` file that allows you to access the cluster.

```
kind: clusterrole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: node-reader
rules:
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["get", "list", "watch"]
```

2. Apply this file to the cluster that you want to log into with the `kubectl` command.

```
ubuntu@Anthos-Admin-Workstation:~$ kubectl apply -f node-reader.yaml
--kubeconfig anthos-cluster01-kubeconfig
clusterrole.rbac.authorization.k8s.io/node-reader created
```

3. Create a Kubernetes service account (KSA) that you can use to log in. Name this account after the user that uses this account to log into the cluster.

```
ubuntu@Anthos-Admin-Workstation:~$ kubectl create serviceaccount netapp-
user --kubeconfig anthos-cluster01-kubeconfig
serviceaccount/netapp-user created
```

4. Create cluster role-binding resources to bind both the view and newly created node-reader roles to the newly created KSA.

```
ubuntu@Anthos-Admin-Workstation:~$ kubectl create clusterrolebinding netapp-user-view --clusterrole view --serviceaccount default:netapp-user --kubeconfig anthos-cluster01-kubeconfig
clusterrolebinding.rbac.authorization.k8s.io/netapp-user-view created
ubuntu@Anthos-Admin-Workstation:~$ kubectl create clusterrolebinding netapp-user-node-reader --clusterrole node-reader --
--serviceaccount default:netapp-user --kubeconfig anthos-cluster01-kubeconfig
clusterrolebinding.rbac.authorization.k8s.io/netapp-user-node-reader created
```

5. If you need to extend permissions further, you can grant the KSA user a role with cluster admin permissions in a similar manner.

```
ubuntu@Anthos-Admin-Workstation:~$ kubectl create clusterrolebinding netapp-user-admin --clusterrole cluster-admin --serviceaccount default:netapp-user --kubeconfig anthos-cluster01-kubeconfig
clusterrolebinding.rbac.authorization.k8s.io/netapp-user-admin created
```

6. With the KSA account created and assigned with correct permissions, you can create a bearer token to allow access with the GKE Console. To do so, set a system variable for the secret name, and pass that variable through a `kubectl` command to generate the token.

```
ubuntu@Anthos-Admin-Workstation:~$ SECRET_NAME=$(kubectl get serviceaccount netapp-user --kubeconfig anthos-cluster01-kubeconfig -o jsonpath='{$.secrets[0].name}')
ubuntu@Anthos-Admin-Workstation:~$ kubectl get secret ${SECRET_NAME} --kubeconfig anthos-cluster01-kubeconfig -o jsonpath='{$.data.token}' | base64 -d
eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlc51dGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRLcy5pbv9zZXJ2aWN1YWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhWx0Iiwia3ViZXJuZXRLcy5pbv9zZXJ2aWN1YWNjb3VudC9zZWNyZXQubmFtZSI6Im51dGFwcC11c2VyLXRva2VuLWJxd3piIiwia3ViZXJuZXRLcy5pbv9zZXJ2aWN1YWNjb3VudC9zZXJ2aWN1LWFjY291bnQubmFtZSI6Im51dGFwcC11c2VyIiwia3ViZXJuZXRLcy5pbv9zZXJ2aWN1YWNjb3VudC9zZXJ2aWN1LWFjY291bnQudWlkIjoiNmIzZTFizjQtMDE3NS0xMWVhLWEzMGUtNmFizMrlYjYwNDBmIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6bmV0YXBwLXVzZXIfQ.YrHn4kYlb3gwxVKCLyo7p6J1f7mwwIgZqNw9eTvIkt4PfyR4IJHxQwawnJ4T6RljIFcbVSQwvWI1yGuTJ981ADdcwtFXHoEfMcOa6SIn4OMVw1d5BGloaESn8150VCK3xES2DHAmLexFBqhvBgcKZ0E4fZDvn4EhYvtFVpK1RbSyaE-DHD59P1bIgPdioiKREgbOddKdMn6XTVsuiP4V4tVKhktcdRNRAuw6cFDY1fPo13BFHr2aNBIe61FLkUqvQN-9nMd63JGdHL4hfXu6PPDxc9By6LgOW0nyaH4__gexy4uIa61fNLKV2SKe4_gAN41ffOCKe4Tq8sa6zMo-8g
```

- With this token, you can visit the [Google Cloud Console](#) and log in to the cluster by clicking the login button and pasting in the token.

## Log in to cluster

Choose the method you want to use for authentication to the cluster

Token

Dxc9By6LgOW0nyaH4\_\_gexy4ula61fNLKV2SKe4\_gAN41ffOCKe4Tq8sa6zMo-8g|

- Basic authentication
- Authenticate with Identity Provider configured for the cluster

[CLOSE](#) [LOGIN](#)

- After login is complete, you see a green check mark next to the cluster name, and information is displayed about the physical environment. Clicking the cluster name displays more verbose information.

[Next: Install and Configure NetApp Trident Storage Provisioner.](#)

### 11. Install and configure NetApp Trident storage provisioner

Trident is a storage orchestrator for containers. With Trident, microservices and containerized applications can take advantage of enterprise-class storage services provided by the full NetApp portfolio of storage systems for persistent storage mounts. Depending on an application's requirements, Trident dynamically provisions storage for ONTAP-based products such as NetApp AFF and FAS systems and Element storage systems like NetApp SolidFire and NetApp HCI.

To install Trident on the deployed user cluster and provision a persistent volume, complete the following steps:



The following instructions are screen-capped from a Trident 21.01 install, but the same steps to manually deploy the Trident Operator also apply to the current 21.04 release.

- Download the installation archive to the admin workstation and extract the contents. The current version of Trident is 21.04, which can be downloaded [here](#).

```
ubuntu@gke-admin-ws-200915-151421:~$ wget
https://github.com/NetApp/trident/releases/download/v21.01.0/trident-
installer-21.01.0.tar.gz
--2021-02-17 12:40:42--
https://github.com/NetApp/trident/releases/download/v21.01.0/trident-
installer-21.01.0.tar.gz
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-
```

```

releases.githubusercontent.com/77179634/0a63b600-6273-11eb-98df-
3d542851f6ff?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210217%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210217T173945Z&X-Amz-Expires=300&X-
Amz-
Signature=58f26bcac7eeee64673a84d46696490acec357b97a651af42653f973b778ee
88&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
21.01.0.tar.gz&response-content-type=application%2Foctet-stream
[following]
--2021-02-17 12:40:43-- https://github-
releases.githubusercontent.com/77179634/0a63b600-6273-11eb-98df-
3d542851f6ff?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210217%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210217T173945Z&X-Amz-Expires=300&X-
Amz-
Signature=58f26bcac7eeee64673a84d46696490acec357b97a651af42653f973b778ee
88&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
21.01.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-
releases.githubusercontent.com) ... 185.199.111.154, 185.199.108.154,
185.199.109.154, ...
Connecting to github-releases.githubusercontent.com (github-
releases.githubusercontent.com)|185.199.111.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38527217 (37M) [application/octet-stream]
Saving to: 'trident-installer-21.01.0.tar.gz'

100% [=====>] 38,527,217 84.9MB/s
in 0.4s

2021-02-17 12:40:44 (84.9 MB/s) - 'trident-installer-21.01.0.tar.gz'
saved [38527217/38527217]

```

## 2. Extract the Trident install from the downloaded bundle.

```

ubuntu@gke-admin-ws-200915-151421:~$ tar -xf trident-installer-
21.01.0.tar.gz
ubuntu@gke-admin-ws-200915-151421:~$ cd trident-installer

```

## 3. First set the location of the user cluster's `kubeconfig` file as an environment variable so that you don't

have to reference it, because Trident has no option to pass this file.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ export  
KUBECONFIG=~/anthos-cluster01-kubeconfig
```

4. The `trident-installer` directory contains manifests for defining all the required resources. Using the appropriate manifests, create the `TridentOrchestrator` custom resource definition.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl create -f  
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml  
customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.tride  
nt.netapp.io created
```

5. If a Trident namespace does not exist, create one in your cluster using the provided manifest.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl apply -f  
deploy/namespace.yaml  
namespace/trident created
```

6. Create the resources required for the Trident operator deployment, such as a `ServiceAccount` for the operator, a `ClusterRole` and `ClusterRoleBinding` to the `ServiceAccount`, a dedicated `PodSecurityPolicy`, or the operator itself.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl create -f  
deploy/bundle.yaml  
serviceaccount/trident-operator created  
clusterrole.rbac.authorization.k8s.io/trident-operator created  
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created  
deployment.apps/trident-operator created  
podsecuritypolicy.policy/tridentoperatorpods created
```

7. You can check the status of the operator after it's deployed with the following commands:

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl get  
deployment -n trident  
NAME          READY   UP-TO-DATE   AVAILABLE   AGE  
trident-operator   1/1      1           1           54s  
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl get pods  
-n trident  
NAME                           READY   STATUS    RESTARTS   AGE  
trident-operator-5c8bbf6754-h957z   1/1     Running   0          68s
```

8. With the operator deployed, we can now use it to install Trident. This requires creating a [TridentOrchestrator](#).

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl create -f
deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl describe
torc trident
Name:          trident
Namespace:
Labels:        <none>
Annotations:   <none>
API Version:  trident.netapp.io/v1
Kind:          TridentOrchestrator
Metadata:
  Creation Timestamp:  2021-02-17T18:25:43Z
  Generation:        1
  Managed Fields:
    API Version:  trident.netapp.io/v1
    Fields Type:   FieldsV1
    fieldsV1:
      f:spec:
        .:
        f:debug:
        f:namespace:
  Manager:      kubectl
  Operation:    Update
  Time:         2021-02-17T18:25:43Z
  API Version:  trident.netapp.io/v1
  Fields Type:   FieldsV1
  fieldsV1:
    f:status:
      .:
      f:currentInstallationParams:
        .:
        f:IPv6:
        f:autosupportHostname:
        f:autosupportImage:
        f:autosupportProxy:
        f:autosupportSerialNumber:
        f:debug:
        f:enableNodePrep:
        f:imagePullSecrets:
        f:imageRegistry:
        f:k8sTimeout:
        f:kubeletDir:
```

```

f:logFormat:
f:silenceAutosupport:
f:tridentImage:
f:message:
f:namespace:
f:status:
f:version:
Manager:          trident-operator
Operation:        Update
Time:             2021-02-17T18:25:43Z
Resource Version: 14836643
Self Link:
/apis/trident.netapp.io/v1/tridentorchestrators/trident
UID:              0e5f2c3b-6ca2-4b85-8453-0382e1426160
Spec:
Debug:            true
Namespace:        trident
Status:
Current Installation Params:
IPv6:
Autosupport Hostname:
Autosupport Image:
Autosupport Proxy:
Autosupport Serial Number:
Debug:
Enable Node Prep:
Image Pull Secrets:      <nil>
Image Registry:
k8sTimeout:
Kubelet Dir:
Log Format:
Silence Autosupport:
Trident Image:
Message:           Installing Trident
Namespace:         trident
Status:            Installing
Version:
Events:
  Type    Reason     Age     From                  Message
  ----  -----     ----   -----
  Normal  Installing  23s    trident-operator.netapp.io  Installing
Trident
  Normal  Installed   15s    trident-operator.netapp.io  Trident
installed

```

9. You can verify that Trident is successfully installed by checking the pods that are running in the namespace

or by using the `tridentctl` binary to check the installed version.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl get pod  
-n trident  
NAME                      READY   STATUS    RESTARTS   AGE  
trident-csi-2cp7x         2/2     Running   0          4m16s  
trident-csi-2xr5h         2/2     Running   0          4m16s  
trident-csi-bnwvh         2/2     Running   0          4m16s  
trident-csi-d6cfc6bb-1xm2p 6/6     Running   0          4m16s  
trident-operator-5c8bbf6754-h957z 1/1     Running   0          8m55s  
  
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ ./tridentctl -n  
trident version  
+-----+-----+  
| SERVER VERSION | CLIENT VERSION |  
+-----+-----+  
| 21.01.1        | 21.01.1       |  
+-----+-----+
```

10. The next step in enabling Trident integration with the NetApp HCI solution and Anthos is to create a backend that enables communication with the storage system. NetApp has been validated for several different protocols through the Anthos-ready partner storage validation program. This allows NetApp Trident to provide support in Anthos environments for NFS through our ONTAP platforms and iSCSI from both the ONTAP and Element storage used in NetApp HCI.



A NetApp HCI platform deploys with NetApp Element storage by default. In this guide we configure a backend for this system specifically. In addition to this, a customer can choose to connect to a remote ONTAP storage system or deploy an ONTAP Select software-defined storage system as a virtual appliance in VMware vSphere to provide additional NFS and iSCSI services. The configuration of each of these additional storage backends is beyond the scope of this guide.

11. There are sample backend files available in the downloaded installation archive in the `sample-input` folder. Copy `backend-solidfire.json` to your working directory and edit it to provide information detailing the storage system environment. For Element-based iSCSI connections, copy and edit the `backend-solidfire.json` file.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ cp sample-  
input/backend-solidfire.json ./  
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ $ vi backend-  
solidfire.json
```

- a. Edit the user, password, and MVIP value on the EndPoint line.
- b. Edit the SVIP value.

```
{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://trident:password@172.21.224.150/json-
rpc/8.0",
  "SVIP": "10.63.172.100:3260",
  "TenantName": "trident",
  "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS": 2000, "burstIOPS": 4000}}, {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS": 6000, "burstIOPS": 8000}}, {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS": 8000, "burstIOPS": 10000}}]
}
```

12. With this backend file in place, run the following command to create your first backend.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ ./tridentctl -n
trident create backend -f backend.json
+-----+
+-----+-----+-----+
|      NAME          | STORAGE DRIVER |           UUID
| STATE   | VOLUMES | 
+-----+-----+
+-----+-----+-----+
| solidfire-backend | solidfire-san | a5f9e159-c8f4-4340-a13a-
c615fef0f433 | online |       0 |
+-----+-----+
+-----+-----+-----+
```

13. With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ cp sample-
input/storage-class-csi.yaml.templ ./storage-class-basic.yaml
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ vi storage-class-
basic.yaml
```

14. The only edit that must be made to this file is to define the `backendType` value to the name of the storage driver from the newly created backend. Also note the `name-field` value that must be referenced in a later step.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
```

15. Run the `kubectl` command to create the storage class.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl create -f
sample-input/storage-class-basic.yaml
```

16. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample `pvc-basic.yaml` file that can be used to perform this action located in `sample-inputs` as well. The only edit that must be made to this file is ensuring that the `storageClassName` field matches the one just created.

```
ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ vi sample-
input/pvc-basic.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

17. Create the PVC by issuing the `kubectl` command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```

ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl create -f
sample-input/pvc-basic.yaml

ubuntu@gke-admin-ws-200915-151421:~/trident-installer$ kubectl get pvc
--watch

NAME      STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS   AGE
basic     Pending
basic      1s
basic     Pending    pvc-2azg0d2c-b13e-12e6-8d5f-5342040d22bf   0
basic      5s
basic     Bound     pvc-2azg0d2c-b13e-12e6-8d5f-5342040d22bf   1Gi
RWO        basic      7s

```

Next: Reference videos.

## Video demos

The following videos demonstrate some of the capabilities documented in this NVA.

- Deploying an application from the Google Cloud Application Marketplace to Anthos:
- <https://docs.netapp.com/us-en/netapp-solutions/media/Anthos-Deploy-App-Demo.mp4> (video)
- Dynamic scaling of Kubernetes clusters deployed on Anthos on VMware:
- <https://docs.netapp.com/us-en/netapp-solutions/media/Anthos-Scaling-Demo.mp4> (video)
- Using NetApp Trident to provision and attach a persistent volume to a Kubernetes pod on Anthos:
- <https://docs.netapp.com/us-en/netapp-solutions/media/Anthos-Trident-Demo.mp4> (video)

## Where to Find Additional Information: NetApp HCI with Anthos

To learn more about the information described in this document, review the following documents and/or websites:

- [Anthos Documentation](#)
- [NetApp HCI Documentation](#)
- [NetApp NDE 1.8 Deployment Guide](#)
- [NetApp Trident Documentation](#)
- [VMware vSphere 6.7U3 Documentation](#)
- [F5 Big-IP Documentation](#)

# Enterprise Applications and Databases

## Oracle Database

### Deploying Oracle Database

#### Solution Overview

##### Automated Deployment of Oracle19c for ONTAP on NFS

Organizations are automating their environments to gain efficiencies, accelerate deployments, and reduce manual effort. Configuration management tools like Ansible are being used to streamline enterprise database operations. In this solution, we demonstrate how you can use Ansible to automate the provisioning and configuration of Oracle 19c with NetApp ONTAP. By enabling storage administrators, systems administrators, and DBAs to consistently and rapidly deploy new storage, configure database servers, and install Oracle 19c software, you achieve the following benefits:

- Eliminate design complexities and human errors, and implement a repeatable consistent deployment and best practices
- Decrease time for provisioning of storage, configuration of DB hosts, and Oracle installation
- Increase database administrators, systems and storage administrators productivity
- Enable scaling of storage and databases with ease

NetApp provides customers with validated Ansible modules and roles to accelerate deployment, configuration, and lifecycle management of your Oracle database environment. This solution provides instruction and Ansible playbook code, to help you:

- Create and configure ONTAP NFS storage for Oracle Database
- Install Oracle 19c on RedHat Enterprise Linux 7/8 or Oracle Linux 7/8
- Configure Oracle 19c on ONTAP NFS storage

For more details or to begin, please see the overview videos below.

#### AWX/Tower Deployments

- Part 1: Getting Started, Requirements, Automation Details and Initial AWX/Tower Configuration

► [https://docs.netapp.com/us-en/netapp-solutions/media/oracle\\_deployment\\_auto\\_v1.mp4](https://docs.netapp.com/us-en/netapp-solutions/media/oracle_deployment_auto_v1.mp4) (video)

- Part 2: Variables and Running the Playbook

► [https://docs.netapp.com/us-en/netapp-solutions/media/oracle\\_deployment\\_auto\\_v2.mp4](https://docs.netapp.com/us-en/netapp-solutions/media/oracle_deployment_auto_v2.mp4) (video)

#### CLI Deployment

- Part 1: Getting Started, Requirements, Automation Details and Ansible Control Host Setup

► [https://docs.netapp.com/us-en/netapp-solutions/media/oracle\\_deployment\\_auto\\_v4.mp4](https://docs.netapp.com/us-en/netapp-solutions/media/oracle_deployment_auto_v4.mp4) (video)

- Part 2: Variables and Running the Playbook

► <https://docs.netapp.com/us-en/netapp-solutions/media/oracle3.mp4> (video)

## Getting started

This solution has been designed to be run in an AWX/Tower environment or by CLI on an Ansible control host.

## AWX/Tower

For AWX/Tower environments, you are guided through creating an inventory of your ONTAP cluster management and Oracle server (IPs and hostnames), creating credentials, configuring a project that pulls the Ansible code from NetApp Automation Github, and the Job Template that launches the automation.

1. Fill out the variables specific to your environment, and copy and paste them into the Extra Vars fields in your job template.
2. After the extra vars have been added to your job template, you can launch the automation.
3. The job template is run in three phases by specifying tags for `ontap_config`, `linux_config`, and `oracle_config`.

## CLI via the Ansible control host

1. To configure the Linux host so that it can be used as an Ansible control host  
[click here for RHEL 7/8 or CentOS 7/8](#), or  
[here for Ubuntu/Debian](#)
2. After the Ansible control host is configured, you can git clone the Ansible Automation repository.
3. Edit the hosts file with the IPs and/or hostnames of your ONTAP cluster management and Oracle server's management IPs.
4. Fill out the variables specific to your environment, and copy and paste them into the `vars.yml` file.
5. Each Oracle host has a variable file identified by its hostname that contains host-specific variables.
6. After all variable files have been completed, you can run the playbook in three phases by specifying tags for `ontap_config`, `linux_config`, and `oracle_config`.

## Requirements

Environment	Requirements
<b>Ansible environment</b>	AWX/Tower or Linux host to be the Ansible control host Ansible v.2.10 and higher Python 3 Python libraries - netapp-lib - xmltodict - jmespath
<b>ONTAP</b>	ONTAP version 9.3 - 9.7 Two data aggregates NFS vlan and ifgrp created

Environment	Requirements
Oracle server(s)	RHEL 7/8
	Oracle Linux 7/8
	Network interfaces for NFS, public, and optional mgmt
	Oracle installation files on Oracle servers

## Automation Details

This automated deployment is designed with a single Ansible playbook that consists of three separate roles. The roles are for ONTAP, Linux, and Oracle configurations.

The following table describes which tasks are being automated.

Role	Tasks
ontap_config	Pre-check of the ONTAP environment
	Creation of NFS based SVM for Oracle
	Creation of export policy
	Creation of volumes for Oracle
	Creation of NFS LIFs
linux_config	Create mount points and mount NFS volumes
	Verify NFS mounts
	OS specific configuration
	Create Oracle directories
	Configure hugepages
	Disable SELinux and firewall daemon
	Enable and start chronyd service
oracle_config	increase file descriptor hard limit
	Create pam.d session file
	Oracle software installation
	Create Oracle listener
	Create Oracle databases
	Oracle environment configuration
	Save PDB state
	Enable instance archive mode
	Enable DNFS client
	Enable database auto startup and shutdown between OS reboots

## Default parameters

To simplify automation, we have preset many required Oracle deployment parameters with default values. It is generally not necessary to change the default parameters for most deployments. A more advanced user can make changes to the default parameters with caution. The default parameters are located in each role folder under defaults directory.

## Deployment instructions

Before starting, download the following Oracle installation and patch files and place them in the `/tmp/archive` directory with read, write, and execute access for all users on each DB server to be deployed. The automation tasks look for the named installation files in that particular directory for Oracle installation and configuration.

```
LINUX.X64_193000_db_home.zip -- 19.3 base installer  
p31281355_190000_Linux-x86-64.zip -- 19.8 RU patch  
p6880880_190000_Linux-x86-64.zip -- opatch version 12.2.0.1.23
```

## License

You should read license information as stated in the Github repository. By accessing, downloading, installing, or using the content in this repository, you agree the terms of the license laid out [here](#).

Note that there are certain restrictions around producing and/or sharing any derivative works with the content in this repository. Please make sure you read the terms of the [License](#) before using the content. If you do not agree to all of the terms, do not access, download, or use the content in this repository.

After you are ready, click [here for detailed AWX/Tower deployment procedures](#) or [here for CLI deployment](#).

## Step-by-step deployment procedure

### AWX/Tower deployment Oracle 19c Database

#### 1. Create the inventory, group, hosts, and credentials for your environment

This section describes the setup of inventory, groups, hosts, and access credentials in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.

1. Configure the inventory.
  - a. Navigate to Resources → Inventories → Add, and click Add Inventory.
  - b. Provide the name and organization details, and click Save.
  - c. On the Inventories page, click the inventory created.
  - d. If there are any inventory variables, paste them in the variables field.
  - e. Navigate to the Groups sub-menu and click Add.
  - f. Provide the name of the group for ONTAP, paste the group variables (if any) and click Save.
  - g. Repeat the process for another group for Oracle.
  - h. Select the ONTAP group created, go to the Hosts sub-menu and click Add New Host.
  - i. Provide the IP address of the ONTAP cluster management IP, paste the host variables (if any), and

click Save.

- j. This process must be repeated for the Oracle group and Oracle host(s) management IP/hostname.
2. Create credential types. For solutions involving ONTAP, you must configure the credential type to match username and password entries.
  - a. Navigate to Administration → Credential Types, and click Add.
  - b. Provide the name and description.
  - c. Paste the following content in Input Configuration:

```
fields:  
  - id: username  
    type: string  
    label: Username  
  - id: password  
    type: string  
    label: Password  
    secret: true  
  - id: vsadmin_password  
    type: string  
    label: vsadmin_password  
    secret: true
```

- d. Paste the following content into Injector Configuration:

```
extra_vars:  
  password: '{{ password }}'  
  username: '{{ username }}'  
  vsadmin_password: '{{ vsadmin_password }}'
```

3. Configure the credentials.
  - a. Navigate to Resources → Credentials, and click Add.
  - b. Enter the name and organization details for ONTAP.
  - c. Select the custom Credential Type you created for ONTAP.
  - d. Under Type Details, enter the username, password, and vsadmin\_password.
  - e. Click Back to Credential and click Add.
  - f. Enter the name and organization details for Oracle.
  - g. Select the Machine credential type.
  - h. Under Type Details, enter the Username and Password for the Oracle hosts.
  - i. Select the correct Privilege Escalation Method, and enter the username and password.

## 2. Create a project

1. Go to Resources → Projects, and click Add.

- a. Enter the name and organization details.
- b. Select Git in the Source Control Credential Type field.
- c. enter [https://github.com/NetApp-Automation/na\\_oracle19c\\_deploy.git](https://github.com/NetApp-Automation/na_oracle19c_deploy.git) as the source control URL.
- d. Click Save.
- e. The project might need to sync occasionally when the source code changes.

### 3. Configure Oracle host\_vars

The variables defined in this section are applied to each individual Oracle server and database.

1. Input your environment-specific parameters in the following embedded Oracle hosts variables or host\_vars form.



The items in blue must be changed to match your environment.

Unresolved directive in ent-apps-db/awx\_automation.adoc - include::ent-apps-db/host\_vars.adoc[]

- a. Fill in all variables in the blue fields.
- b. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower.
- c. Navigate back to AWX or Tower and go to Resources → Hosts, and select and open the Oracle server configuration page.
- d. Under the Details tab, click edit and paste the copied variables from step 1 to the Variables field under the YAML tab.
- e. Click Save.
- f. Repeat this process for any additional Oracle servers in the system.

### 4. Configure global variables

Variables defined in this section apply to all Oracle hosts, databases, and the ONTAP cluster.

1. Input your environment-specific parameters in following embedded global variables or vars form.



The items in blue must be changed to match your environment.

Unresolved directive in ent-apps-db/awx\_automation.adoc - include::ent-apps-db/vars.adoc[]

2. Fill in all variables in blue fields.
3. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower into the following job template.

### 5. Configure and launch the job template.

1. Create the job template.
  - a. Navigate to Resources → Templates → Add and click Add Job Template.
  - b. Enter the name and description
  - c. Select the Job type; Run configures the system based on a playbook, and Check performs a dry run of a playbook without actually configuring the system.

- d. Select the corresponding inventory, project, playbook, and credentials for the playbook.
  - e. Select the all\_playbook.yml as the default playbook to be executed.
  - f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.
  - g. Check the box Prompt on Launch in the Job Tags field.
  - h. Click Save.
2. Launch the job template.
- a. Navigate to Resources → Templates.
  - b. Click the desired template and then click Launch.
  - c. When prompted on launch for Job Tags, type in requirements\_config. You might need to click the Create Job Tag line below requirements\_config to enter the job tag.
-  requirements\_config ensures that you have the correct libraries to run the other roles.
- d. Click Next and then Launch to start the job.
  - e. Click View → Jobs to monitor the job output and progress.
  - f. When prompted on launch for Job Tags, type in ontap\_config. You might need to click the Create "Job Tag" line right below ontap\_config to enter the job tag.
  - g. Click Next and then Launch to start the job.
  - h. Click View → Jobs to monitor the job output and progress
  - i. After the ontap\_config role has completed, run the process again for linux\_config.
  - j. Navigate to Resources → Templates.
  - k. Select the desired template and then click Launch.
  - l. When prompted on launch for the Job Tags type in linux\_config, you might need to select the Create "job tag" line right below linux\_config to enter the job tag.
  - m. Click Next and then Launch to start the job.
  - n. Select View → Jobs to monitor the job output and progress.
  - o. After the linux\_config role has completed, run the process again for oracle\_config.
  - p. Go to Resources → Templates.
  - q. Select the desired template and then click Launch.
  - r. When prompted on launch for Job Tags, type oracle\_config. You might need to select the Create "Job Tag" line right below oracle\_config to enter the job tag.
  - s. Click Next and then Launch to start the job.
  - t. Select View → Jobs to monitor the job output and progress.

## 6. Deploy additional database on same Oracle host

The Oracle portion of the playbook creates a single Oracle container database on an Oracle server per execution. To create additional container databases on the same server, complete the following steps.

1. Revise host\_vars variables.
  - a. Go back to step 2 - Configure Oracle host\_vars.

- b. Change the Oracle SID to a different naming string.
  - c. Change the listener port to different number.
  - d. Change the EM Express port to a different number if you are installing EM Express.
  - e. Copy and paste the revised host variables to the Oracle Host Variables field in the Host Configuration Detail tab.
2. Launch the deployment job template with only the oracle\_config tag.

Unresolved directive in ent-apps-db/awx\_automation.adoc - include::ent-apps-db/validation.adoc[]

#### Step-by-step deployment procedure

### CLI deployment Oracle 19c Database

This section covers the steps required to prepare and deploy Oracle19c Database with the CLI. Make sure that you have reviewed the [Getting Started and Requirements section](#) and prepared your environment accordingly.

#### Download Oracle19c repo

1. From your ansible controller, run the following command:

```
git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
```

2. After downloading the repository, change directories to na\_oracle19c\_deploy <cd na\_oracle19c\_deploy>.

#### Edit the hosts file

Complete the following before deployment:

1. Edit your hosts file na\_oracle19c\_deploy directory.
2. Under [ontap], change the IP address to your cluster management IP.
3. Under the [oracle] group, add the oracle hosts names. The host name must be resolved to its IP address either through DNS or the hosts file, or it must be specified in the host.
4. After you have completed these steps, save any changes.

The following example depicts a host file:

```
#ONTAP Host<div>
[ontap]
<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>10.61.184.183<i></i></span>
</div>
#Oracle hosts<div>
<div>
[oracle]<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>rtpora01<i></i></span>
<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>rtpora02<i></i></span>
</div>
```

This example executes the playbook and deploys oracle 19c on two oracle DB servers concurrently. You can also test with just one DB server. In that case, you only need to configure one host variable file.



The playbook executes the same way regardless of how many Oracle hosts and databases you deploy.

### Edit the `host_name.yml` file under `host_vars`

Each Oracle host has its host variable file identified by its host name that contains host-specific variables. You can specify any name for your host. Edit and copy the `host_vars` from the Host VARS Config section and paste it into your desired `host_name.yml` file.



The items in blue must be changed to match your environment.

Unresolved directive in `ent-apps-db/cli_automation.adoc` - include::`ent-apps-db/host_vars.adoc`[]

### Edit the `vars.yml` file

The `vars.yml` file consolidates all environment-specific variables (ONTAP, Linux, or Oracle) for Oracle deployment.

- Edit and copy the variables from the VARS section and paste these variables into your `vars.yml` file.

Unresolved directive in `ent-apps-db/cli_automation.adoc` - include::`ent-apps-db/vars.adoc`[]

### Run the playbook

After completing the required environment prerequisites and copying the variables into `vars.yml` and `your_host.yml`, you are now ready to deploy the playbooks.



<username> must be changed to match your environment.

1. Run the ONTAP playbook by passing the correct tags and ONTAP cluster username. Fill the password for ONTAP cluster, and vsadmin when prompted.

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t  
ontap_config -e @vars/vars.yml
```

2. Run the Linux playbook to execute Linux portion of deployment. Input for admin ssh password as well as sudo password.

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t  
linux_config -e @vars/vars.yml
```

3. Run the Oracle playbook to execute Oracle portion of deployment. Input for admin ssh password as well as sudo password.

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t  
oracle_config -e @vars/vars.yml
```

## Deploy Additional Database on Same Oracle Host

The Oracle portion of the playbook creates a single Oracle container database on an Oracle server per execution. To create additional container database on the same server, complete the following steps:

1. Revise the host\_vars variables.
  - a. Go back to step 3 - Edit the `host_name.yml` file under `host_vars`.
  - b. Change the Oracle SID to a different naming string.
  - c. Change the listener port to different number.
  - d. Change the EM Express port to a different number if you have installed EM Express.
  - e. Copy and paste the revised host variables to the Oracle host variable file under `host_vars`.
2. Execute the playbook with the `oracle_config` tag as shown above in [Run the playbook](#).

Unresolved directive in ent-apps-db/cli\_automation.adoc - include::ent-apps-db/validation.adoc[]

## Microsoft SQL Server

# TR-4897: SQL Server on Azure NetApp Files - Real Deployment View

Niyaz Mohamed, NetApp

IT organizations face constant change. Gartner reports nearly 75% of all databases will require cloud-based storage by 2022. As a leading relational database management system (RDBMS), Microsoft SQL Server is the go-to choice for Windows platform-designed applications and organizations that rely on SQL Server for everything from enterprise resource planning (ERP) to analytics to content management. SQL Server has helped to revolutionize the way enterprises manage massive data sets and power their applications to meet the schema and query performance demands.

Most IT organizations follow a cloud-first approach. Customers in a transformation phase evaluate their current IT landscape and then migrate their database workloads to the cloud based on an assessment and discovery exercise. Some factors driving customers toward cloud migration include elasticity/burst, data center exit, data center consolidation, end-of-life scenarios, mergers, acquisitions, and so on. The reason for migration can vary based on each organization and their respective business priorities. When moving to the cloud, choosing the right cloud storage is very important in order to unleash the power of SQL Server database cloud deployment.

## Use case

Moving the SQL Server estate to Azure and integrating SQL Server with Azure's vast array of platform-as-a-service (PaaS) features such as Azure Data Factory, Azure IoT Hub, and Azure Machine Learning creates tremendous business value to support digital transformation. Adopting the cloud also enables the respective business unit to focus on productivity and delivering new features and enhancements faster (Dev/Test use case) than relying on the CAPEX model or traditional private cloud models. This document covers a real-time deployment of SQL Server Always On availability group (AOAG) on Azure NetApp Files leveraging Azure Virtual Machines.

Azure NetApp Files provides enterprise-grade storage with continuously available file shares. Continuously available shares are required by SQL Server production databases on SMB file share to make sure that the node always has access to the database storage, including during disruptive scenarios such as controller upgrades or failures. Continuously available file shares eliminate the need to replicate data between storage nodes. Azure NetApp Files uses SMB 3.0 scale-out, persistent handles, and transparent failover to support nondisruptive operations (NDOs) for planned and unplanned downtime events, including many administrative tasks.

When planning cloud migrations, you should always evaluate the best approach to use. The most common and easiest approach for application migration is rehosting (also known as lift and shift). The example scenario provided in this document uses the rehosting method. SQL Server on Azure virtual machines with Azure NetApp Files allows you to use full versions of SQL Server in the cloud without having to manage on-premises hardware. SQL Server virtual machines (VMs) also simplify licensing costs when you pay as you go and provides elasticity and bursting capabilities for development, test, and estate refresh scenarios.

## Factors to consider

### VM performance

Selecting the right VM size is important for optimal performance of a relational database in a public cloud. Microsoft recommends that you continue using the same database performance-tuning options that are applicable to SQL Server in on-premises server environments. Use [memory-optimized](#) VM sizes for the best performance of SQL Server workloads. Collect the performance data of existing deployment to identify the RAM and CPU utilization while choosing the right instances. Most deployments choose between the D, E, or M series.

### Notes:

- For the best performance of SQL Server workloads, use memory-optimized VM sizes.
- NetApp and Microsoft recommend that you identify the storage performance requirements before choosing the instance type with the appropriate memory-to-vCore ratio. This also helps select a lower-instance type with the right network bandwidth to overcome storage throughput limits of the VM.

## VM redundancy

To increase redundancy and high availability, SQL Server VMs should either be in the same [availability set](#) or different [availability zones](#). When creating Azure VMs, you must choose between configuring availability sets versus availability zones; an Azure VM cannot participate in both.

## High availability

For high availability, configuring SQL Server AOAG or Always On Failover Cluster Instance (FCI) is the best option. For AOAG, this involves multiple instances of SQL Server on Azure Virtual Machines in a virtual network. If high availability is required at the database level, consider configuring SQL Server availability groups.

## Storage configuration

Microsoft SQL Server can be deployed with an SMB file share as the storage option. Starting with SQL Server 2012, system databases (master, model, msdb, or tempdb), and user databases can be installed with Server Message Block (SMB) file server as a storage option. This applies to both SQL Server stand-alone and SQL Server FCI.



File share storage for SQL Server databases should support continuously available property. This provides uninterrupted access to the file-share data.

Azure NetApp Files provides high performing file storage to meet any demanding workload, and it reduces SQL Server TCO as compared to block storage solutions. With block storage, VMs have imposed limits on I/O and bandwidth for disk operations; network bandwidth limits alone are applied against Azure NetApp Files. In other words, no VM-level I/O limits are applied to Azure NetApp Files. Without these I/O limits, SQL Server running on smaller VMs connected to Azure NetApp Files can perform as well as SQL Server running on much larger VMs. Azure NetApp Files reduce SQL Server deployment costs by reducing compute and software licensing costs. For detailed cost analysis and performance benefits of using Azure NetApp Files for SQL Server deployment, see the [Benefits of using Azure NetApp Files for SQL Server deployment](#).

## Benefits

The benefits of using Azure NetApp Files for SQL Server include the following:

- Using Azure NetApp Files allows you to use smaller instances, thus reducing compute cost.
- Azure NetApp Files also reduces software licensing costs, which reduce the overall TCO.
- Volume reshaping and dynamic service level capability optimizes cost by sizing for steady-state workloads and avoiding overprovisioning.

## Notes:

- To increase redundancy and high availability, SQL Server VMs should either be in the same [availability set](#) or in different [availability zones](#). Consider file path requirements if user-defined data files are required; in which case, select SQL FCI over SQL AOAG.
- The following UNC path is supported: `\ANFSMB-b4ca.anf.test\SQLDB` and `\ANFSMB-b4ca.anf.test\SQLDB\`.

- The loopback UNC path is not supported.
- For sizing, use historic data from your on-premises environment. For OLTP workloads, match the target IOPS with performance requirements using workloads at average and peak times along with the disk reads/sec and disk writes/sec performance counters. For data warehouse and reporting workloads, match the target throughput using workloads at average and peak times and the disk read bytes/sec and disk write bytes/sec. Average values can be used in conjunction with volume reshaping capabilities.

### Create continuously available shares

Create continuously available shares with the Azure portal or Azure CLI. In the portal, select the Enable Continuous Availability property option. For the Azure CLI, specify the share as a continuously available share by using the `az netappfiles volume create --with-smb-continuously-avl $True`. To learn more about creating a new, continuous availability-enabled volume, see [Creating a Continuously Available Share](#).

### Notes:

- Enable continuous availability for the SMB volume as shown in the following image.
- If a non-administrator domain account is used, make sure the account has the required security privilege assigned.
- Set the appropriate permissions at the share level and proper file-level permissions.
- A continuously available property cannot be enabled on existing SMB volumes. To convert an existing volume to use a continuously available share, use NetApp Snapshot technology. For more information, see [Convert existing SMB volumes to use Continuous Availability](#).

## Create a volume



Basics    **Protocol**    Tags    Review + create

Configure access to your volume.

#### Access

Protocol type

NFS     SMB     Dual-protocol (NFSv3 and SMB)

#### Configuration

Active Directory \* ⓘ

10.0.0.100 - anf.test/join



Share name \* ⓘ

SQLDB

Enable Continuous Availability ⓘ



**Review + create**

< Previous

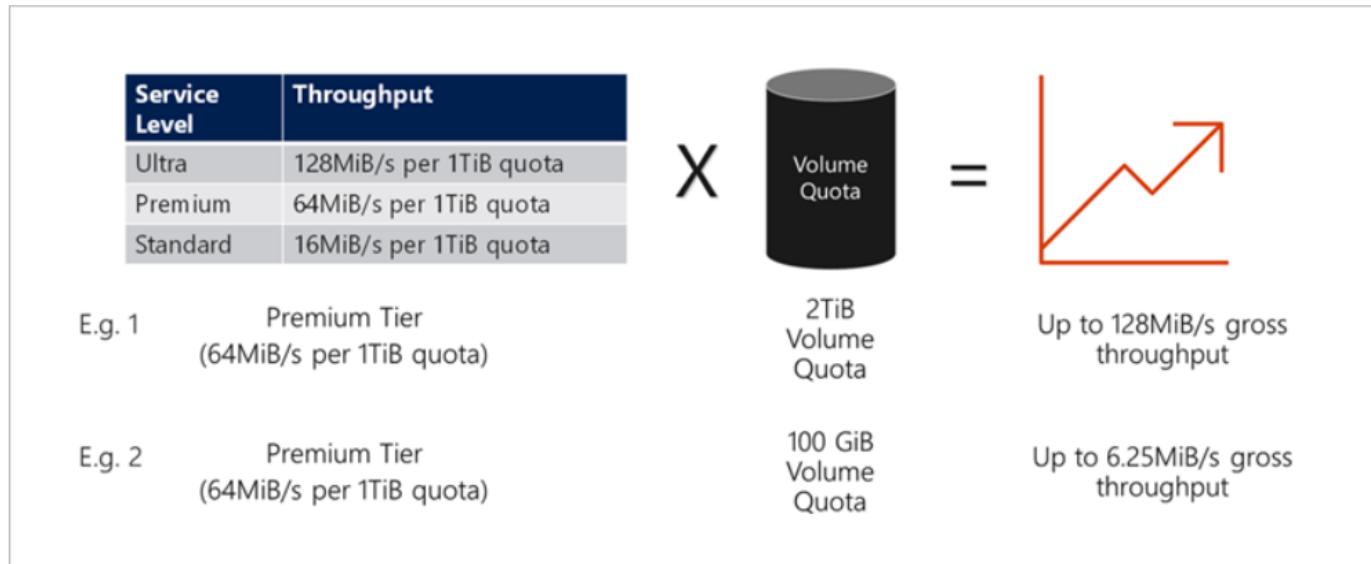
Next : Tags >

## Performance

Azure NetApp Files supports three service levels: Standard (16MBps per terabyte), Premium (64MBps per terabyte), and Ultra (128MBps per terabyte). Provisioning the right volume size is important for optimal performance of the database workload. With Azure NetApp Files, volume performance and the throughput limit are based on a combination of the following factors:

- The service level of the capacity pool to which the volume belongs
- The quota assigned to the volume
- The quality of service (QoS) type (auto or manual) of the capacity pool

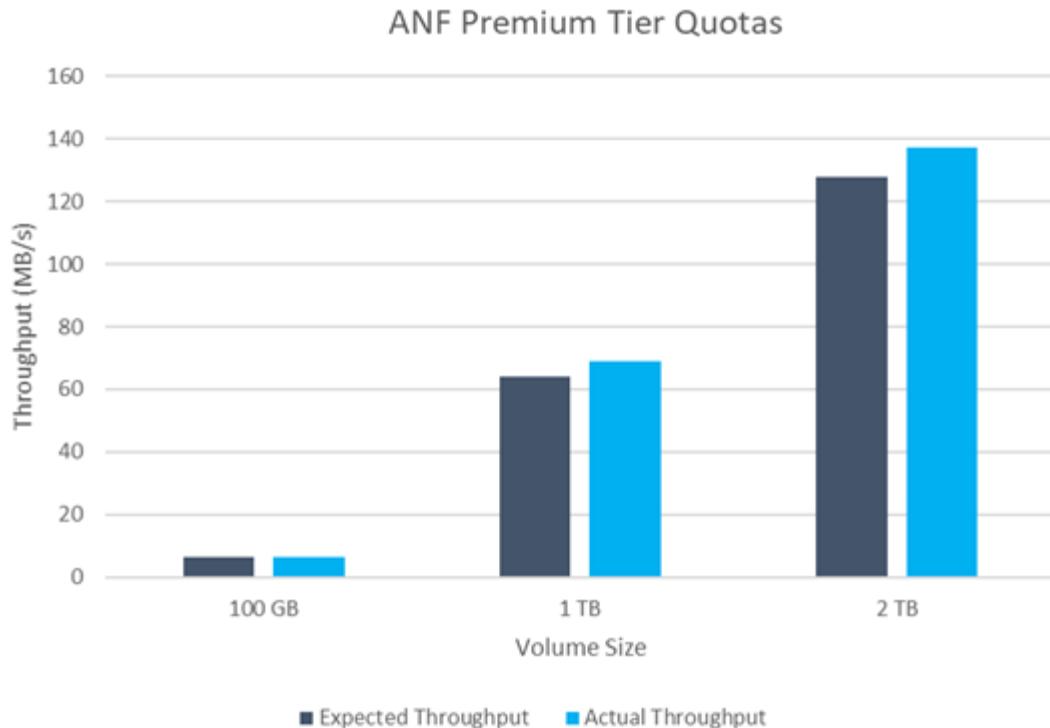
For more information, see [Service levels for Azure NetApp Files](#).



## Performance validation

As with any deployment, testing the VM and storage is critical. For storage validation, tools such as HammerDB, Apploader, the [SQL Server storage benchmark \(SB\) tool](#), or any custom script or FIO with the appropriate read/write mix should be used. Keep in mind however that most SQL Server workloads, even busy OLTP workloads, are closer to 80%–90% read and 10%–20% write.

To showcase performance, a quick test was performed against a volume using premium service levels. In this test, the volume size was increased from 100GB to 2TB on the fly without any disruption to application access and zero data migration.



Here is another example of real time performance testing with HammerDB performed for the deployment covered in this paper. For this testing, we used a small instance with eight vCPUs, a 500GB Premium SSD, and a 500GB SMB Azure NetApp Files volume. HammerDB was configured with 80 warehouses and eight users.

The following chart shows that Azure NetApp Files was able to deliver 2.6x the number of transactions per minute at 4x lower latency when using a comparable sized volume (500GB).

An additional test was performed by resizing to a larger instance with 32x vCPUs and a 16TB Azure NetApp Files volume. There was a significant increase in transactions per minute with consistent 1ms latency. HammerDB was configured with 80 warehouses and 64 users for this test.



## Cost optimization

Azure NetApp Files allows nondisruptive, transparent volume resizing and the ability to change the service levels with zero downtime and no effect on applications. This is a unique capability allowing dynamic cost management that avoids the need to perform database sizing with peak metrics. Rather, you can use steady state workloads, which avoids upfront costs. The volume reshaping and dynamic service-level change allows you to adjust the bandwidth and service level of Azure NetApp Files volumes on demand almost instantaneously without pausing I/O, while retaining data access.

Azure PaaS offerings such as LogicApp or Functions can be used to easily resize the volume based on a specific webhook or alert rule trigger to meet the workload demands while dynamically handling the cost.

For example, consider a database that needs 250MBps for steady state operation; however, it also requires a peak throughput of 400MBps. In this case, the deployment should be performed with a 4TB volume within the Premium service level to meet the steady-state performance requirements. To handle the peak workload, increase the volume size using Azure functions to 7TB for that specific period, and then downsize the volume to make the deployment cost effective. This configuration avoids overprovisioning of the storage.

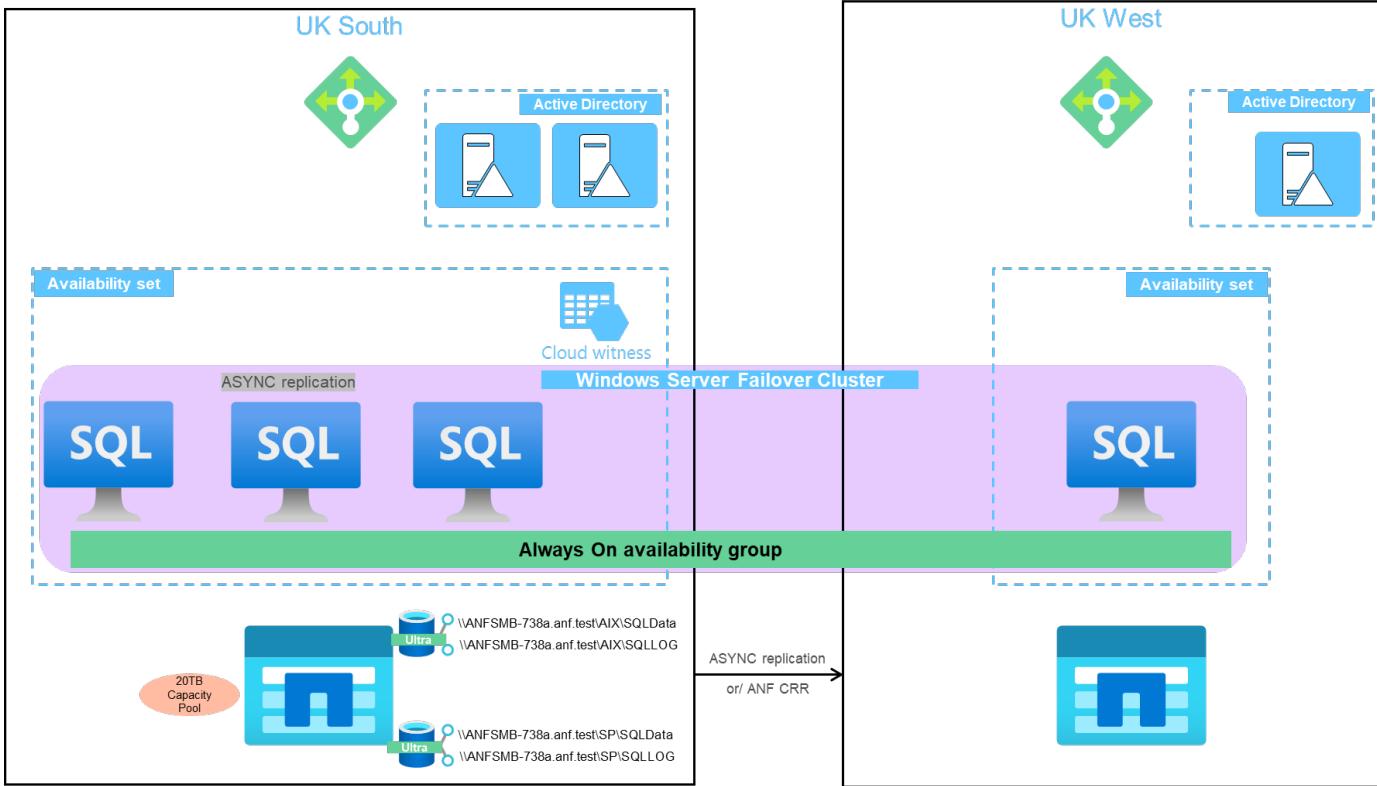
## Real-time, high-level reference design

This section covers a real-time deployment of a SQL database estate in an AOAG configuration using an Azure NetApp Files SMB volume.

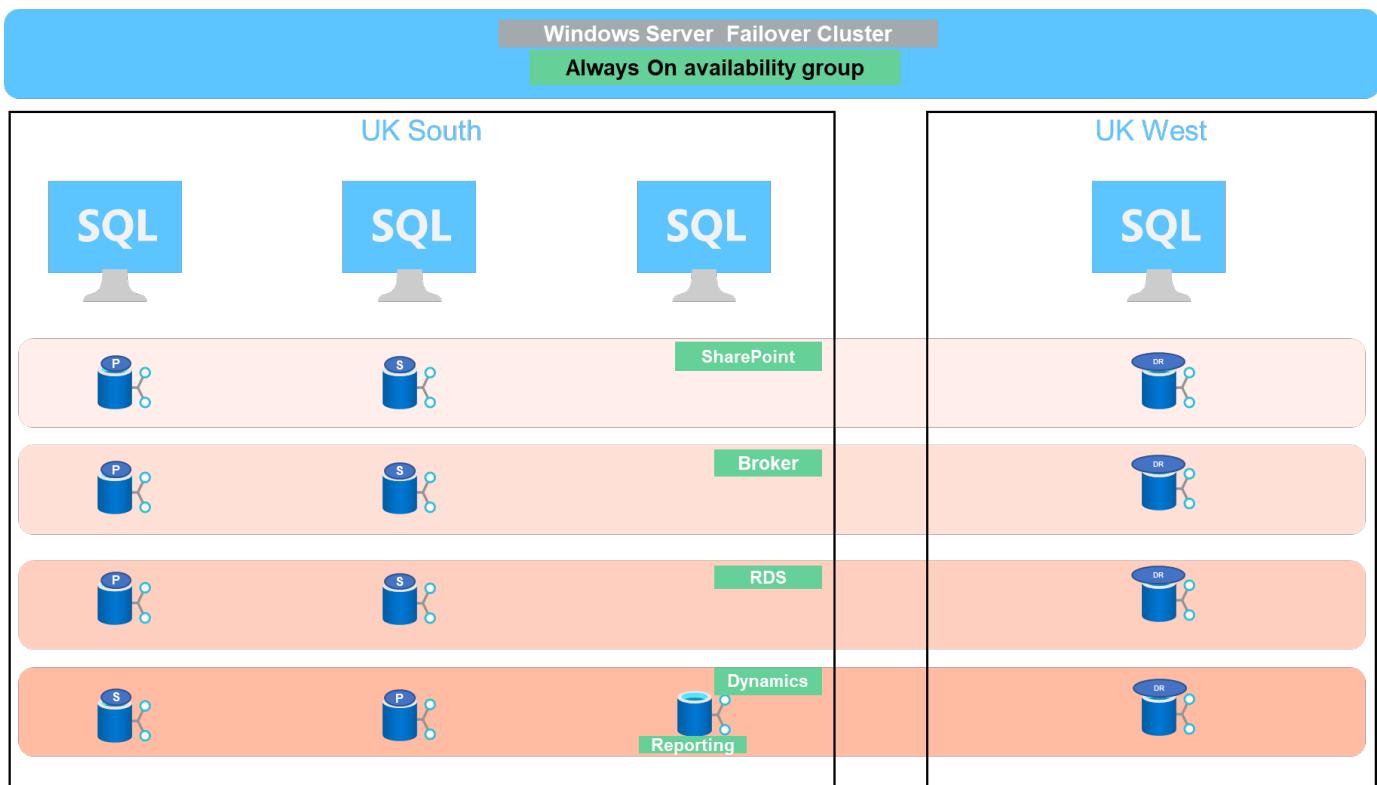
- Number of nodes: 4
- Number of databases: 21
- Number of availability groups: 4
- Backup retention: 7 days
- Backup archive: 365 days



Deploying FCI with SQL Server on Azure virtual machines with an Azure NetApp Files share provides a cost-efficient model with a single copy of the data. This solution can prevent add-file operation issues if the file path differs from the secondary replica.



The following image shows the databases within AOAG spread across the nodes.



#### Data layout

The user database files (.mdf) and user database transaction log files (.ldf) along with tempDB are stored on the same volume. The service level is Ultra.

The configuration consists of four nodes and four AGs. All 21 databases (part of Dynamic AX, SharePoint, RDS connection broker, and indexing services) are stored on the Azure NetApp Files volumes. The databases are balanced between the AOAG nodes to use the resources on the nodes effectively. Four D32 v3 instances are added in the WSFC, which participates in the AOAG configuration. These four nodes are provisioned in the Azure virtual network and are not migrated from on-premises.

#### Notes:

- If the logs require more performance and throughput depending on the nature of the application and the queries executed, the database files can be placed on the Premium service level, and the logs can be stored at the Ultra service level.
- If the tempdb files have been placed on Azure NetApp Files, then the Azure NetApp Files volume should be separated from the user database files. Here is an example distribution of the database files in AOAG.

#### Notes:

- To retain the benefits of Snapshot copy-based data protection, NetApp recommends not combining data and log data into the same volume.
- An add-file operation performed on the primary replica might fail on the secondary databases if the file path of a secondary database differs from the path of the corresponding primary database. This can happen if the share path is different on primary and secondary nodes (due to different computer accounts). This failure could cause the secondary databases to be suspended. If the growth or performance pattern cannot be predicted and the plan is to add files later, a SQL Server failover cluster with Azure NetApp Files is an acceptable solution. For most deployments, Azure NetApp Files meets the performance requirements.

## Migration

There are several ways to migrate an on-premises SQL Server user database to SQL Server in an Azure virtual machine. The migration can be either online or offline. The options chosen depend on the SQL Server version, business requirements, and the SLAs defined within the organization. To minimize downtime during the database migration process, NetApp recommends using either the AlwaysOn option or the transactional replication option. If it is not possible to use these methods, you can migrate the database manually.

The simplest and most thoroughly tested approach for moving databases across machines is backup and restore. Typically, you can start with a database backup followed by a copy of the database backup into Azure. You can then restore the database. For the best data transfer performance, migrate the database files into the Azure VM using a compressed backup file. The high-level design referenced in this document uses the backup approach to Azure file storage with Azure file sync and then restore to Azure NetApp files.



Azure Migrate can be used to discover, assess, and migrate SQL Server workloads.

To perform a migration, complete the following high-level steps:

1. Based on your requirements, set up connectivity.
2. Perform a full database backup to an on-premises file-share location.
3. Copy the backup files to an Azure file share with Azure file sync.
4. Provision the VM with the desired version of SQL Server.
5. Copy the backup files to the VM by using the `copy` command from a command prompt.
6. Restore the full databases to SQL Server on Azure virtual machines.



To restore 21 databases, it took approximately nine hours. This approach is specific to this scenario. However, other migration techniques listed below can be used based on your situation and requirements.

Other migration options to move data from an on-premises SQL Server to Azure NetApp Files include the following:

- Detach the data and log files, copy them to Azure Blob storage, and then attach them to SQL Server in the Azure VM with an ANF file share mounted from the URL.
- If you are using Always On availability group deployment on-premises, use the [Add Azure Replica Wizard](#) to create a replica in Azure and then perform failover.
- Use SQL Server [transactional replication](#) to configure the Azure SQL Server instance as a subscriber, disable replication, and point users to the Azure database instance.
- Ship the hard drive using the Windows Import/Export Service.

## Backup and recovery

Backup and recovery are an important aspect of any SQL Server deployment. It is mandatory to have the appropriate safety net to quickly recover from various data failure and loss scenarios in conjunction with high availability solutions such as AOAG. SQL Server Database Quiesce Tool, Azure Backup (streaming), or any third-party backup tool such as Commvault can be used to perform an application-consistent backup of the databases,

Azure NetApp Files Snapshot technology allows you to easily create a point-in-time (PiT) copy of the user databases without affecting performance or network utilization. This technology also allows you to restore a Snapshot copy to a new volume or quickly revert the affected volume to the state it was in when that Snapshot copy was created by using the revert volume function. The Azure NetApp Files snapshot process is very quick and efficient, which allows for multiple daily backups, unlike the streaming backup offered by Azure backup. With multiple Snapshot copies possible in a given day, the RPO and RTO times can be significantly reduced. To add application consistency so that data is intact and properly flushed to the disk before the Snapshot copy is taken, use the SQL Server database quiesce tool ([SCSQLAPI tool](#); access to this link requires NetApp SSO login credentials). This tool can be executed from within PowerShell, which quiesces the SQL Server database and in turn can take the application-consistent storage Snapshot copy for backups.

\*Notes: \*

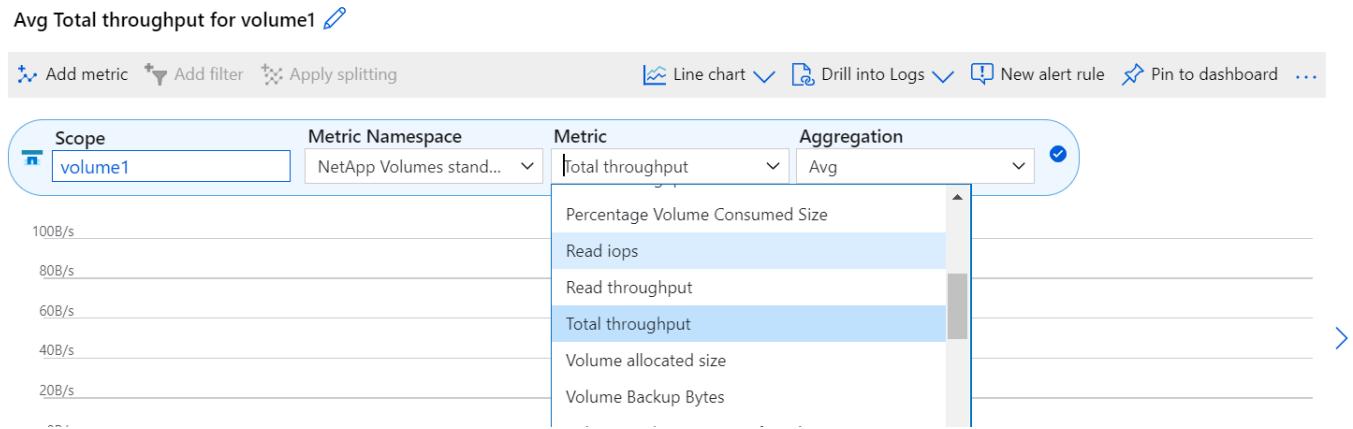
- The SCSSQLAPI tool only supports the 2016 and 2017 versions of SQL Server.
- The SCSSQLAPI tool only works with one database at a time.
- Isolate the files from each database by placing them onto a separate Azure NetApp Files volume.

Because of SCSSQL API's vast limitations, [Azure Backup](#) was used for data protection in order to meet the SLA requirements. It offers a stream-based backup of SQL Server running in Azure Virtual Machines and Azure NetApp Files. Azure Backup allows a 15-minute RPO with frequent log backups and PiT recovery up to one second.

## Monitoring

Azure NetApp Files is integrated with Azure Monitor for the time series data and provides metrics on allocated storage, actual storage usage, volume IOPS, throughput, disk read bytes/sec, disk write bytes/sec, disk reads/sec and disk writes/sec, and associated latency. This data can be used to identify bottlenecks with alerting and to perform health checks to verify that your SQL Server deployment is running in an optimal configuration.

In this HLD, ScienceLogic is used to monitor Azure NetApp Files by exposing the metrics using the appropriate service principal. The following image is an example of the Azure NetApp Files Metric option.



## Dev/Test using thick clones

With Azure NetApp Files, you can create instantaneous copies of databases to test functionality that should be implemented by using the current database structure and content during the application development cycles, to use the data extraction and manipulation tools when populating data warehouses, or to even recover data that was mistakenly deleted or changed. This process does not involve copying data from Azure Blob containers, which makes it very efficient. After the volume is restored, it can be used for read/write operations, which significantly reduces validation and time to market. This needs to be used in conjunction with SCSQLAPI for application consistency. This approach provides yet another continuous cost optimization technique along with Azure NetApp Files leveraging the Restore to New volume option.

### Notes:

- The volume created from the Snapshot copy using the Restore New Volume option consumes capacity from the capacity pool.
- You can delete the cloned volumes by using REST or Azure CLI to avoid additional costs (in case the capacity pool must be increased).

## Hybrid storage options

Although NetApp recommends using the same storage for all the nodes in SQL Server availability groups, there are scenarios in which multiple storage options can be used. This scenario is possible for Azure NetApp Files in which a node in AOAG is connected with an Azure NetApp Files SMB file share and the second node is connected with an Azure Premium disk. In these instances, make sure that the Azure NetApp Files SMB share is holding the primary copy of the user databases and the Premium disk is used as the secondary copy.

### Notes:

- In such deployments, to avoid any failover issues, make sure that continuous availability is enabled on the SMB volume. With no continuously available attribute, the database can fail if there is any background maintenance at the storage layer.
- Keep the primary copy of the database on the Azure NetApp Files SMB file share.

## Business continuity

Disaster recovery is generally an afterthought in any deployment. However, disaster recovery must be addressed during the initial design and deployment phase to avoid any impact to your business. With Azure

NetApp Files, the cross-region replication (CRR) functionality can be used to replicate the volume data at the block level to the paired region to handle any unexpected regional outage. The CRR-enabled destination volume can be used for read operations, which makes it an ideal candidate for disaster recovery simulations. In addition, the CRR destination can be assigned with the lowest service level (for instance, Standard) to reduce the overall TCO. In the event of a failover, replication can be broken, which makes the respective volume read/write capable. Also, the service level of the volume can be changed by using the dynamic service level functionality to significantly reduce disaster recovery cost. This is another unique feature of Azure NetApp Files with block replication within Azure.

## Long-term Snapshot copy archive

Many organizations must perform long-term retention of snapshot data from database files as a mandatory compliance requirement. Although this process is not used in this HLD, it can be easily accomplished by using a simple batch script using [AzCopy](#) to copy the snapshot directory to the Azure Blob container. The batch script can be triggered based on a specific schedule by using scheduled tasks. The process is straightforward—it includes the following steps:

1. Download the AzCopy V10 executable file. There is nothing to install because it is an [exe](#) file.
2. Authorize AzCopy by using a SAS token at the container level with the appropriate permissions.
3. After AzCopy is authorized, the data transfer begins.

### Notes:

- In batch files, make sure to escape the % characters that appear in SAS tokens. This can be done by adding an additional % character next to existing % characters in the SAS token string.
- The [Secure Transfer Required](#) setting of a storage account determines whether the connection to a storage account is secured with Transport Layer Security (TLS). This setting is enabled by default. The following batch script example recursively copies data from the Snapshot copy directory to a designated Blob container:

```
SET source="Z:\~snapshot"
echo %source%
SET
dest="https://testanfacct.blob.core.windows.net/azcopts?sp=racwdl&st=2020
-10-21T18:41:35Z&se=2021-10-22T18:41:00Z&sv=2019-12
-12&sr=c&sig=ZxRUJwF1LXgHS8As7HzXJOadXXVJ7PxxIX3ACpx56XY%%3D"
echo %dest%
```

The following example cmd is executed in PowerShell:

```
-recursive
```

```
INFO: Scanning...
INFO: Any empty folders will not be processed, because source and/or
destination doesn't have full folder support
Job b3731dd8-da61-9441-7281-17a4db09ce30 has started
Log file is located at: C:\Users\niyaz\.azcopy\b3731dd8-da61-9441-7281-
17a4db09ce30.log
0.0 %, 0 Done, 0 Failed, 2 Pending, 0 Skipped, 2 Total,
INFO: azcopy.exe: A newer version 10.10.0 is available to download
0.0 %, 0 Done, 0 Failed, 2 Pending, 0 Skipped, 2 Total,
Job b3731dd8-da61-9441-7281-17a4db09ce30 summary
Elapsed Time (Minutes): 0.0333
Number of File Transfers: 2
Number of Folder Property Transfers: 0
Total Number of Transfers: 2
Number of Transfers Completed: 2
Number of Transfers Failed: 0
Number of Transfers Skipped: 0
TotalBytesTransferred: 5
Final Job Status: Completed
```

## Notes:

- A similar backup feature for long-term retention will soon be available in Azure NetApp Files.
- The batch script can be used in any scenario that requires data to copied to Blob container of any region.

## Cost optimization

With volume reshaping and dynamic service level change, which is completely transparent to the database, Azure NetApp Files allows continuous cost optimizations in Azure. This capability is used in this HLD extensively to avoid overprovisioning of additional storage to handle workload spikes.

Resizing the volume can be easily accomplished by creating an Azure function in conjunction with the Azure alert logs.

## Conclusion

Whether you are targeting an all-cloud or hybrid cloud with stretch databases, Azure NetApp Files provides excellent options to deploy and manage the database workloads while reducing your TCO by making data requirements seamless to the application layer.

This document covers recommendations for planning, designing, optimizing, and scaling Microsoft SQL Server deployments with Azure NetApp Files, which can vary greatly between implementations. The right solution depends on both the technical details of the implementation and the business requirements driving the project.

## Takeaways

The key points of this document include:

- You can now use Azure NetApp Files to host the database and file share witness for SQL Server cluster.

- You can boost the application response times and deliver 99.9999% availability to provide access to SQL Server data when and where it is needed.
- You can simplify the overall complexity of the SQL Server deployment and ongoing management, such as raid striping, with simple and instant resizing.
- You can rely on intelligent operations features to help you deploy SQL Server databases in minutes and speed development cycles.
- If Azure Cloud is the destination, Azure NetApp Files is the right storage solution for optimized deployment.

## Where to find additional information

To learn more about the information described in this document, refer to the following website links:

- Solution architectures using Azure NetApp Files

<https://docs.microsoft.com/en-us/azure/azure-netapp-files/azure-netapp-files-solution-architectures>

- Benefits of using Azure NetApp Files for SQL Server deployment

<https://docs.microsoft.com/en-us/azure/azure-netapp-files/solutions-benefits-azure-netapp-files-sql-server>

- SQL Server on Azure Deployment Guide Using Azure NetApp Files

<https://www.netapp.com/pdf.html?item=/media/27154-tr-4888.pdf>

- Fault tolerance, high availability, and resilience with Azure NetApp Files

<https://cloud.netapp.com/blog/azure-anf-blr-fault-tolerance-high-availability-and-resilience-with-azure-netapp-files>

# Data Protection and Security

## Data Protection

### TR-4830: NetApp HCI Disaster Recovery with Cleondris

Michael White, NetApp

#### Overview of Business Continuity and Disaster Recovery

The business continuity and disaster recovery (BCDR) model is about getting people back to work. Disaster recovery focuses on bringing technology, such as an email server, back to life. Business continuity makes it possible for people to access that email server. Disaster recovery alone would mean that the technology is working, but nobody might be using it; BCDR means that people have started using the recovered technology.

#### Business Impact Assessment

It is hard to know what is required to make a tier 1 application work. It is usually obvious that authentication servers and DNS are important. But is there a database server somewhere too?

This information is critical because you need to package tier 1 applications so that they work in both a test failover and a real failover. An accounting firm can perform a business impact assessment (BIA) to provide you with all the necessary information to successfully protect your applications: for example, determining the required components, the application owner, and the best support person for the application.

#### Application Catalog

If you do not have a BIA, you can do a version of it yourself: an application catalog. It is often done in a spreadsheet with the following fields: application name, components, requirements, owner, support, support phone number, and sponsor or business application owner. Such a catalog is important and useful in protecting your applications. The help desk can sometimes help with an application catalog; they often have already started one.

#### What Not to Protect

There are applications that should not be protected. For example, you can easily and cheaply have a domain controller running as a virtual machine (VM) at your disaster recovery site, so there is no need to protect one. In fact, recovering a domain controller can cause issues during recovery. Monitoring software that is used in the production site does not necessarily work in the disaster recovery site if it is recovered there.

It is usually unnecessary to protect applications that can be protected with high availability. High availability is the best possible protection; its failover times are often less than a second. Therefore, disaster recovery orchestration tools should not protect these applications, but high availability can. An example is the software in banks that support ATMs.

You can tell that you need to look at high-availability solutions for an application when an application owner has a 20-second recovery time objective (RTO). That RTO is beyond replication solutions.

#### Product Overview

The Cleondris HCI Control Center (HCC) adds disaster recovery capabilities to new and existing NetApp HCI deployments. It is fully integrated with the NetApp SolidFire storage engine and can protect any kind of data and applications. When a customer site fails, HCC can be used to recover all data at a secondary NetApp HCI

site, including policy-based VM startup orchestration.

Setting up replication for multiple volumes can be time consuming and error prone when performed manually. HCC can help with its Replication Wizard. The wizard helps set up the replication correctly so that the servers can access the volumes if a disaster occurs. With HCC, the VMware environment can be started on the secondary system in a sandbox without affecting production. The VMs are started in an isolated network and a functional test is possible.

## Installing Cleondris: NetApp HCI DR with Cleondris

This section will detail the prerequisites and deployment steps for installing Cleondris.

### Prerequisites

There are several things to have ready before you start with the installation.

This technical report assumes that you have your NetApp HCI infrastructure working at both your production site and your disaster recovery site.

- **DNS.** You should have DNS prepared for your HCC disaster recovery tool when you install it.
- **FQDN.** A fully qualified domain name for the disaster recovery tool should be prepared before installation.
- **IP address.** The IP will be part of the FQDN before it is put into DNS.
- **NTP.** You need a Network Time Protocol (NTP) server address. It can be either your own internal or external address, but it needs to be accessible.
- **Storage location.** When you install HCC, you must know which datastore it should be installed to.
- **vCenter Server service account.** You will need to have a service account created in vCenter Server on both the disaster recovery and production side for HCC to use. It does not require administrator-level permissions at the root level. If you like, you can find exactly what is required in the HCC user guide.
- **NetApp HCI service account.** You need a service account in your NetApp HCI storage for both the disaster recovery and production side for HCC to use. Full access is required.
- **Test network.** This network should be connected to all your hosts in the disaster recovery site, and it should be isolated and nonrouting. This network is used to make sure applications work during a test failover. The built-in test network that is temporary only is a one-host network. Therefore, if your test failover has VMs scattered on multiple hosts, they will not be able to communicate. I recommend that you create a distributed port group in the disaster recovery site that spans all hosts but is isolated and nonrouting. Testing is important to success.
- **RTOs.** You should have RTOs approved by management for your application groups. Often it is 1 or 2 hours for tier 1 applications; for tier 4 applications, it can be as long as 12 hours. These decisions must be approved by management because they will determine how quickly things work after a critical outage. These times will determine replication schedules.
- **Application information.** You should know which application you need to protect first, and what it needs to work. For example, Microsoft Exchange needs a domain controller that has a role of Global Catalog to start. In my own experience, a customer said that they had one email server to protect. It did not test well, and when I investigated, I discovered the customer had 24 VMs that were part of the email application.

### Download Information

You can download HCC from the [Cleondris site](#). When you buy it, you receive an email with a download link as well.

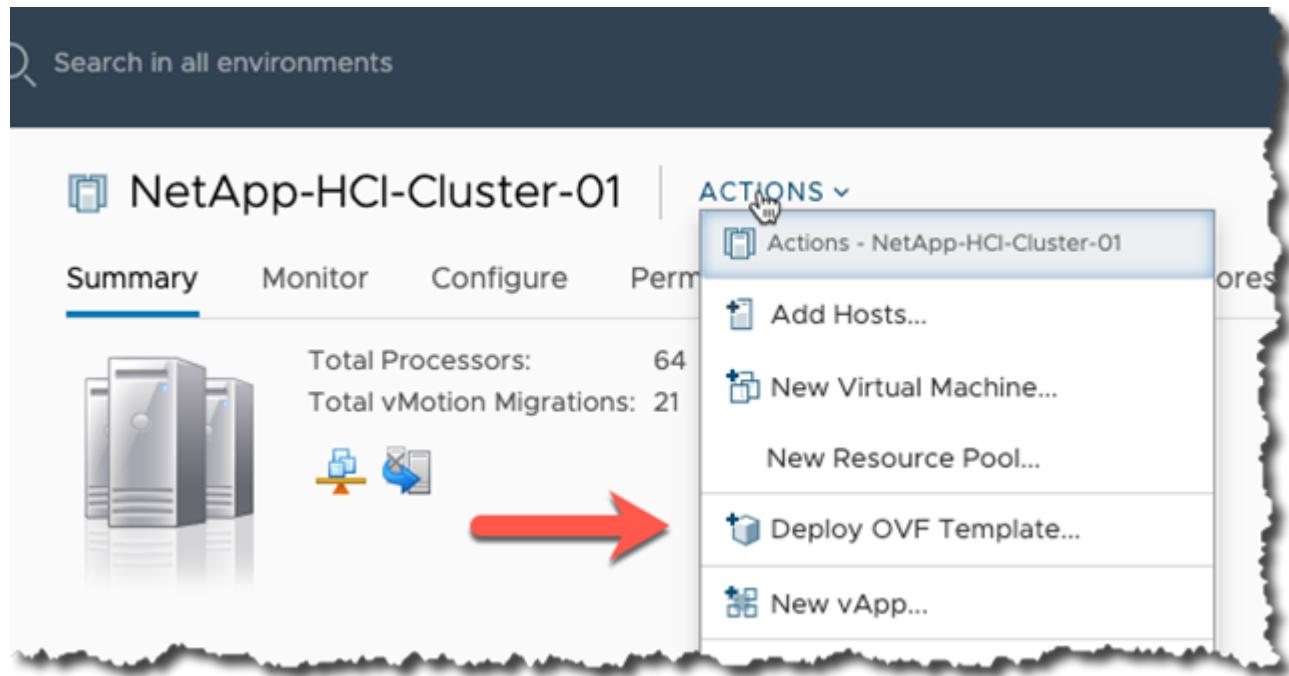
## License

Your license will arrive in an email when you purchase or if you get a not-for-resale (NFR) version. You can get a trial license through the [Cleondris Support Portal](#).

## Deployment

You download an OVF file, so it is deployed like many other things.

1. Start by using the Actions menu available at the cluster level.



2. Select the file.

## Deploy OVF Template

### 1 Select an OVF template

- 2 Select a name and folder
- 3 Select a compute resource
- 4 Review details
- 5 Select storage
- 6 Ready to complete

### Select an OVF template

Select an OVF template from remote URL or local file system

Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

URL

http | https://remoteserver-address/filetodeploy.ovf | .ova

Local file

Choose Files cleondris-appliance-1705.ova

3. Name the appliance and select the location for it in the vCenter infrastructure.

## Deploy OVF Template

✓ 1 Select an OVF template

**2 Select a name and folder**

3 Select a compute resource

4 Review details

5 Select storage

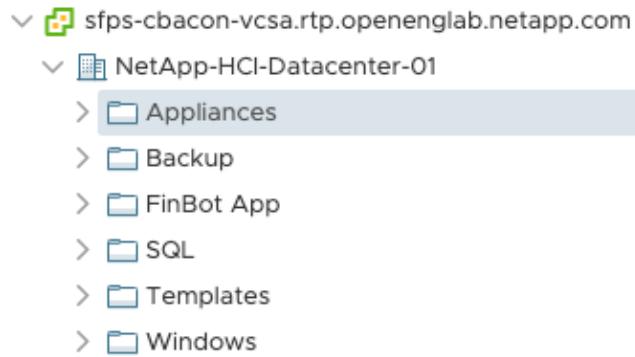
6 Ready to complete

Select a name and folder

Specify a unique name and target location

Virtual machine name: dr-hcc

Select a location for the virtual machine.



4. Select the Compute location.
5. Confirm the details.
6. Accept the license details.
7. Select the appropriate storage location.
8. Select the network that our appliance will work on.
9. Review the details again and click Finish.
10. Now wait for the appliance to be deployed, and then power it up. As it powers up, you might see a message saying that VMware tools are not installed. You can ignore this message; it will go away automatically.

### Initial Configuration

To start the initial configuration, complete the following steps:

1. This phase involves doing the configuration in the Appliance Configurator, which is the VM console. So, after the appliance powers up, change to work in the console by using the VMware Remote Console (VMRC) or the HTML5 VMRC version. Look for a blue Cleondris screen.

## Cleondris Appliance Configurator

The web GUI is available at

http(s)://10.193.136.224

http(s)://fe80::250:56ff:fe93:8b0a

Hostname: cdm.localdomain

MAC: 00:50:56:93:8B:0A

NTP time sync not available

Local Time: Thu Mar 5 20:04:14 2020 CET

Press any key to continue

2. Press any key to proceed, and configure the following:

- The web administrator password
- The network configuration: IP, DNS, and so on
- The time zone
- NTP

3. Select the Reboot and Activate Network/NTP Settings. You will see the appliance reboot. Afterward, do a ping test to confirm the FQDN and IP.

### Patching Cleondris

To update your Cleondris product, complete the following steps:

1. When you first log in to the appliance, you see a screen like the following:

## Almost done!



You have successfully installed this Cleondris appliance and configured it for network access.

To ensure the best experience, you now need to install the latest update of your Cleondris product which you can download from the Cleondris website.

Please select the .zip file containing the update:

No file chosen

Update

2. Click Choose File to select the update you downloaded from the Cleondris website.

## Almost done!

You have successfully installed this Cleondris appliance and configured it for network access.

To ensure the best experience, you now need to install the latest update of your Cleondris product which you can download from the Cleondris website.

Please select the .zip file containing the update:

cdm-linux-x64....4.2001P6.zip

Update

3. Upload the patch. After the appliance reboots, the following login screen is displayed:



4. You can now see the new version and build information; confirming that the update was successful. Now you can continue with the configuration.

#### Software Used

This technical report uses the following software versions:

- vSphere 6.5 on production
- vSphere 6.7 U3 on DR
- NetApp Element 11.5 on production
- NetApp Element 12.0 on DR
- Cleondris HCC 8.0.2007 Build 20200707-1555 and 8.0.2007X2 build 20200709-1936.

#### Configuring Cleondris: NetApp HCI DR with Cleondris

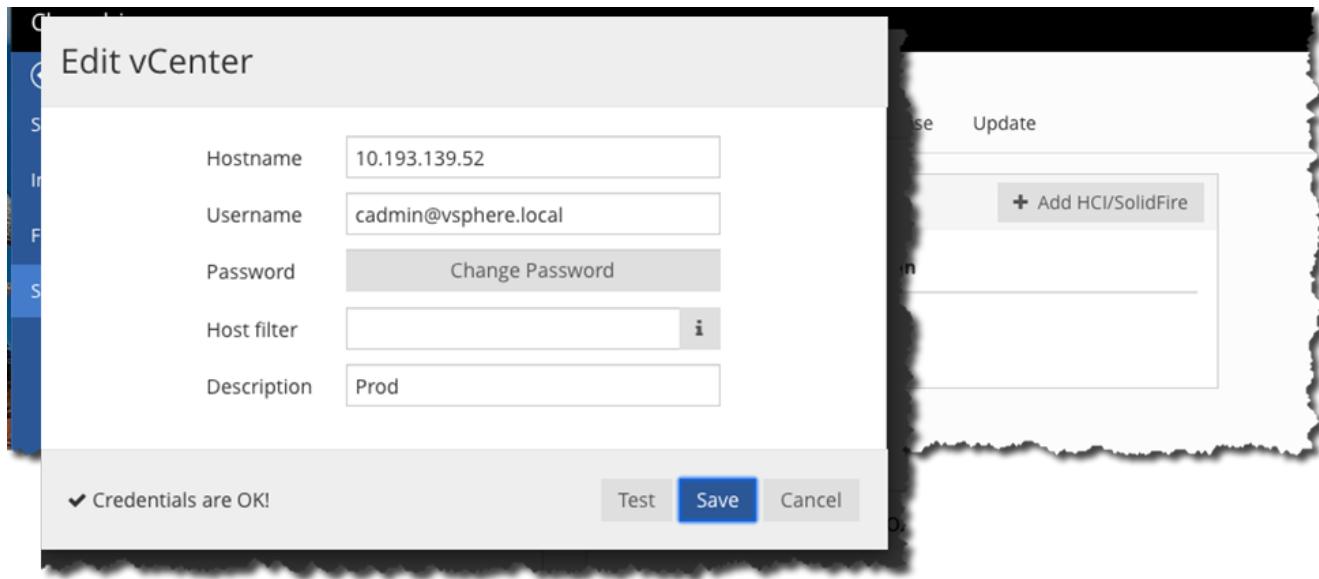
You now configure Cleondris to communicate with your vCenter Servers and storage. If you have logged out, returned, and log in again to start here, you are prompted for the following information:

1. Accept the EULA.
2. Copy and paste the license.
3. You are prompted to perform configuration, but skip this step for now. It is better to perform this configuration as detailed later in this paper.
4. When you log back in and see the green boxes, you must change to the Setup area.

#### Add vCenter Servers

To add the vCenter Servers, complete the following steps:

1. Change to the VMware tab and add your two vCenter Servers. When you are defining them, add a good description and use the Test button.



This example uses an IP address instead of an FQDN. (This FQDN didn't work at first; I later found out that I had not entered the proper DNS information. After correcting the DNS information, the FQDN worked fine.) Also notice the description, which is useful.

2. After both vCenter Servers are done, the screen displays them.

VMware		
VMware vCenter		
Hostname	Username	Description
sfps-megatron-vcsa.rtp.openenglab.netapp.com	cadmin@vsphere.local	Prod
sfps-cbacon-vcsa.rtp.openenglab.netapp.com	administrator@vsphere.local	DR

#### Add NetApp HCI Clusters

To add the NetApp HCI clusters, complete the following steps:

1. Change to the NetApp tab and add your production and disaster recovery storage. Again, add a good description and use the Test button.

## Register HCI/SolidFire

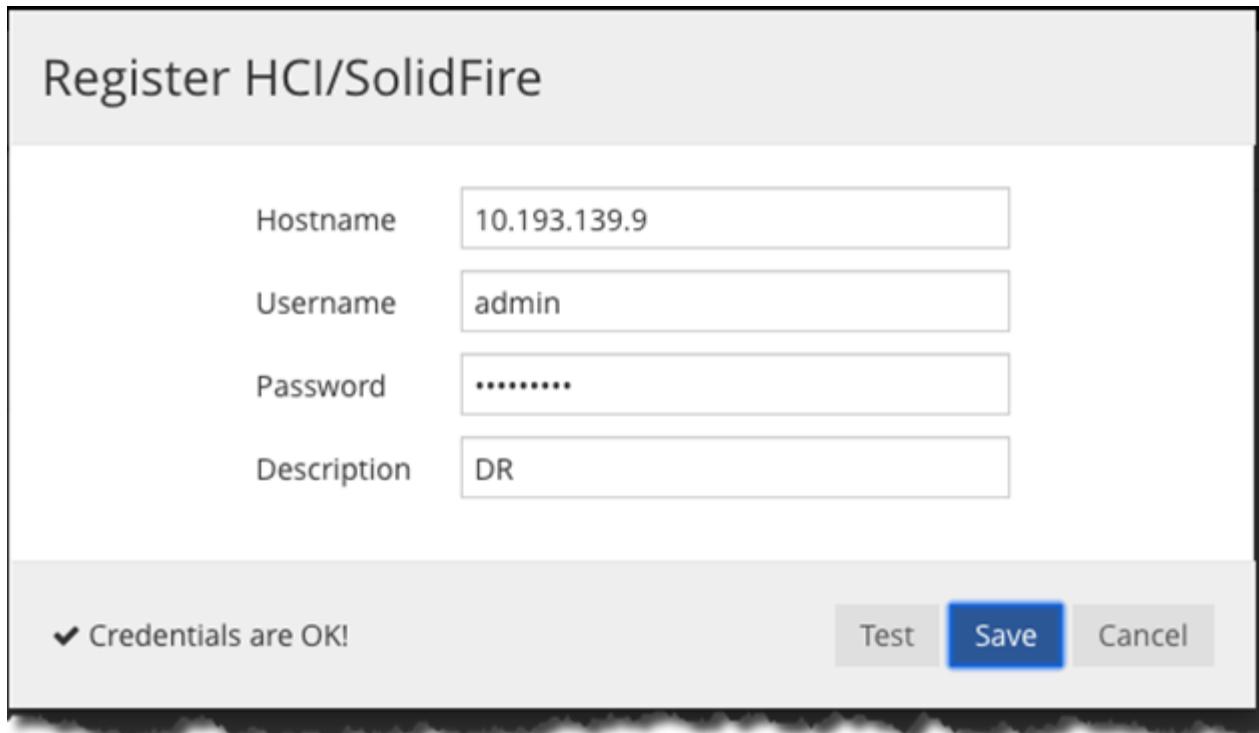
Hostname: 10.193.139.9

Username: admin

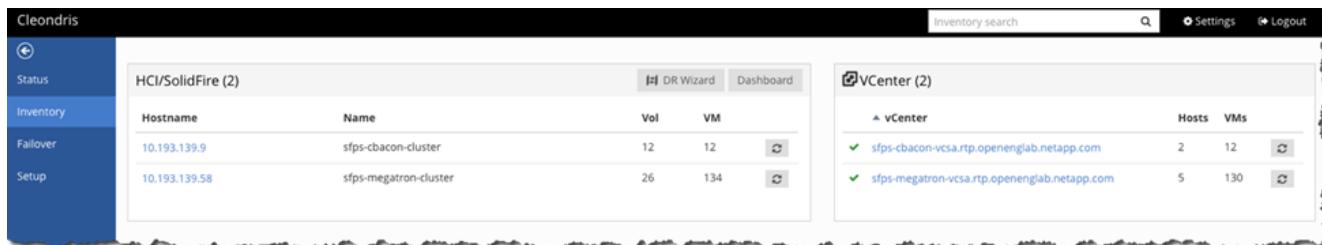
Password: .....  
Description: DR

✓ Credentials are OK!

Test Save Cancel



- When you have added your storage and vCenter Servers, change to the Inventory view so that you can see the results of your configuration.



Host	Name	Vol	VM
10.193.139.9	sfps-cbacon-cluster	12	12
10.193.139.58	sfps-megatron-cluster	26	134

vCenter	Hosts	VMs
sfps-cbacon-vcsa.rtp.openenglab.netapp.com	2	12
sfps-megatron-vcsa.rtp.openenglab.netapp.com	5	130

Here you can see the number of objects, which is a good way to confirm that things are working.

### Replication

You can use HCC to enable replication between your two sites. This allows us to stay in the HCC UI and decide what volumes to replicate.

**Important:** If a replicated volume contains VMs that are in two plans, only the first plan that fails over works because it will disable replication on that volume.

I recommend that each tier 1 application have its own volume. Tier 4 applications can all be on one volume, but there should be only one failover plan.

### Disaster Recovery Pairing: NetApp HCI DR with Cleondris

- Display the Failover page.
- On the diagram of your vCenter Servers and storage, select the Protection tab.

Cleondris

>Status  
Inventory  
Backup  
Restore  
**Failover**  
Setup

Overview   **Protection**   Plans   Activity

**VMware Virtual Machine Protection**

**vCenter**

[sfps-megatron-vcsa.rtp.openenglab.netapp.com](#)  
[sfps-primus-vcsa.rtp.openenglab.netapp.com](#)

**HCI Storage Protection**

**Cluster**

[sfps-megatron-cluster](#)  
[sfps-primus-cluster](#)

The far side of the screen displays some useful information, such as how many protected VMs you have. (In this example, none right now.) You can also access the Replication Wizard here.

**Protected Datastores    Protected VMs**

0/24                  0/133

0/5                  0/6

 Replication Wizard

**Protected Datastores    Protected VMs**

0/17                  0/137

0/3                  0/6

2004P6 - API-20200410-2157 - Copyright © Cleondris GmbH 2010-2020

This wizard makes the replication setup easy.

## HCI Replication Wizard

Source Volumes      Destination      vCenter      Preview

Select the cluster you want to protect:

Cluster: sfps-megatron-cluster

ID	Type	Name
<input type="checkbox"/> 1	Primary	<a href="#">NetApp-HCI-Datastore-01</a>
<input type="checkbox"/> 2	Primary	<a href="#">NetApp-HCI-Datastore-02</a>
<input type="checkbox"/> 3	Primary	<a href="#">NetApp-HCI-Select-Install</a>
<input type="checkbox"/> 4	Primary	<a href="#">NetApp-HCI-Select-Data-01</a>
<input type="checkbox"/> 5	Primary	<a href="#">NetApp-HCI-Select-Data-02</a>
<input type="checkbox"/> 6	Primary	<a href="#">NetApp-HCI-Select-Data-03</a>
<input type="checkbox"/> 7	Primary	<a href="#">NetApp-HCI-Select-Data-04</a>
<input type="checkbox"/> 8	Primary	<a href="#">INFRASTRUCTURE</a>
<input type="checkbox"/> 12	Primary	<a href="#">DESKTOP02</a>
<input checked="" type="checkbox"/> 15	Primary	<a href="#">DESKTOP03</a>
<input type="checkbox"/> 16	Primary	<a href="#">DESKTOP04</a>
<input type="checkbox"/> 569	Primary	<a href="#">workload-db-mongo-1</a>

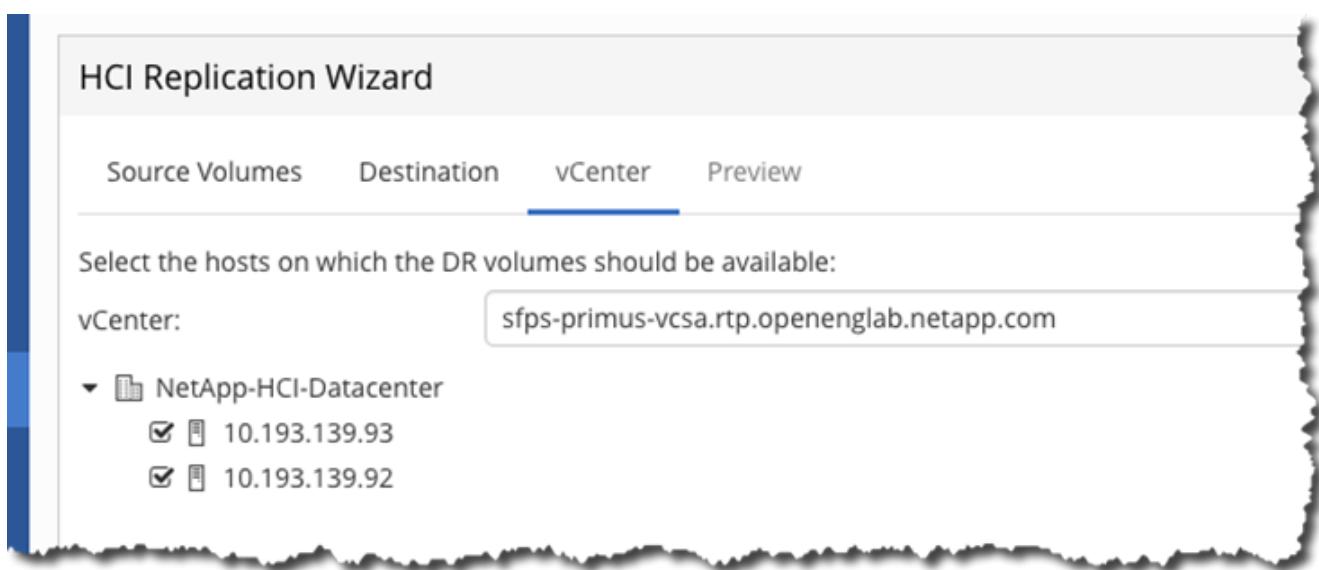
3. You can select the volumes that are important to you, but also make sure that you have the proper vCenter Server selected at the top in the cluster field.

At the far right, you see the pairing type, and only Sync is allowed or supported.

After you click Next, the destination area is displayed.



4. The default information is normally right, but it's still worth checking. Then click Next.



It is important to make sure that the disaster recovery site vCenter Server is displayed and that all hosts are selected. After that is complete, use the Preview button.

5. Next you see a summary. You can click Create DR to set the volume pairing and start replication.

Depending on your settings, replication might take a while. I suggest that you wait overnight.

## Recovery Planning: NetApp HCI DR with Cleondris

This section discusses successful failover of applications in a crisis or in a planned migration. It first looks at protecting complex multitier applications, and then simpler applications. You can build disaster recovery plans that are slow or fast, so this section provides examples of the highest-performing plans.

### Multitier Applications

1. From the Failover page, select the Plans tab.

The screenshot shows the Cleondris HCC interface. On the left, there is a sidebar with the following navigation options: Status, Inventory, Backup, Restore, Failover (which is highlighted in blue), and Setup. The main content area has tabs at the top: Overview, Protection, Plans (which is underlined in blue), and Activity. Below the tabs, the section title is "Failover Plans". Under "Name", it says "No failover plans have been defined".

2. On the far right is an +Add Failover Group button.

The screenshot shows the "Failover Plan Editor" dialog box. At the top, there is a "Plan Name:" input field and a checkbox for "Create temporary network when running in sandbox mode" which is checked. Below this is a table titled "Failover Groups" with columns: Prio, Name, VM Filter, Additional VMs, Delay, Unregister, and Wait for Tools. A note below the table says "No failover groups have been defined yet". To the right of the table is a "+ Add Failover Group" button. At the bottom of the dialog box are two sections: "Network Mapping" (with "Production Network" and "DR Network" fields) and "Storage Affinity" (with "Storage System" and "Hosts" fields). Both sections have notes: "No network mapping has been defined yet" and "No hosts are associated to specific storage cluster". At the very bottom right are "Save" and "Cancel" buttons.

In this example, we called this plan Multi-Tier. We will use the network mapping in the bottom left to change the virtual switch that is in use on production to the one in use on DR.

## Edit Network Mapping

Select the production and DR network you want to map to each other:

Production

vCenter sfps-megatron-vcsa.

Datacenter NetApp-HCI-Datacenter

- HCI\_Internal\_mNode\_Network
- HCI\_Internal\_OTS\_Network
- K8S-PG
- Desktops
- VM\_Network
- HCI\_Internal\_vCenter\_Network
- NetApp HCI Uplinks
- 10.193.138.0\_VL20
- vMotion
- Management Network

DR

vCenter sfps-primus-vcsa.rtp

Datacenter NetApp-HCI-Datacenter

- NetApp HCI Uplinks 01
- NetApp HCI VDS 01-HCI\_Internal\_Storage\_Network
- NetApp HCI VDS 01-HCI\_Internal\_mNode\_Network
- NetApp HCI VDS 01-Management Network
- NetApp HCI VDS 01-HCI\_Internal\_NKS\_Management
- NetApp HCI VDS 01-HCI\_Internal\_NKS\_Data
- TestNetwork
- NetApp HCI VDS 01-VM\_Network
- NetApp HCI VDS 01-vMotion
- NetApp HCI VDS 01-HCI\_Internal\_vCenter\_Network

» Map

Mappings

From	To
10.193.138.0_VL20	NetApp HCI VDS 01-VM_Network

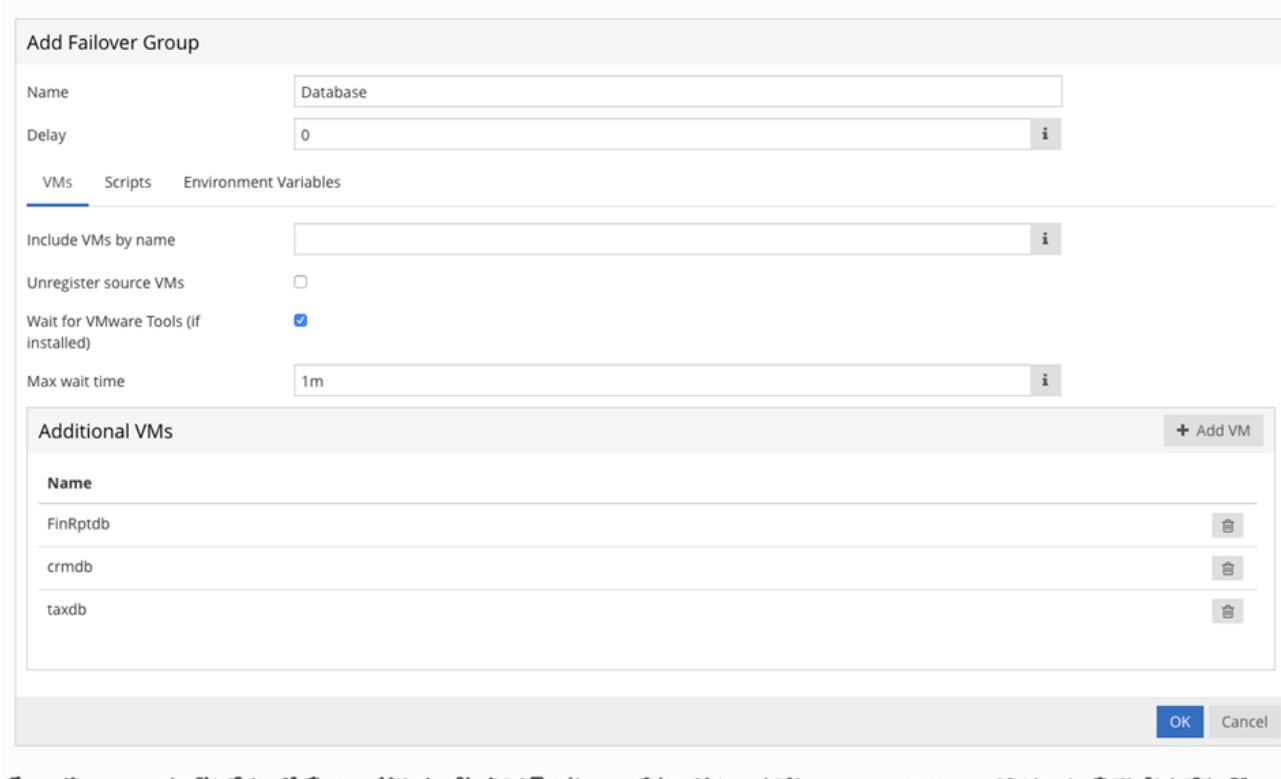
Delete

Save Cancel

The previous screenshot shows how you can choose the network switch in production and then in DR, use the Map button to select them, and then use Save. You can have more than one mapping if necessary.

3. To select the VMs to protect, click Add Failover Group.

Because this plan will protect multitier applications, the first group will be for databases.



Notice how this example enables Wait for VMware Tools. This setting is important, because it helps make sure that the applications are running. We used the Add VM button to add VMs that are databases. We didn't enable Unregister Source VMs, because it will slow down the failover. We now use the Add Failover button to protect the applications.

4. Do the same thing for web servers. When that is done, the screen resembles the following example.

**Failover Plan Editor**

Plan Name: MultiTier

Create temporary network when running in sandbox mode:

Prio	Name	VM Filter	Additional VMs	Delay	Unregister	Wait for Tools
1	Database		FinRptdb,crmdb,taxdb	0	✓	
2	Apps		FinRptA,crmA,taxA	0	✓	
3	Web		FinRptW,crmW,taxW	0	✓	

**Network Mapping**

Production Network	DR Network
10.193.138.0_VL20	NetApp HCI VDS 01-VM_Network

**Storage Affinity**

Storage System	Hosts
No hosts are associated to specific storage cluster	

**Save** **Cancel**

The important part of this plan is to get all the databases working; then the applications start, find the databases, and start working. Then the web servers start, and the applications are complete and working. This approach is the fastest way to set up this sort of recovery.

- Click Save before you continue.

#### Simple or Mass Applications to Fail Over

The order in which the VMs start is important, so that they work; that is what the previous section accomplished. Now we will fail over a set of VMs for which order is unimportant.

Let's create a new failover plan, with one failover group that has several VMs. We still need to do the network mapping.

The screenshot shows the Failover Plan Editor interface. At the top, it displays the plan name "Mass" and a checked checkbox for creating a temporary network when running in sandbox mode. Below this is a section titled "Failover Groups" with a table header "Prio Name VM Filter Additional VMs". A single row is listed with Prio 1, Name "VMs", VM Filter "mass01,mass02,mass03,mass04,mass06,mass05,mass07,mass08,mass09,mass10,mass11,mass12,mass13,mass14,mass15,mass16,mass17,mass18," and an empty "Additional VMs" field. To the right of the table is a "+ Add Failover Group" button. Below the table are two sections: "Network Mapping" and "Storage Affinity". The "Network Mapping" section contains tables for "Production Network" (IP 10.193.138.0\_VL20) and "DR Network" (NetApp HCI VDS 01-VM\_Network), with an "Edit Network Mapping" button. The "Storage Affinity" section contains tables for "Storage System" and "Hosts", both of which currently show "No hosts are associated to specific storage cluster". To the right of these sections are "+ Add Network Mapping" and "+ Add Storage Affinity" buttons. At the bottom right are "Save" and "Cancel" buttons.

Notice that there are several VMs in this plan. They will also start at different times, but that is OK because they are not related to each other.

### Planned Migration

Planned migration is similar to a disaster recovery failover, but because it is not a disaster recovery situation, it can be handled slightly differently. It is still good to practice the planned migration, but you can add something to your failover group: You can unregister the VM from the source. That takes a little more time, but in a planned migration that is not a bad thing.

A planned migration is usually a move to a new data center. Sometimes it is also used if destructive weather is approaching but has not yet arrived.

### Plan of Plans

With a plan of plans, you can trigger one plan and it will take care of all the failover plans.

The Plans tab contains a Plan of Plans section. You can use the "+Add Sub-Plan" to start a plan and add other plans to it.

The screenshot shows the "Create Plan of Plans" dialog. It has a "Plan of Plans Name" field containing "Master Plan" and a "+ Add Sub-Plan" button. Below this is a "Sub-Plan Name" section with two entries: "Mass" and "MultiTier", each with up and down arrow buttons for reordering. At the bottom right are "Save" and "Cancel" buttons.

In this example, the plan of plans is called Master Plan, and we added the two plans to it. Now when we execute a failover, or test failover, we will have the option for the Master Plan too.

This approach is good because it is best to test your application failovers in their own plan. Each plan is much easier to troubleshoot and fix, and when it is working well, you add it to your master plan.

### Script Support

You can use scripts as part of a test failover or for a wide range of other purposes. Uses include the following:

- Turning on anti-spam hardware
- Turning on security hardware
- Populating signage
- Updating IPAM hardware
- Changing the language settings in a database

If you edit your plan and then edit your failover group, you will see entries under Scripts.

The screenshot shows the 'Edit Failover Group' dialog box. At the top, there are fields for 'Name' (set to 'VMs') and 'Delay' (set to '0'). Below these are three tabs: 'VMs' (selected), 'Scripts' (highlighted with a blue underline), and 'Environment Variables'. The 'Scripts' tab displays a table with one row. The columns are 'Host Type', 'Host', 'Position', 'Script Type', and 'Script'. The data in the table is:

Host Type	Host	Position	Script Type	Script
Virtual Machine	ABC01	POST	POWERSHELL	\$A = Get-Date; Add-Content c:\utils\test_log.txt "\$A - test recording date."

In the following screenshot, the word Host refers to the VM that executes scripts. Click the edit button to see the Edit Script window:

## Edit Script

Host Type	VM
VM Name	ABC01
User	administrator
Password	<a href="#">Change Password</a>
Group Order	POST
Type	PowerShell
Script	<pre>\$A = Get-Date; Add-Content c:\utils\test_log.txt "\$A - test recording date. "</pre>

**OK** **Cancel**

You should make sure to test your script before you copy and paste it into this dialog box. You should also select Post in the Group Order field. Make sure to use the right credentials.

If you follow the execution, the following screenshot indicates that the script ran successfully.

Waiting for guest tools on VM ABC01

Executing POST script in failover group 'VMs' on VM ABC01

The script execution completed with exit code 0

If the exit code is anything other than 0, then the script was not successful.

## Script Troubleshooting

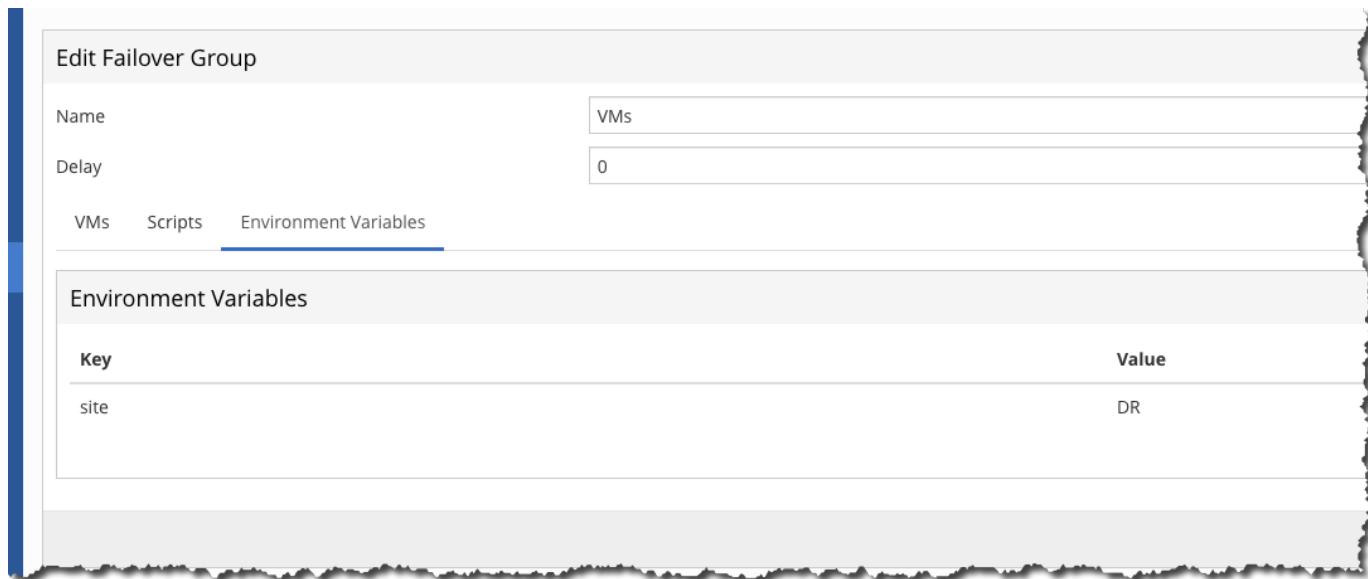
If a script does not execute properly, then check the following issues:

- VMware Tools only let one external process run at a time. Therefore, if VMware Tools is updating itself, then the script will not execute. This can occur if you set your VMs to automatically upgrade VMware Tools. This is done in VM settings > VM Options > VMware Tools.
- Check for credentials issues.
- Check for script issues, such as a prompt or other functionality that requires human input.

It is a best practice to run simple scripts that only perform essential tasks. You might also want to include a log file for troubleshooting purposes.

## Environment Variables

Environmental variables allow a running script to pull information from the environment whether the script is running at the production site or a DR site. Environment variables can be entered in Edit Failover Group dialog box. You can first edit your plan and then edit your failover group.



Note that these environment variables are not in the environment that we normally think of, and you cannot use the `Set` command to see them. To see the full list of variables, run the script from the following screenshot. This script contains `Get-Variable * > c:\utils\var_log.txt` to capture all variables.

## Edit Script

Host Type	VM
VM Name	ABC01
User	administrator
Password	Change Password
Group Order	POST
Type	PowerShell
Script	<pre>\$A = Get-Date; Get-Variable * &gt; c:\utils\var_log.txt Add-Content c:\utils\test_log.txt "\$A - test recording date."</pre>

**OK** **Cancel**

This lists the 50+ variables available plus any variable that you have added, which are seen at the end of the list.

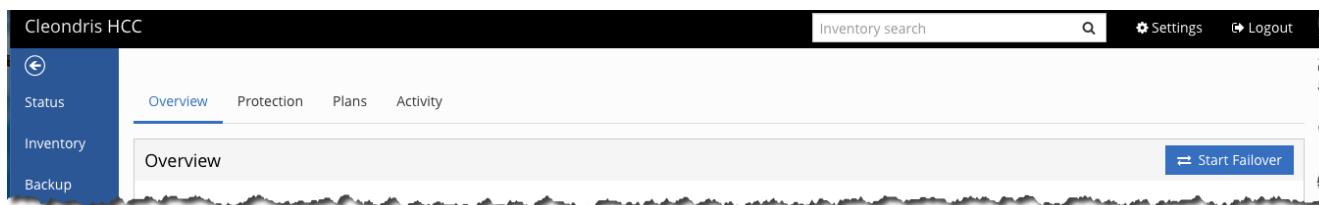
### **Failover: NetApp HCI DR with Cleondris**

#### **Test Failover**

A test failover is important, because it proves to you, your application owner, your manager, and the BCDR people that your disaster recovery plan works.

To test failover, complete the following steps:

1. From the Failover page, click Start Failover.



2. On the Failover page, you have some choices to make.

Failover Plan	MultiTier
Source HCI Cluster	sfps-megatron-cluster
Destination HCI Cluster	sfps-primus-cluster
Destination vCenter	sfps-megatron-vcsa.rtp.openenglab.netapp.com
Destination Datacenter	NetApp-HCI-Datacenter

Preview    Cancel

Carefully specify the plan, where the VMs came from, and where they are going to be recovered.

From: sfps-megatron-cluster      To: sfps-primus-cluster      ⚠ 3 VMs not included in this plan will lose protection

Plan	Priority	Name	Datastore	Source Volume	Destination Volume	Current vCenter	Destination vCenter
MultiTier	1	taxdb	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	1	crmdb	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	1	FinRptdb	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	2	crmA	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	2	FinRptA	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	2	taxA	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	3	taxW	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	3	crmW	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng
MultiTier	3	FinRptW	DESKTOP03	DESKTOP03 ID: 15	DESKTOP03 ID: 138	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-megatron-vcsa.rtp.openeng

Failover to Sandbox
Start
Cancel

The screen displays a list of the VMs that are in the plan. In this example, a warning at the top right says that three VMs are not included. That means there are three VMs we did not make part of the plan in the replicated volume.

If you see a red X in the first column on the left, you can click it and learn what the problem is.

- At the bottom right of the screen, you must choose whether to test the failover (Failover to Sandbox) or start a real failover. In this example, we select Failover to Sandbox.

Cleondris HCC

Inventory search Settings Logout

Status Inventory Backup Restore **Failover** Setup

Overview Protection Plans **Activity**

**Failover Plan Execution** Show Historical

ID	Description	User	Plan	Date	Status
2	Sandbox failover using plan Mass	admin	Mass	2020-04-14 13:21	Running

8.0.2004P6 - API-20200410-2157 - Copyright © Cleondris GmbH 2010-2020

- A summary now lists plans in action. For more information, use the magnifying glass in the far left (described in “Monitoring,” later in this document).

## Running Failover

At first, the failover is the same as the test failover. But the procedure changes when you arrive at the point shown here:

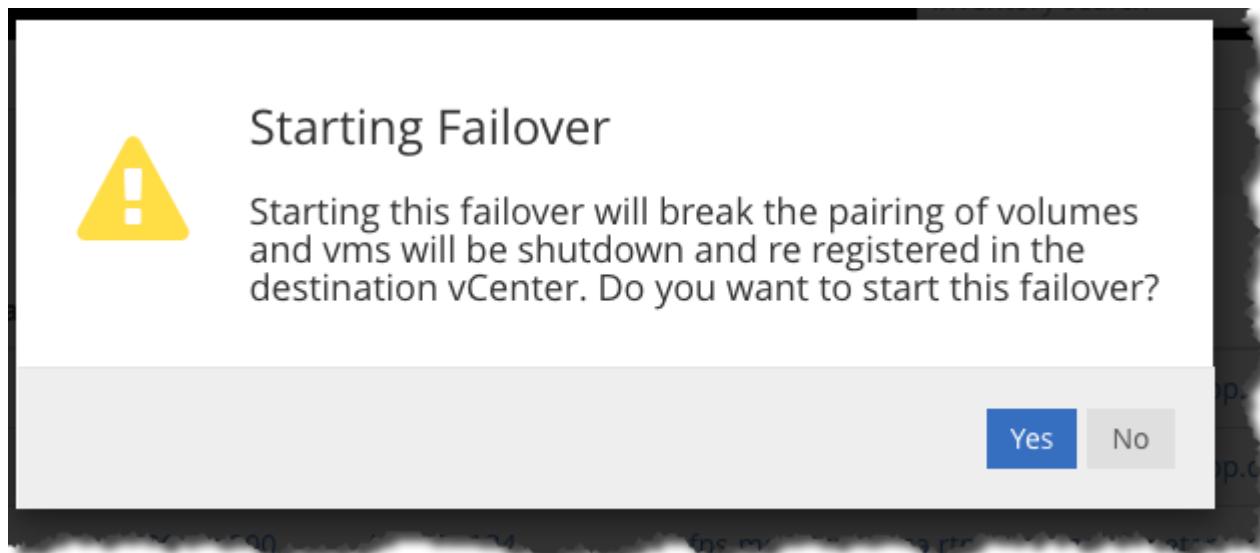
- Instead of selecting the Failover to Sandbox option, select Start.

The screenshot shows the 'Failover Preview' interface. On the left sidebar, 'Failover' is selected. The main area displays a table with columns: Plan, Priority, Name, Datastore, Source Volume, Destination Volume, Current vCenter, and Destination vCenter. Three rows are listed:

Plan	Priority	Name	Datastore	Source Volume	Destination Volume	Current vCenter	Destination vCenter
ABC	1	ABC01	ABC	ABC ID: 800	ABC ID: 134	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-primus-vcsa.rtp.openenglab.netapp.com
ABC	1	ABC03	ABC	ABC ID: 800	ABC ID: 134	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-primus-vcsa.rtp.openenglab.netapp.com
ABC	1	ABC02	ABC	ABC ID: 800	ABC ID: 134	sfps-megatron-vcsa.rtp.openenglab.netapp.com	sfps-primus-vcsa.rtp.openenglab.netapp.com

At the bottom right are buttons for 'Failover to Sandbox', 'Start', and 'Cancel'.

2. Select Yes.



3. The screen shows that this is a failover, and it is running. For more information, use the magnifying glass (discussed in the "Monitoring" section).

The screenshot shows the 'Failover Plan Execution' screen. On the left sidebar, 'Failover' is selected. The main area has tabs: Overview, Protection, Plans, and Activity. The 'Activity' tab is selected. It shows a table for 'Failover Plan Execution' with columns: Id, Description, User, Plan, Date, and Status. One row is listed:

Id	Description	User	Plan	Date	Status
4	Failover using plan ABC	admin	ABC	2020-04-15 08:25	Running

A 'Show Historical' button is located at the top right of the table area.

#### Monitoring During a Failover

1. When a failover or a test failover is running, you can monitor it by using the magnifying glass at the far right.

The screenshot shows the Cleondris HCC web interface. On the left, a vertical navigation bar has items: Status, Inventory, Backup, Restore, **Failover**, and Setup. The 'Failover' item is highlighted with a blue background. The main content area has a header 'Failover Plan Execution' with a 'Show Historical' button. Below is a table with columns: Id, Description, User, Plan, Date, and Status. One row is shown: Id 2, Description 'Sandbox failover using plan Mass', User 'admin', Plan 'Mass', Date '2020-04-14 13:21', Status 'Running'. At the bottom right of the main area, there's a small note: '8.0.2004P6 - API-20200410-2157 - Copyright © Cleondris GmbH 2010-2020'.

2. Click the magnifying glass to see much more detail.

This screenshot shows a detailed view of a failover ticket. The left sidebar shows the 'Failover' menu item is selected. The main panel displays ticket information: Ticket: Sandbox failover using plan Mass, Started: 2020-04-14 12:30, Status: Finished. Below this are tabs for Log, Details, and a Download Report button. The Log tab is active, showing a table with columns Date, Type, and Message. Three log entries are listed: 2020-04-14 12:30 Log "Starting failover activity using plan 'Mass'", 2020-04-14 12:30 Log "Sandbox mode = on", and 2020-04-14 12:30 Log "Creating clone of volume 137".

3. As the failover or test failover progresses, a VM Screenshots option appears.

This screenshot shows the same failover ticket as before, but now the 'VM Screenshots' tab is visible under the Log, Details, and VM Screenshots tabs. The ticket details remain the same: Ticket: Sandbox failover using plan Mass, Started: 2020-04-14 13:29, Status: WaitRelease. The Log table shows the same three log entries. The VM Screenshots tab is present but no screenshots are displayed.

Sometimes it is useful to see the screenshots to confirm that the VM is running. It is not logged in, so you cannot tell if the applications are running, but at least you know that the VM is.

## Looking at History When No Failover Is Running

To view past tests or failovers, click the Show Historical button on the Activity tab. Use the magnifying glass for more detail.

Cleondris HCC

Inventory search  Settings Logout

Status Overview Protection Plans **Activity**

Failover Plan Execution

Show Historical

ID	Description	User	Plan	Date	Status	More
2	Sandbox failover using plan Mass	admin	Mass	2020-04-14 13:21	Running	

8.0.2004P6 - API-20200410-2157 - Copyright © Cleondris GmbH 2010-2020

Cleondris HCC

Inventory search  Settings Logout

Status Overview Protection Plans **Activity**

Failover Plan Execution

Hide Historical

ID	Description	User	Plan	Date	Status	More
2	Sandbox failover using plan Mass	admin	Mass	2020-04-14 13:21	Running	
1	Sandbox failover using plan Mass	admin	Mass	2020-04-14 12:30	Finished	

8.0.2004P6 - API-20200410-2157 - Copyright © Cleondris GmbH 2010-2020

You can also download a report with the details.

Cleondris HCC

Inventory search  Settings Logout

Status Overview Protection Plans **Activity**

Ticket: Sandbox failover using plan Mass  
Started: 2020-04-14 12:30  
Status: Finished

**Download Report**

Log Details

Date	Type	Message
2020-04-14 12:30	Log	Starting failover activity using plan 'Mass'
2020-04-14 12:30	Log	Sandbox mode = on
2020-04-14 12:30	Log	Creating clone of volume 137

These reports have various uses: for example, to prove to an application owner that you tested the failover of that application. Also, the report can provide details that might help you troubleshoot a failed failover.

You can add text to a report by adding the text to the plan in the comment field.

Failover Plan Editor	
Plan Name:	ABC
Comment (Added to the report)	App expert is Joe Smith.
Create temporary network when running in sandbox mode:	<input type="checkbox"/>
Network to use for sandbox mode	TestNetwork

## Best Practices: NetApp HCI DR with Cleondris

### Recommendations for Success

The following tips can help you be more successful with your BCDR work.

### Applications

Know your applications and what makes them work. The more time you spend on them, the more successful you will be with your real and test failovers. When there are issues, you will be able to solve them faster.

Protect one application first. Choose a relatively simple one, and demo the test failover to your peers and management. The demonstration will help you with management and peer support, and the test will help you learn more before you protect other applications.

Your tier 1 applications should be on their own volume.

### Practice

You need to practice often in as realistic a scenario as possible. For example, practice off-site, sometimes with poor internet in a hotel conference room. Practicing often is key, and try changing the teams around so that application team X is recovering application Y; this approach will help with knowledge sharing.

### Executive Sponsor

Make sure to have an executive sponsor. You'll need executive support when teams are not working well together, or when you need application teams to be reasonable about recovery time.

### Plan for Partial or Full Outage

Most disaster recovery events are partial ones, so make sure your tier 1 applications can be recovered without having to recover everything.

### Trigger Time

Practice the failovers, but also practice managing others who are authorized to trigger a failover. They need to practice, and they need to know how successful or unsuccessful the failovers are. Make sure they practice with you in as realistic a scenario as possible. You can do a sand-table-type exercise in which operations people bring up issues and managers discuss their response.

## Why Does Disaster Recovery Fail?

There are several possible reasons for a disaster recovery plan failing:

- BCDR is needed.
- Attitude is missing: People do not care as much as they should.
- The executive sponsor is missing or not assigned.
- There isn't enough practice, or it isn't real enough.
- Data from the test gets into the product. This situation is serious and must be avoided.

## Additional Uses for Disaster Recovery Orchestration Tools

Over time, customers have found other uses for disaster recovery orchestration tools. For example, they test application and OS upgrades in a test failover. This testing is better than testing in a lab, because it uses the actual production bits—which means that, when done in production, the process will be as smooth as in the test failover. I have also seen security vulnerability testing done as a test failover first to determine what applications might be negatively affected.

### Active-Active Site

Currently, to protect an active-active site, you must install HCC on both sites and protect as normal. There is currently no overview of the protection. Active-active is the best model, because you can split your applications over two sites; when there is an outage, you only need to fail over half.

### Allowing Extra Resources in Test Failover

Sometimes it is necessary to have more resources in the test failover so that a proper application test can occur. For example, these resources might consist of things like physical anti-spam appliances or load balancers. You can also include things like databases, which has the potential to cause problems, because you must make sure test data does not get into production. To perform this process reliably, use the following steps as a guideline.

1. A script executes in the disaster recovery test process (or use a manual process if necessary).
2. A separate logical partition (LPAR) is created.
3. A virtual network is added to the separate LPAR, and it is already connected to the test network.
4. A script exports and copies the appropriate data to the separate and new LPAR. It's likely that you'll need to have the application on the separate partition, too.
5. You might need to tweak DNS names or the configuration of the application in the test network to access this new server.
6. The test completes successfully.
7. After the test is done, and the cleanup occurs, another script runs, and it deletes the separate partition. That step keeps anything from getting into production accidentally.

You can use a similar process to get a domain controller into a test failover:

1. Power off the domain controller in the disaster recovery site. Make sure there is another domain controller still running.
2. After the domain controller is off, clone it.
3. Power on the original domain controller.

4. Put the cloned domain controller on the test network.
5. Power on the clone domain controller.
6. You should be able to use the domain controller in the test now, whether for authentication or DNS.
7. When the test is done, delete the cloned domain controller. Don't skip this step, because you don't want that domain database talking to the production domain.

It's best to script these steps and execute the script from the recovery plan. However, to do that, you need a script or batch file that can tell whether it is executing in test or real failover—and in real failover, it does nothing.

## Syslog

It is useful to capture events from Cleondris by using syslog. Groups such as security or operations might benefit.

1. To do this, use the Setup page and the Events tab. Then use the Add Receiver button.

**Edit Event Receiver**

Type	SYSLOG
Hostname	10.193.136.33
Message Format	DEFAULT
Filter	*

**Send Test Event** **Save** **Cancel**

1. Specify which event to send. In this example, the best idea might be to send all of them for now. Select the boxes; some do not apply to Cleondris HCC and BCDR, but they will not be generated if not used.

You can see the BCDR events in the Events section at the bottom of the list.

CDM-09670	Default	User creates BCDR plan	User %(u) creates BCDR plan %(s)
CDM-09671	Default	User updates BCDR plan	User %(u) updates BCDR plan %(s)
CDM-09672	Default	User deletes BCDR plan	User %(u) deletes BCDR plan %(s)
CDM-09680	Default	User executes BCDR plan	User %(u) executes BCDR plan %(s)
CDM-09681	Default	User tests BCDR plan	User %(u) tests BCDR plan %(s)

## VM State

The VM state is preserved during a failover. A VM that is powered on or off in production remains in the same state after a failover or during a test failover. However, be aware that HCC scans vCenter every 20 minutes. Therefore, you need to wait for that scan or use the refresh button in HCC to immediately refresh.

	Hosts	VMs	Refresh
sfps-megatron-vcsa.rtp.openenglab.netapp.com	5	133	
sfps-primus-vcsa.rtp.openenglab.netapp.com	2	6	

## Add an Execute-Only Account

An execute-only account can be useful for a manager to trigger a failover without saving the changes. You create this account yourself.

First, create a role that has the following privileges:

- Login
- Inventory\_sf\_view
- Inventory\_vc\_view
- Restore\_exec\_sf\_failover
- Failover\_view
- Failover\_job\_modify
- Failover\_config\_view

When the role is done, create a user with that role; the resulting account is an execute-only account. This set of privileges lets the user look at and change things but not save the changes.

## Idle Time Out

This parameter can be set to perform an automatic log out when there is no activity in the browser. Working on a different tab counts as activity.

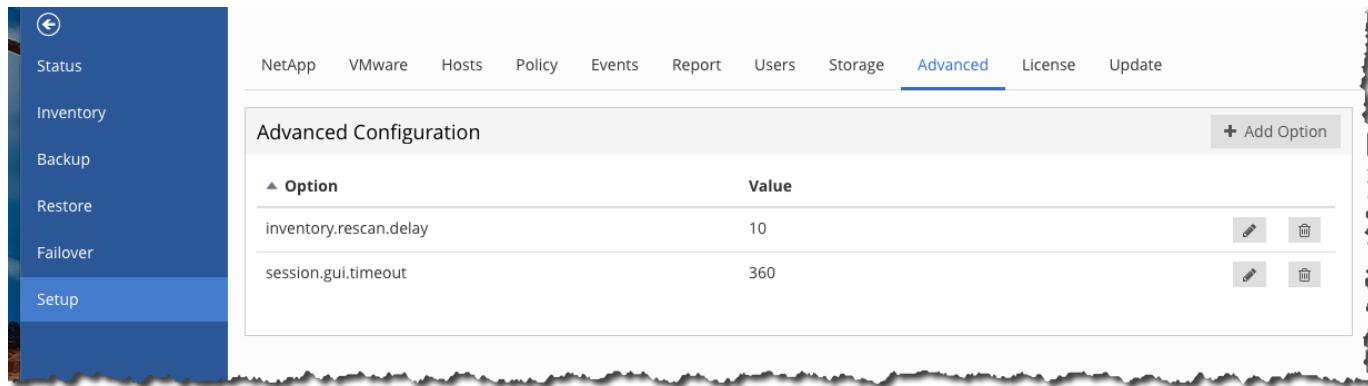
Select the Setup option and then select the Advanced tab to see the Advanced Configuration window.

Option	Value	Actions
session.gui.timeout	360	

Click the Add Option button to add the option and value. In the screenshot above, 360 seconds must pass before a timeout if there is no activity in the browser.

## Inventory Rescan

The inventory rescan setting is used when a VM state is not preserved when it should be. For example, a VM should not be powered on in a failover if it is off in production. The value for the rescan interval can be set between 5 minutes and 1440 minutes; it is set to 20 minutes by default.



The screenshot shows the 'Advanced' tab selected in the navigation bar. Under 'Advanced Configuration', there is a table with two rows:

Option	Value
inventory.rescan.delay	10
session.gui.timeout	360

A '+ Add Option' button is located at the top right of the configuration table.

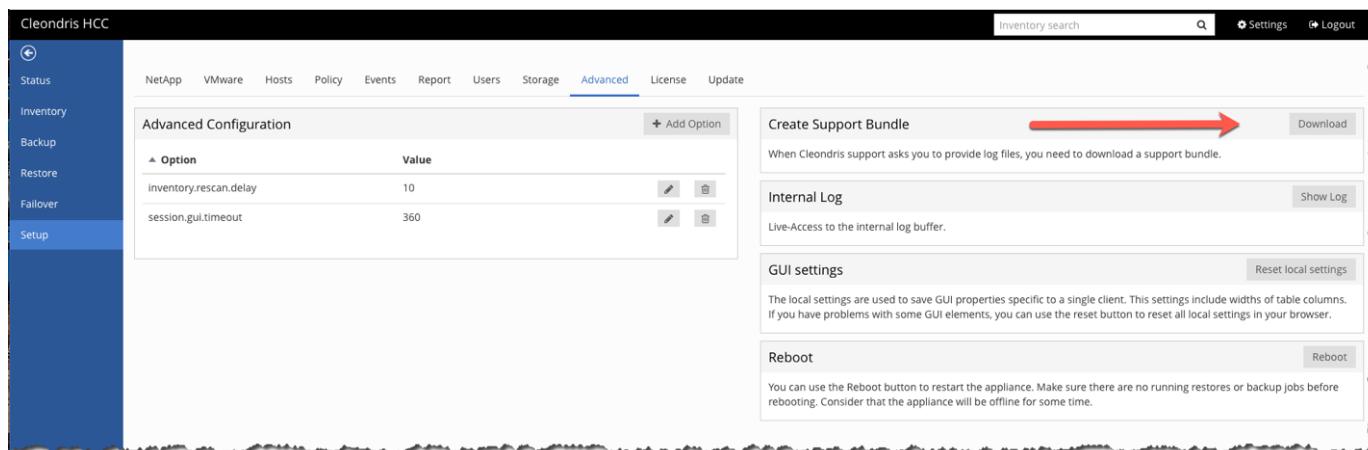
In the previous screenshot, the interval is set for 10 minutes.

Be aware that this setting changes the vCenter rescan time and also the Solidfire rescan time.

## General Support

The following best practices can improve your experience with Cleondris and assist with support.

- Always include a support bundle when you ask for support.



The screenshot shows the 'Advanced' tab selected in the navigation bar. Under 'Advanced Configuration', there is a table with two rows:

Option	Value
inventory.rescan.delay	10
session.gui.timeout	360

Below the table, there is a 'Create Support Bundle' section with a 'Download' button. A large red arrow points to the 'Download' button. To the right of the download button is a 'Show Log' button. Further down are sections for 'Internal Log' (with a 'Show Log' button) and 'GUI settings' (with a 'Reset local settings' button). At the bottom is a 'Reboot' section with a 'Reboot' button.

- With certain edge cases, additional logging is very helpful for support. Enable the additional logging, and then perform the action that you are having trouble with again. You can then delete `log.level` because you do not want to routinely debug this level.

Advanced Configuration		+ Add Option
▲ Option	Value	
inventory.rescan.delay	10	 
log.level	debug	 
session.gui.timeout	360	 

- A busy vCenter Server Appliance (VCSA) can cause issues under some conditions. To minimize this problem, add more memory to the VCSCA.
- Issues can also be caused by the fact that one or two VMs might not be cleaned up in a test failover. You can clean these VMs up with the following steps:
  - Power off the VMs. This may take some time.
  - Remove the VMs from inventory.
 Often, these two steps allow the datastore to disappear. You can then perform a Rescan Storage operation.

#### Where to Find Additional Information: NetApp HCI DR with Cleondris

To learn more about the information that is described in this document, review the following websites:

- NetApp HCI Documentation Center  
<https://docs.netapp.com/hci/index.jsp>
- NetApp HCI Documentation Resources page  
<https://www.netapp.com/us/documentation/hci.aspx>
- NetApp Product Documentation  
<https://www.netapp.com/us/documentation/index.aspx>
- Cleondris HCC product page  
<https://www.cleondris.com/en/hci-control-center.xhtml>
- Cleondris Support Portal  
<https://support.cleondris.com/>

## Security

# Infrastructure

## NVA-1148: NetApp HCI with Red Hat Virtualization

Alan Cowles, Nikhil M Kulkarni, NetApp

NetApp HCI with Red Hat Virtualization is a verified, best-practice architecture for the deployment of an on-premises virtual datacenter environment in a reliable and dependable manner.

This architecture reference document serves as both a design guide and a deployment validation of the Red Hat Virtualization solution on NetApp HCI. The architecture described in this document has been validated by subject matter experts at NetApp and Red Hat to provide a best-practice implementation for an enterprise virtual datacenter deployment using Red Hat Virtualization on NetApp HCI within your own enterprise datacenter environment.

### Use Cases

The NetApp HCI for Red Hat OpenShift on Red Hat Virtualization solution is architected to deliver exceptional value for customers with the following use cases:

1. Infrastructure to scale on demand with NetApp HCI
2. Enterprise virtualized workloads in Red Hat Virtualization

### Value Proposition and Differentiation of NetApp HCI with Red Hat Virtualization

NetApp HCI provides the following advantages with this virtual infrastructure solution:

- A disaggregated architecture that allows for independent scaling of compute and storage.
- The elimination of virtualization licensing costs and a performance tax on independent NetApp HCI storage nodes.
- NetApp Element storage provides quality of service (QoS) per storage volume and allows for guaranteed storage performance for workloads on NetApp HCI, preventing adjacent workloads from negatively affecting performance.
- The data fabric powered by NetApp allows data to be replicated from an on-premise to on-premise location or replicated to the cloud to move the data closer to where the application needs the data.
- Support through NetApp Support or Red Hat Support.

### NetApp HCI Design

NetApp HCI, is the industry's first and leading disaggregated hybrid cloud infrastructure, providing the widely recognized benefits of hyperconverged solutions. Benefits include lower TCO and ease of acquisition, deployment, and management for virtualized workloads, while also allowing enterprise customers to independently scale compute and storage resources as needed. NetApp HCI with Red Hat Virtualization provides an open source, enterprise virtualization environment based on Red Hat Enterprise Linux.

By providing an agile turnkey infrastructure platform, NetApp HCI enables you to run enterprise-class virtualized and containerized workloads in an accelerated manner. At its core, NetApp HCI is designed to provide predictable performance, linear scalability of both compute and storage resources, and a simple deployment and management experience.

## **Predictable**

One of the biggest challenges in a multitenant environment is delivering consistent, predictable performance for all your workloads. Running multiple enterprise-grade workloads can result in resource contention, where one workload interferes with the performance of another. NetApp HCI alleviates this concern with storage quality-of-service (QoS) limits that are available natively with NetApp Element software. Element enables the granular control of every application and volume, helps to eliminate noisy neighbors, and satisfies enterprise performance SLAs. NetApp HCI multitenancy capabilities can help eliminate many traditional performance-related problems.

## **Flexible**

Previous generations of hyperconverged infrastructure typically required fixed resource ratios, limiting deployments to four-node and eight-node configurations. NetApp HCI is a disaggregated hyper-converged infrastructure that can scale compute and storage resources independently. Independent scaling prevents costly and inefficient overprovisioning, eliminates the 10% to 30% HCI tax from controller virtual machine (VM) overhead, and simplifies capacity and performance planning. NetApp HCI is available in mix-and-match, small, medium, and large storage and compute configurations.

The architectural design choices offered enable you to confidently scale on your terms, making HCI viable for core Tier-1 data center applications and platforms. NetApp HCI is architected in building blocks at either the chassis or the node level. Each chassis can hold four nodes in a mixed configuration of storage or compute nodes.

## **Simple**

A driving imperative within the IT community is to simplify deployment and automate routine tasks, eliminating the risk of user error while freeing up resources to focus on more interesting, higher-value projects. NetApp HCI can help your IT department become more agile and responsive by both simplifying deployment and ongoing management.

## **Business Value**

Enterprises that perform virtualization in an open-source data center with Red Hat products can realize the value of this solution by following the recommended design, deployment, and best practices described in this document. The detailed setup of RHV on NetApp HCI provides several benefits when deployed as part of an enterprise virtualization solution:

- High availability at all layers of the stack
- Thoroughly documented deployment procedures
- Nondisruptive operations and upgrades to hypervisors and the manager VM
- API-driven, programmable infrastructure to facilitate management
- Multitenancy with performance guarantees
- The ability to run virtualized workloads based on KVM with enterprise-grade features and support
- The ability to scale infrastructure independently based on workload demands

NetApp HCI with Red Hat Virtualization acknowledges these challenges and helps address each concern by implementing a verified architecture for solution deployment.

## **Technology Overview**

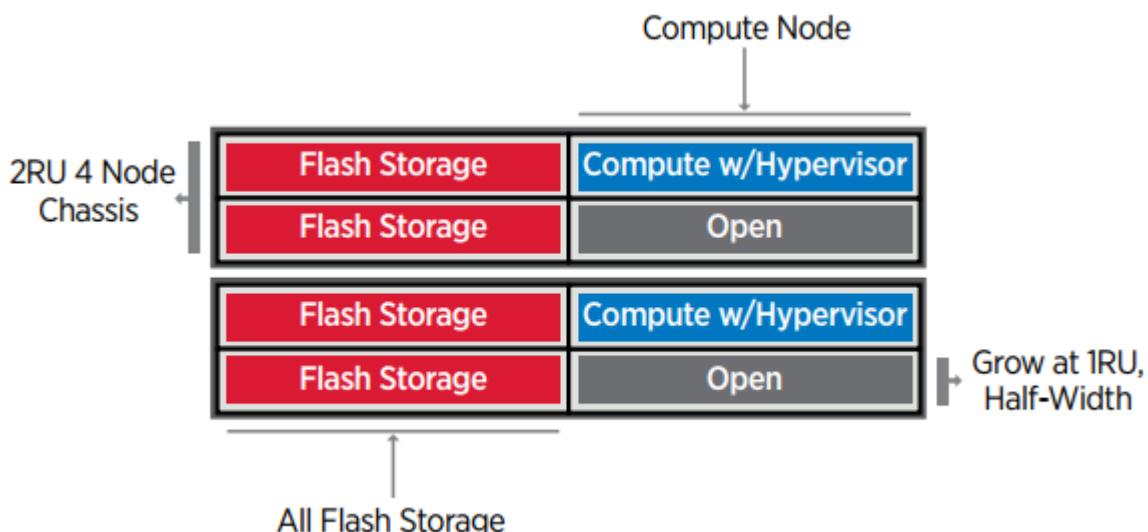
With NetApp HCI for Red Hat Virtualization, you can deploy a fully integrated, production-grade virtual data

center that allows you to take advantage of the following features:

- NetApp HCI compute and storage nodes
  - Enterprise-grade hyperconverged infrastructure designed for hybrid cloud workloads
  - NetApp Element storage software
  - Intel-based server compute nodes, including options for NVIDIA GPUs
- Red Hat Virtualization
  - Enterprise hypervisor solution for deployment and management of virtual infrastructures

### NetApp HCI

NetApp HCI is an enterprise-scale disaggregated hybrid cloud infrastructure (HCI) solution that delivers compute and storage resources in an agile, scalable, and easy-to-manage two-rack unit (2RU) four-node building block. It can also be configured with 1RU compute and server nodes. The minimum deployment consists of four NetApp HCI storage nodes and two NetApp HCI compute nodes. The compute nodes are installed as RHV-H hypervisors in an HA cluster. This minimum deployment can be easily scaled to fit customer enterprise workload demands by adding additional NetApp HCI storage or compute nodes to expand available resources.



The design for NetApp HCI for Red Hat Virtualization consists of the following components in a minimum starting configuration:

- NetApp H-Series all-flash storage nodes running NetApp Element software
- NetApp H-Series compute nodes running the Red Hat Virtualization RHV-H hypervisor

For more information about compute and storage nodes in NetApp HCI, see the [NetApp HCI Datasheet](#).

### NetApp Element Software

NetApp Element software provides modular, scalable performance, with each storage node delivering guaranteed capacity and throughput to the environment. You can also specify per-volume storage QoS policies to support dedicated performance levels for even the most demanding workloads.

## iSCSI Login Redirection and Self-Healing Capabilities

NetApp Element software uses the iSCSI storage protocol, a standard way to encapsulate SCSI commands on a traditional TCP/IP network. When SCSI standards change or when Ethernet network performance improves, the iSCSI storage protocol benefits without the need for any changes.

Although all storage nodes have a management IP and a storage IP, NetApp Element software advertises a single storage virtual IP address (SVIP address) for all storage traffic in the cluster. As a part of the iSCSI login process, storage can respond that the target volume has been moved to a different address, and therefore it cannot proceed with the negotiation process. The host then reissues the login request to the new address in a process that requires no host-side reconfiguration. This process is known as iSCSI login redirection.

iSCSI login redirection is a key part of the NetApp Element software cluster. When a host login request is received, the node decides which member of the cluster should handle the traffic based on IOPS and the capacity requirements for the volume. Volumes are distributed across the NetApp Element software cluster and are redistributed if a single node is handling too much traffic for its volumes or if a new node is added. Multiple copies of a given volume are allocated across the array. In this manner, if a node failure is followed by volume redistribution, there is no effect on host connectivity beyond a logout and login with redirection to the new location. With iSCSI login redirection, a NetApp Element software cluster is a self-healing, scale-out architecture that is capable of non-disruptive upgrades and operations.

## NetApp Element Software Cluster QoS

A NetApp Element software cluster allows QoS to be dynamically configured on a per-volume basis. You can use per-volume QoS settings to control storage performance based on SLAs that you define. The following three configurable parameters define the QoS:

- **Minimum IOPS.** The minimum number of sustained IOPS that the NetApp Element software cluster provides to a volume. The minimum IOPS configured for a volume is the guaranteed level of performance for a volume. Per-volume performance does not drop below this level.
- **Maximum IOPS.** The maximum number of sustained IOPS that the NetApp Element software cluster provides to a specific volume.
- **Burst IOPS.** The maximum number of IOPS allowed in a short burst scenario. The burst duration setting is configurable, with a default of 1 minute. If a volume has been running below the maximum IOPS level, burst credits are accumulated. When performance levels become very high and are pushed, short bursts of IOPS beyond the maximum IOPS are allowed on the volume.

## Multitenancy

Secure multitenancy is achieved with the following features:

- **Secure authentication.** The Challenge-Handshake Authentication Protocol (CHAP) is used for secure volume access. The Lightweight Directory Access Protocol (LDAP) is used for secure access to the cluster for management and reporting.
- **Volume access groups (VAGs).** Optionally, VAGs can be used in lieu of authentication, mapping any number of iSCSI initiator-specific iSCSI Qualified Names (IQNs) to one or more volumes. To access a volume in a VAG, the initiator's IQN must be in the allowed IQN list for the group of volumes.
- **Tenant virtual LANs (VLANs).** At the network level, end-to-end network security between iSCSI initiators and the NetApp Element software cluster is facilitated by using VLANs. For any VLAN that is created to isolate a workload or a tenant, Element software creates a separate iSCSI target SVIP address that is accessible only through the specific VLAN.
- **VPN routing/forwarding (VRF)-enabled VLANs.** To further support security and scalability in the data center, Element software allows you to enable any tenant VLAN for VRF-like functionality. This feature

adds these two key capabilities:

- **L3 routing to a tenant SVIP address.** This feature allows you to situate iSCSI initiators on a separate network or VLAN from that of the NetApp Element software cluster.
- **Overlapping or duplicate IP subnets.** This feature enables you to add a template to tenant environments, allowing each respective tenant VLAN to be assigned IP addresses from the same IP subnet. This capability can be useful for service provider environments where scale and preservation of IP-space are important.

## Enterprise Storage Efficiencies

The NetApp Element software cluster increases overall storage efficiency and performance. The following features are performed inline, are always on, and require no manual configuration by the user:

- **Deduplication.** The system only stores unique 4K blocks. Any duplicate 4K blocks are automatically associated with an already stored version of the data. Data is on block drives and is mirrored with Element Helix data protection. This system significantly reduces capacity consumption and write operations within the system.
- **Compression.** Compression is performed inline before data is written to NVRAM. Data is compressed, stored in 4K blocks, and remains compressed in the system. This compression significantly reduces capacity consumption, write operations, and bandwidth consumption across the cluster.
- **Thin provisioning.** This capability provides the right amount of storage at the time that you need it, eliminating capacity consumption that caused by overprovisioned volumes or underutilized volumes.
- **Helix.** The metadata for an individual volume is stored on a metadata drive and is replicated to a secondary metadata drive for redundancy.



Element was designed for automation. All the storage features mentioned above can be managed with APIs. These APIs are the only method that the UI uses to control the system and can be incorporated into user workflows to ease the management of the solution.

## Red Hat Virtualization

Red Hat Virtualization (RHV) is an enterprise virtual data center platform that runs on Red Hat Enterprise Linux using the KVM hypervisor.

For more information about Red Hat Virtualization, see the website located [here](#).

RHV provides the following features:

- **Centralized management of VMs and hosts.** The RHV manager runs as a physical or VM in the deployment and provides a web-based GUI for the management of the solution from a central interface.
- **Self-Hosted Engine.** To minimize the hardware requirements, RHV allows RHV Manager to be deployed as a VM on the same hosts that run guest VMs.
- **High Availability.** To avoid disruption from host failures, RHV allows VMs to be configured for high availability. The highly available VMs are controlled at the cluster level using resiliency policies.
- **High Scalability.** A single RHV cluster can have up to 200 hypervisor hosts, enabling it to support the requirements of massive VMs to hold resource-greedy enterprise-class workloads.
- **Enhanced security.** Inherited from RHEL, Secure Virtualization (sVirt) and Security Enhanced Linux (SELinux) technologies are employed by RHV for the purposes of elevated security and hardening for the hosts and VMs. The key advantage from these features is logical isolation of a VM and its associated resources.

## Red Hat Virtualization Manager

Red Hat Virtualization Manager (RHV-M) provides centralized enterprise-grade management for the physical and logical resources within the RHV virtualized environment. A web-based GUI with different role-based portals is provided to access RHV-M features.

RHV-M exposes configuration and management of RHV resources with open-source, community-driven RESTful APIs. It also supports full-fledged integration with Red Hat CloudForms and Red Hat Ansible for automation and orchestration.

## Red Hat Virtualization Hosts

Hosts (also called hypervisors) are the physical servers that provide hardware resources for the VMs to run on. A kernel-based virtual machine (KVM) provides full virtualization support, and Virtual Desktop Server Manager (VDSM) is the host agent that is responsible for host communication with the RHV-M.

The two types of hosts supported in Red Hat Virtualization are Red Hat Virtualization Hosts (RHV-H) and Red Hat Enterprise Linux hosts (RHEL).

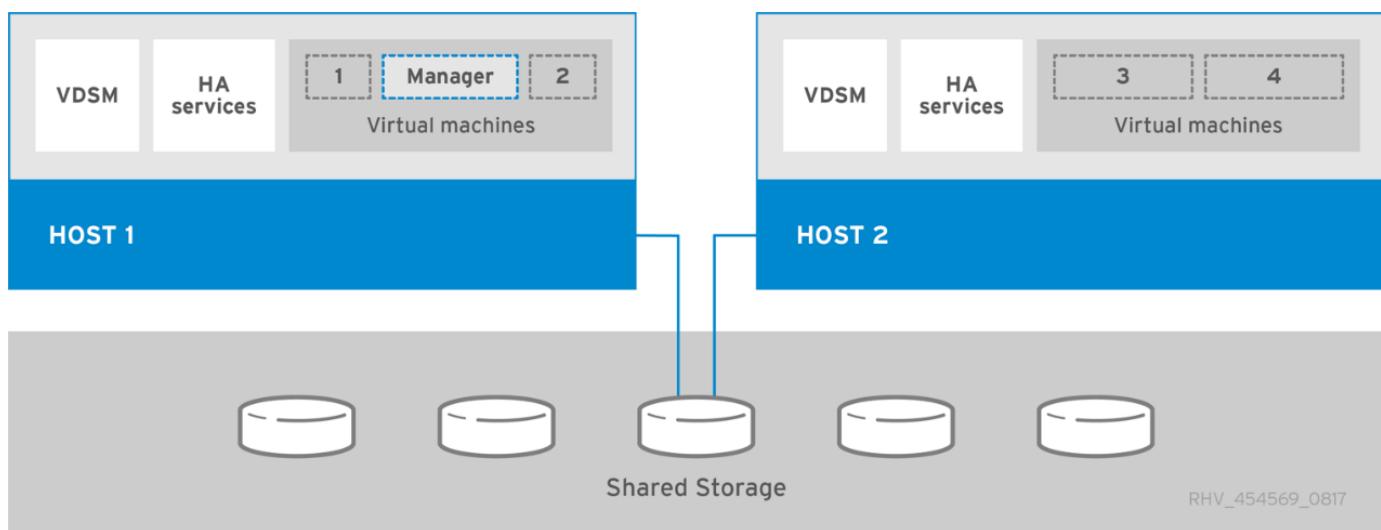
RHV-H is a minimal, light-weight operating system based on Red Hat Enterprise Linux that is optimized for the ease of setting up physical servers as RHV hypervisors.

RHEL hosts are servers that run the standard Red Hat Enterprise Linux operating system. They can then be configured with the required subscriptions to install the packages required to permit the physical servers to be used as RHV hosts.

## Red Hat Virtualization Architecture

Red Hat Virtualization can be deployed in two different architectures, with the RHV-M as a physical server in the infrastructure or with the RHV-M configured as a self-hosted engine. NetApp recommends using the self-hosted engine deployment, in which the RHV-M is a VM hosted in the same environment as other VMs, as we do in this guide.

A minimum of two self-hosted nodes are required for high availability of guest VMs and RHV-M. To provide high availability for the manager VM, HA services are enabled and run on all the self-hosted engine nodes.



[Next: Architecture Overview](#)

## Architecture Overview: NetApp HCI with RHV

### Hardware Requirements

The following table lists the minimum number of hardware components that are required to implement the solution. The hardware components that are used in specific implementations of the solution might vary based on customer requirements.

Hardware	Model	Quantity
NetApp HCI compute nodes	NetApp H410C	2
NetApp HCI storage nodes	NetApp H410S	4
Data switches	Mellanox SN2010	2
Management switches	Cisco Nexus 3048	2

### Software Requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any implementation of the solution might vary based on customer requirements.

Software	Purpose	Version
NetApp HCI	Infrastructure (compute/storage)	1.8
NetApp Element	Storage	12.0
Red Hat Virtualization	Virtualization	4.3.9

[Next: Design Considerations](#)

## Design Considerations: NetApp HCI with RHV

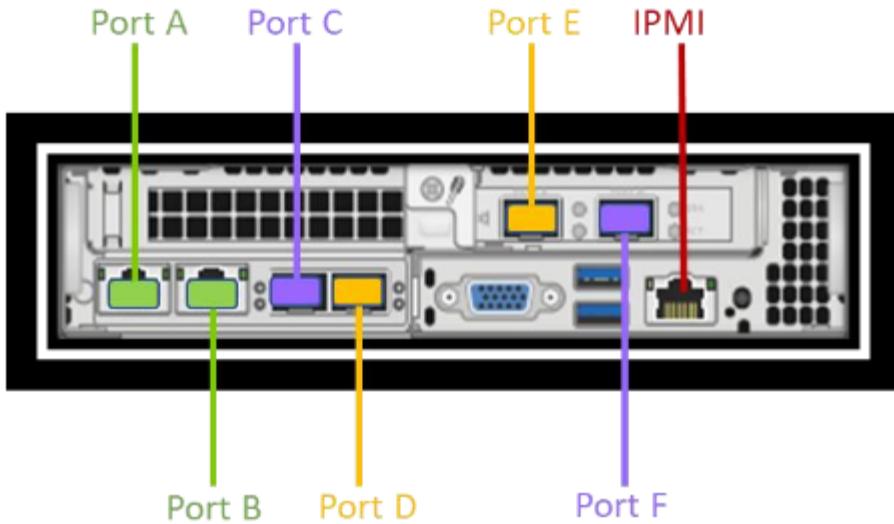
Review the following design considerations when developing your deployment strategy.

### Networking Requirements

This section describes the networking requirements for the deployment of Red Hat Virtualization on NetApp HCI as a validated solution. It provides physical diagrams of the network ports on both the NetApp HCI compute nodes and the switches deployed in the solution. This section also describes the arrangement and purpose of each virtual network segment used in the solution.

### Port Identification

NetApp HCI consists of NetApp H-Series nodes dedicated to either compute or storage. Both node configurations are available with two 1GbE ports (ports A and B) and two 10/25GbE ports (ports C and D) on board. The compute nodes have additional 10/25GbE ports (ports E and F) available in the first mezzanine slot. Each node also has an additional out-of-band management port that supports Intelligent Platform Management Interface (IPMI) functionality. Each of these ports on the rear of an H410C node can be seen in the following figure.



## Network Design

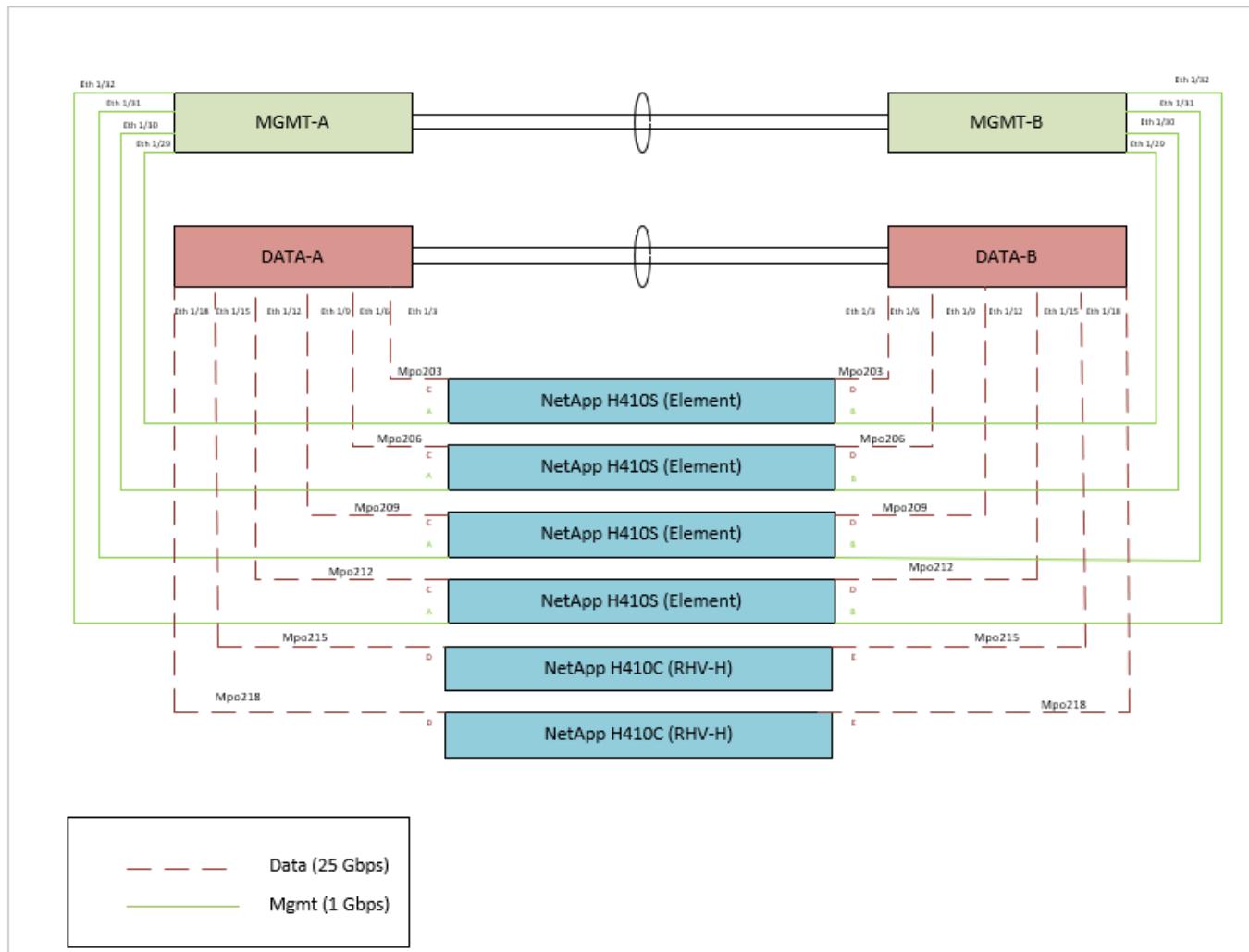
The NetApp HCI with Red Hat Virtualization solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality.

## Cabling Storage Nodes

The management ports A and B must be active on each storage node to configure the NetApp HCI cluster, and provide management accessibility to Element after the solution is deployed. The two 25Gbps ports (C and D) should be connected, one to each data switch, to provide physical fault tolerance. The switch ports should be configured for multi-chassis link aggregation (MLAG) and the data ports on the node should be configured for LACP with jumbo-frames support enabled. The IPMI ports on each node can be used to remotely manage the node after it is installed in a data center. With IPMI, the node can be accessed with a web-browser-based console to run the initial installation, run diagnostics, and reboot or shut down the node if necessary.

## Cabling Compute Nodes

The two 25Gbps ports (C and E) should be connected, one to each data switch, to provide physical fault tolerance. The switch ports should be configured for multi-chassis link aggregation (MLAG), and the data ports on the node should be configured for LACP with jumbo-frames support enabled. The IPMI ports can also be used to remotely manage the node after it is installed in a data center. With IPMI, the node can be accessed with a web-browser-based console to run the initial installation, run diagnostics, and reboot or shut down the node if necessary.



## VLAN Requirements

The solution is designed to logically separate network traffic for different purposes by using Virtual Local Area Networks (VLANs). NetApp HCI requires a minimum of three network segments. However, this configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution, as well as the specific VLAN IDs that are used later in the validated architecture deployment.

VLANs	Purpose	VLAN Used
Out-of-band management network	Management for HCI nodes / IPMI	16
In-band management network	Management for HCI nodes / ovirtmgmt	1172
Storage network	Storage network for NetApp Element.	3343
Migration network	Network for virtual guest migration.	3345
VM network	Network for virtual guests.	3346

## Network Infrastructure Support Resources

The following infrastructure should be in place prior to the deployment of the Red Hat Virtualization on NetApp HCI solution:

- At least one DNS server providing full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- Outbound internet connectivity is recommended, but not required, for both the in-band management network and the VM network.

[Next: Deployment Procedures](#)

## Deployment Summary: NetApp HCI with RHV

The detailed steps provided in this section provide a validation for the minimum hardware and software configuration required to deploy and validate the NetApp HCI with Red Hat Virtualization solution.

Deploying Red Hat Virtualization for NetApp HCI involves the following high-level tasks:

1. [Configure Management Switches](#)
2. [Configure Data Switches](#)
3. [Deploy Element Storage System on HCI Storage Nodes](#)
4. [Install RHV-H to HCI Compute Nodes](#)
5. [Deploy RHV Manager as a Self-hosted Engine](#)
6. [Deploy Test VMs](#)
7. [Test HA Functionality](#)

[Next: Best Practices - Updating RHV Manager and RHV-H Hosts](#)

### 1. Configure Management Switches: NetApp HCI with RHV

Cisco Nexus 3048 switches are used in this deployment procedure to provide 1Gbps connectivity for in and out-of-band management of the compute and storage nodes. These steps begin after the switches have been racked, powered, and put through the initial setup process. To configure the switches to provide management connectivity to the infrastructure, complete the following steps:

#### Enable Advanced Features for Cisco Nexus

Run the following commands on each Cisco Nexus 3048 switch to configure advanced features:

1. Enter configuration mode.

```
Switch-01# configure terminal
```

2. Enable VLAN functionality.

```
Switch-01(config)# feature interface-vlan
```

3. Enable LACP.

```
Switch-01(config)# feature lacp
```

4. Enable virtual port channels (vPCs).

```
Switch-01(config)# feature vpc
```

5. Set the global port-channel load-balancing configuration.

```
Switch-01(config)# port-channel load-balance src-dst ip-l4port
```

6. Perform global spanning-tree configuration.

```
Switch-01(config)# spanning-tree port type network default
Switch-01(config)# spanning-tree port type edge bpduguard default
```

## Configure Ports on the Switch for In-Band Management

1. Run the following commands to create VLANs for management purposes:

```
Switch-01(config)# vlan 2
Switch-01(config-vlan)# Name Native_VLAN
Switch-01(config-vlan)# vlan 16
Switch-01(config-vlan)# Name OOB_Network
Switch-01(config-vlan)# vlan 1172
Switch-01(config-vlan)# Name MGMT_Network
Switch-01(config-vlan)# exit
```

2. Configure the ports ETH1/29-32 as VLAN trunk ports that connect to management interfaces on each HCI storage node.

```
Switch-01(config)# int eth 1/29
Switch-01(config-if)# description HCI-STG-01 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 1172
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# int eth 1/30
Switch-01(config-if)# description HCI-STG-02 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 1172
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# int eth 1/31
Switch-01(config-if)# description HCI-STG-03 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 1172
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# int eth 1/32
Switch-01(config-if)# description HCI-STG-04 PortA
Switch-01(config-if)# switchport mode trunk
Switch-01(config-if)# switchport trunk native vlan 2
Switch-01(config-if)# switchport trunk allowed vlan 1172
Switch-01(config-if)# spanning tree port type edge trunk
Switch-01(config-if)# exit
```

## Configure Ports on the Switch for Out-of-Band Management

Run the following commands to configure the ports for cabling the IPMI interfaces on each HCI node.

```
Switch-01(config)# int eth 1/13
Switch-01(config-if)# description HCI-CMP-01 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# int eth 1/14
Switch-01(config-if)# description HCI-STG-01 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# int eth 1/15
Switch-01(config-if)# description HCI-STG-03 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# exit
```



In the validated configuration, we cabled odd-node IPMI interfaces to Switch-01 and even-node IPMI interfaces to Switch-02.

### Create a vPC Domain to Ensure Fault Tolerance

1. Activate the ports used for the vPC peer-link between the two switches.

```
Switch-01(config)# int eth 1/1
Switch-01(config-if)# description vPC peer-link Switch-02 1/1
Switch-01(config-if)# int eth 1/2
Switch-01(config-if)# description vPC peer-link Switch-02 1/2
Switch-01(config-if)# exit
```

2. Perform the vPC global configuration.

```
Switch-01 (config) # vpc domain 1
Switch-01 (config-vpc-domain) # role priority 10
Switch-01 (config-vpc-domain) # peer-keepalive destination <switch-
02_mgmt_address> source <switch-01_mgmt_address> vrf management
Switch-01 (config-vpc-domain) # peer-gateway
Switch-01 (config-vpc-domain) # auto recovery
Switch-01 (config-vpc-domain) # ip arp synchronize
Switch-01 (config-vpc-domain) # int eth 1/1-2
Switch-01 (config-vpc-domain) # channel-group 10 mode active
Switch-01 (config-vpc-domain) # int Po10
Switch-01 (config-if) # description vPC peer-link
Switch-01 (config-if) # switchport mode trunk
Switch-01 (config-if) # switchport trunk native vlan 2
Switch-01 (config-if) # switchport trunk allowed vlan 16, 1172
Switch-01 (config-if) # spanning-tree port type network
Switch-01 (config-if) # vpc peer-link
Switch-01 (config-if) # exit
```

[Next: 2. Configure Data Switches](#)

## 2. Configure Data Switches: NetApp HCI with RHV

Mellanox SN2010 switches are used in this deployment procedure to provide 25Gbps connectivity for the data plane of the compute and storage nodes. These steps begin after the switches have been racked, cabled, and put through the initial setup process. To configure the switches to provide data connectivity to the infrastructure, complete the following steps:

### Create MLAG Cluster to Provide Fault Tolerance

1. Run the following commands on each Mellanox SN210 switch for general configuration:

- Enter configuration mode.

```
Switch-01 enable
Switch-01 configure terminal
```

- Enable the LACP required for the Inter-Peer Link (IPL).

```
Switch-01 (config) # lacp
```

- Enable the Link Layer Discovery Protocol (LLDP).

```
Switch-01 (config) # lldp
```

- d. Enable IP routing.

```
Switch-01 (config) # ip routing
```

- e. Enable the MLAG protocol.

```
Switch-01 (config) # protocol mlag
```

- f. Enable global QoS.

```
Switch-01 (config) # dcb priority-flow-control enable force
```

2. For MLAG to function, the switches must be made peers to each other through an IPL. This should consist of two or more physical links for redundancy. The MTU for the IPL is set for jumbo frames (9216), and all VLANs are enabled by default. Run the following commands on each switch in the domain:

- a. Create port channel 10 for the IPL.

```
Switch-01 (config) # interface port-channel 10
Switch-01 (config interface port-channel 10) # description IPL
Switch-01 (config interface port-channel 10) # exit
```

- b. Add interfaces ETH 1/20 and 1/22 to the port channel.

```
Switch-01 (config) # interface ethernet 1/20 channel-group 10 mode
active
Switch-01 (config) # interface ethernet 1/20 description ISL-SWB_01
Switch-01 (config) # interface ethernet 1/22 channel-group 10 mode
active
Switch-01 (config) # interface ethernet 1/22 description ISL-SWB_02
```

- c. Create a VLAN outside of the standard range dedicated to IPL traffic.

```
Switch-01 (config) # vlan 4000
Switch-01 (config vlan 4000) # name IPL VLAN
Switch-01 (config vlan 4000) # exit
```

- d. Define the port channel as the IPL.

```
Switch-01 (config) # interface port-channel 10 ipl 1
Switch-01 (config) # interface port-channel 10 dcb priority-flow-
control mode on force
```

- e. Set an IP for each IPL member (non-routable; it is not advertised outside of the switch).

```
Switch-01 (config) # interface vlan 4000
Switch-01 (config vlan 4000) # ip address 10.0.0.1 255.255.255.0
Switch-01 (config vlan 4000) # ipl 1 peer-address 10.0.0.2
Switch-01 (config vlan 4000) # exit
```

3. Create a unique MLAG domain name for the two switches and assign a MLAG virtual IP (VIP). This IP is used for keep-alive heartbeat messages between the two switches. Run these commands on each switch in the domain:

- a. Create the MLAG domain and set the IP address and subnet.

```
Switch-01 (config) # mlag-vip MLAG-VIP-DOM ip a.b.c.d /24 force
```

- b. Create a virtual MAC address for the system MLAG.

```
Switch-01 (config) # mlag system-mac AA:BB:CC:DD:EE:FF
```

- c. Configure the MLAG domain so that it is active globally.

```
Switch-01 (config) # no mlag shutdown
```

The IP used for the MLAG VIP must be in the same subnet as the switch management network (mgmt0). Also, The MAC address used can be any unicast MAC address and must be set to the same value on both switches in the MLAG domain.

## Configure Ports to Connect to Storage and Compute Hosts

1. Create each of the VLANs needed to support the services for NetApp HCI. Run these commands on each switch in the domain:

- a. Create the VLANs.

```
Switch-01 (config) # vlan 1172
Switch-01 (config vlan 1172) exit
Switch-01 (config) # vlan 3343
Switch-01 (config vlan 3343) exit
Switch-01 (config) # vlan 3344
Switch-01 (config vlan 3345) exit
Switch-01 (config) # vlan 3345
Switch-01 (config vlan 3346) exit
```

- b. Create names for each VLAN for easier accounting.

```
Switch-01 (config) # vlan 1172 name "MGMT_Network"
Switch-01 (config) # vlan 3343 name "Storage_Network"
Switch-01 (config) # vlan 3345 name "Migration_Network"
Switch-01 (config) # vlan 3346 name "VM_Network"
```

2. Create MLAG interfaces and hybrid VLANs on ports identified so that you can distribute connectivity between the switches and tag the appropriate VLANs for the NetApp HCI compute nodes.

- a. Select the ports you want to work with.

```
Switch-01 (config) # interface ethernet 1/15
```

- b. Set the MTU for each port.

```
Switch-01 (config interface ethernet 1/15) # mtu 9216 force
```

- c. Modify spanning-tree settings for each port.

```
Switch-01 (config interface ethernet 1/15) # spanning-tree bpdufilter
enable
Switch-01 (config interface ethernet 1/15) # spanning-tree port type
edge
Switch-01 (config interface ethernet 1/15) # spanning-tree bpduguard
enable
```

- d. Set the switchport mode to hybrid.

```
Switch-01 (config interface ethernet 1/15) # switchport mode hybrid
Switch-01 (config interface ethernet 1/15) # exit
```

- e. Create descriptions for each port being modified.

```
Switch-01 (config) # interface ethernet 1/15 description HCI-CMP-01  
PortD
```

- f. Create and configure the MLAG port channels.

```
Switch-01 (config) # interface mlag-port-channel 215  
Switch-01 (config interface mlag-port-channel 215) # exit  
Switch-01 (config) # interface mlag-port-channel 215 no shutdown  
Switch-01 (config) # interface mlag-port-channel 215 mtu 9216 force  
Switch-01 (config) # interface ethernet 1/15 lacp port-priority 10  
Switch-01 (config) # interface ethernet 1/15 lacp rate fast  
Switch-01 (config) # interface ethernet 1/15 mlag-channel-group 215  
mode active
```

- g. Tag the appropriate VLANs for the NetApp HCI environment.

```
Switch-01 (config) # interface mlag-port-channel 215 switchport  
hybrid  
Switch-01 (config) # interface mlag-port-channel 215 switchport  
hybrid allowed-vlan add 1172  
Switch-01 (config) # interface mlag-port-channel 215 switchport  
hybrid allowed-vlan add 3343  
Switch-01 (config) # interface mlag-port-channel 215 switchport  
hybrid allowed-vlan add 3345  
Switch-01 (config) # interface mlag-port-channel 215 switchport  
hybrid allowed-vlan add 3346
```

3. Create MLAG interfaces and hybrid VLAN ports identified so that you can distribute connectivity between the switches and tag the appropriate VLANs for the NetApp HCI storage nodes.

- a. Select the ports that you want to work with.

```
Switch-01 (config) # interface ethernet 1/3
```

- b. Set the MTU for each port.

```
Switch-01 (config interface ethernet 1/3) # mtu 9216 force
```

- c. Modify spanning tree settings for each port.

```
Switch-01 (config interface ethernet 1/3) # spanning-tree bpdulfiler  
enable  
Switch-01 (config interface ethernet 1/3) # spanning-tree port type  
edge  
Switch-01 (config interface ethernet 1/3) # spanning-tree bpduguard  
enable
```

d. Set the switchport mode to hybrid.

```
Switch-01 (config interface ethernet 1/3) # switchport mode hybrid  
Switch-01 (config interface ethernet 1/3) # exit
```

e. Create descriptions for each port being modified.

```
Switch-01 (config) # interface ethernet 1/3 description HCI-STG-01  
PortD
```

f. Create and configure the MLAG port channels.

```
Switch-01 (config) # interface mlag-port-channel 203  
Switch-01 (config interface mlag-port-channel 203) # exit  
Switch-01 (config) # interface mlag-port-channel 203 no shutdown  
Switch-01 (config) # interface mlag-port-channel 203 mtu 9216 force  
Switch-01 (config) # interface mlag-port-channel 203 lacp-individual  
enable force  
Switch-01 (config) # interface ethernet 203 lacp port-priority 10  
Switch-01 (config) # interface ethernet 203 lacp rate fast  
Switch-01 (config) # interface ethernet 1/3 mlag-channel-group 203  
mode active
```

g. Tag the appropriate VLANs for the storage environment.

```
Switch-01 (config) # interface mlag-port-channel 203 switchport mode  
hybrid  
Switch-01 (config) # interface mlag-port-channel 203 switchport  
hybrid allowed-vlan add 1172  
Switch-01 (config) # interface mlag-port-channel 203 switchport  
hybrid allowed-vlan add 3343
```



The configurations in this section show the configuration for a single port as example. They must also be run for each additional port connected in the solution, as well as on the associated port of the second switch in the MLAG domain. NetApp recommends that the descriptions for each port are updated to reflect the device ports that are being cabled and configured on the other switch.

## Create Uplink Ports for the Switches

1. Create an MLAG interface to provide uplinks to both Mellanox SN2010 switches from the core network.

```
Switch-01 (config) # interface mlag port-channel 201
Switch-01 (config interface mlag port-channel) # description Uplink
CORE-SWITCH port PORT
Switch-01 (config interface mlag port-channel) # exit
```

2. Configure the MLAG members.

```
Switch-01 (config) # interface ethernet 1/1 description Uplink to CORE-
SWITCH port PORT
Switch-01 (config) # interface ethernet 1/1 speed 10000 force
Switch-01 (config) # interface mlag-port-channel 201 mtu 9216 force
Switch-01 (config) # interface ethernet 1/1 mlag-channel-group 201 mode
active
```

3. Set the switchport mode to hybrid and allow all VLANs from the core uplink switches.

```
Switch-01 (config) # interface mlag-port-channel switchport mode hybrid
Switch-01 (config) # interface mlag-port-channel switchport hybrid
allowed-vlan all
```

4. Verify that the MLAG interface is up.

```
Switch-01 (config) # interface mlag-port-channel 201 no shutdown
Switch-01 (config) # exit
```



The configurations in this section must also be run on the second switch in the MLAG domain. NetApp recommends that the descriptions for each port are updated to reflect the device ports that are being cabled and configured on the other switch.

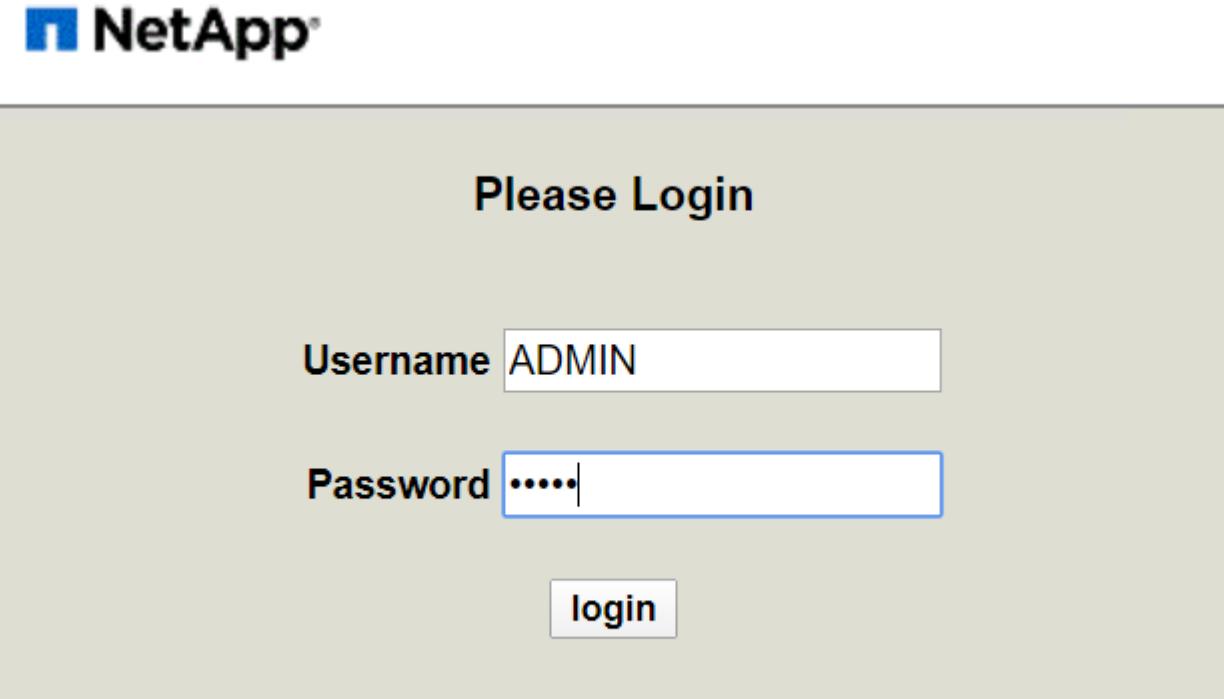
[Next: 3. Deploy the Element Storage System on the HCI Storage Nodes](#)

### 3. Deploy the Element Storage System on the HCI Storage Nodes: NetApp HCI with RHV

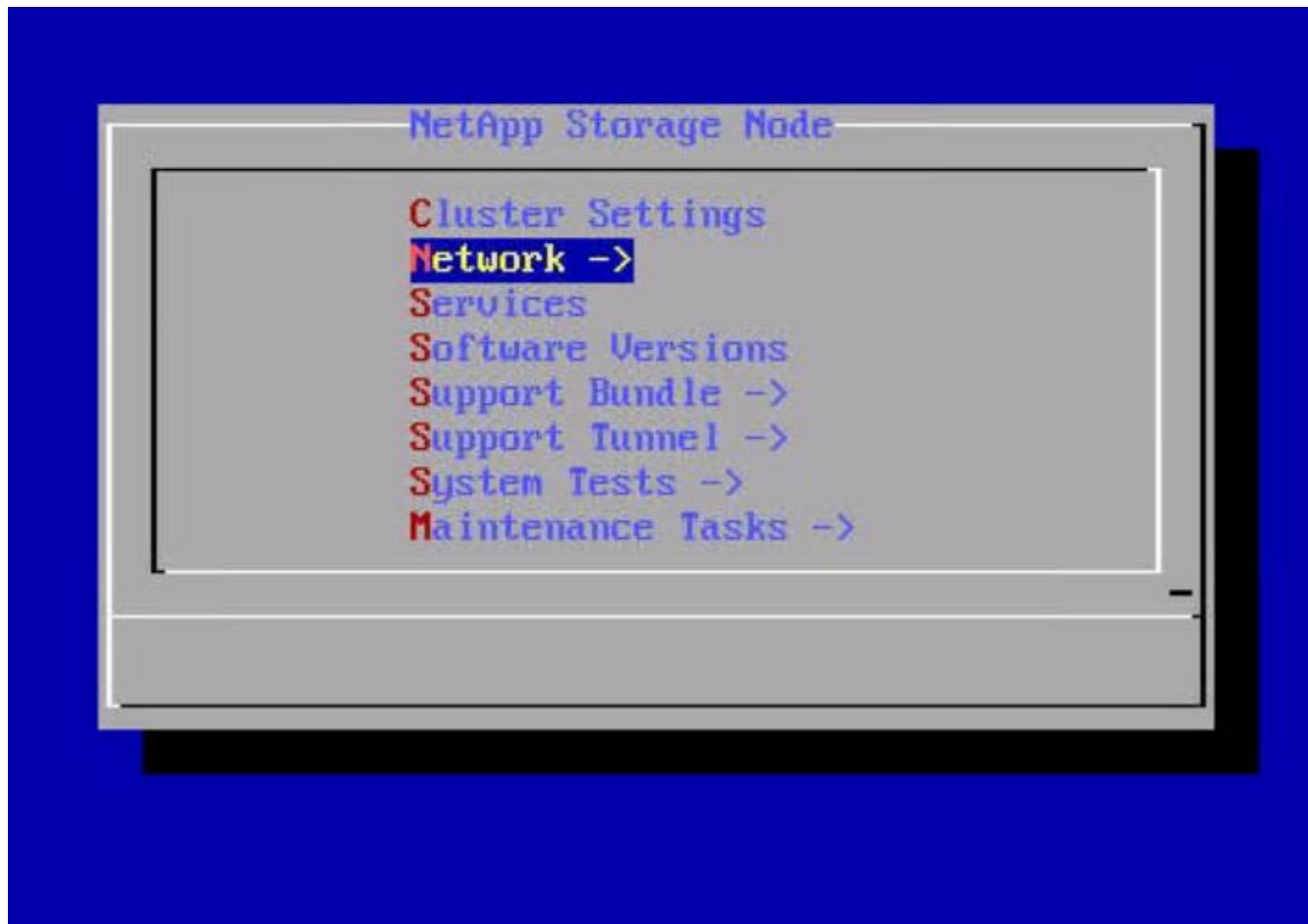
#### Basic NetApp Element Storage Setup

NetApp Element cluster setup is performed in a manner similar to a standalone NetApp SolidFire storage setup. These steps begin after the nodes have been racked, and cabled, and the IPMI port has been configured on each node using the console. To setup a storage cluster, complete the following steps:

1. Access the out-of-band management console for the storage nodes in the cluster and log in with the default credentials ADMIN/ADMIN.



2. Click the Remote Console Preview image in the center of the screen to download a JNLP file launched by Java Web Start, which launches an interactive console to the system.



3. Navigate to Network > Network Config > Bond1G (Management) and configure the Bond1G interface. The Bond1G interface should be in ActivePassive bond mode and must have an IP, a netmask, and a gateway set statically. Its VLAN must correspond to IB Management network and DNS servers defined for the environment. Then click OK.

NetApp Storage Node -> Network -> Network Config -> Bond1G

Hit 'tab' to navigate between the form and buttons. Use ↑↓ to navigate between fields. Start typing or hit ←→ to enter the field to make changes. Press 'enter' with a field selected, or hit 'tab' then 'enter' to submit all pending changes.

\* denotes required fields.

Method:	static
Link speed:	1000
*IPv4 Address:	10.63.172.136
*IPv4 Subnet_Mask:	255.255.255.0
*IPv4 Gateway:	10.63.172.1
Mtu:	1500
Dns:	10.61.184.251, 10.61.184.252
Domains:	cie.netapp.com
IPv6 Address:	
IPv6 Gateway:	
*Bond mode:	ActivePassive
*Status:	UpAndRunning
Vlan:	1172

< **OK** >

<**Cancel**>

< **Help** >

4. Select Bond10G (Storage) and configure the Bond10G interface. The Bond 10G interface must be in LACP bonding mode and have the MTU set to 9000 to enable jumbo frames. It must be assigned an IP address and netmask that are available on the defined storage VLAN. Click OK after entering the details.

NetApp Storage Node -> Network -> Network Config -> Bond10G

Hit 'tab' to navigate between the form and buttons. Use ↑↓ to navigate between fields. Start typing or hit ←→ to enter the field to make changes. Press 'enter' with a field selected, or hit 'tab' then 'enter' to submit all pending changes.

\* denotes required fields.

Method:	static
Link speed:	50000
*IPv4 Address:	172.21.87.130
*IPv4 Subnet_Mask:	255.255.255.0
IPv4 Gateway:	
Mtu:	9000
*Bond mode:	LACP
*Status:	UpAndRunning
Vlan:	3343

< **OK** >

<**Cancel**>

< **Help** >

5. Go back to the initial screen, navigate to Cluster Settings, and click Change Settings. Enter the Cluster Name of your choice and click OK.

## Change Cluster Settings

Hit 'tab' to navigate between the form and buttons. Use ↑/↓ to navigate between fields. Start typing or hit ←/→ to enter the field to make changes. Press 'enter' with a field selected, or hit 'tab' then 'enter' to submit all pending changes.  
\* denotes required fields.

*Hostname:	SF-1A94
Cluster:	RHV-Store
*Management Interface:	Bond1G

< [OK](#) >

<[Cancel](#)>

6. Repeat steps 1 to 5 for all HCI storage nodes.
7. After all the storage nodes are configured, use a web browser to log into the IB Management IP of one of the storage nodes. This presents the setup page with the Create a New Cluster dialog. Management VIP, storage VIP, and other details of the Element cluster are configured on this page. The storage nodes that were configured in the previous step are automatically detected. Make sure that any nodes that you do not want in the cluster are unchecked before proceeding. Accept the End User License Agreement and click Create New Cluster to begin the cluster creation process. It takes a few minutes to get the cluster up.



In some cases, visiting the IB management address automatically connects on port 442 and launches the NDE setup wizard. If this happens, delete the port specification from the URL and reconnect to the page.

IP Address	Version	Include
172.21.87.30	12.0.0.333	<input checked="" type="checkbox"/>
172.21.87.32	12.0.0.333	<input checked="" type="checkbox"/>
172.21.87.130	12.0.0.333	<input checked="" type="checkbox"/>
172.21.87.132	12.0.0.333	<input checked="" type="checkbox"/>

8. After the cluster is created, it redirects to the Element cluster management interface available at the assigned MVIP address. Log in with the credentials provided in the previous step.
9. After you log in, the cluster automatically detects the number of available drives and requests for confirmation to add all drives. Click Add Drives to add all drives at once.
10. The Element cluster is ready to use. Navigate to Cluster > Nodes, and all four nodes should be in a healthy state with active drives.

Active	Node ID	Node Name	Node Role	Node Type	Active Drives	Management IP	Cluster IP	Storage IP	Management VLAN ID	Storage VLAN ID
Active	4	SF-1D1B	Ensemble Node	H410S-1	6	10.63.172.138	172.21.87.132	172.21.87.132	1172	3343
Active	3	SF-1A94	Ensemble Node	H410S-1	6	10.63.172.136	172.21.87.130	172.21.87.130	1172	3343
Active	2	SF-34F7	Cluster Master, Ensemble Node	H410S-1	6	10.63.172.139	172.21.87.32	172.21.87.32	1172	3343
Active	1	SF-1FA7	-	H410S-1	6	10.63.172.137	172.21.87.30	172.21.87.30	1172	3343

Showing 1 - 4 of 4 Nodes

## Element Storage Configuration to Support RHV Deployment

In our NetApp HCI for Red Hat Virtualization solution, we use a NetApp Element storage system to provide the backend storage support for RHV's requirement of shared storage domains. The self-hosted engine architecture of RHV deployment requires two storage domains at a minimum—one for the hosted engine storage domain and one for the guest VM data domain.

For this part of deployment, you must configure an account, two volumes of appropriate size, and the associated initiators. Then map these components to an access group that allows the RHV hosts to map the

block volumes for use. Each of these actions can be performed through the web user interface or through the native API for the Element system. For this deployment guide, we go through the steps with the GUI.

Log in to the NetApp Element cluster GUI at its MVIP address using a web browser. Navigate to the Management tab and complete the following steps:

1. To create accounts, go to the Accounts sub-tab and click Create Account. Enter the name of your choice and click Create Account.

**Create a New Account** X

---

**Account Details**

Username

**CHAP Settings**

Initiator Secret

Target Secret

---

**Create Account** **Cancel**

2. To create volumes, complete the following steps:
  - a. Navigate to the Volumes sub-tab and click Create Volume.
  - b. To create the volume for the self-hosted engine storage domain, enter the name of your choice, select the account you created in the last step, enter the size of the volume for the self-hosted engine storage domain, configure the QoS setting, and click Create Volume.

## Volume Details

Volume Name

Volume Size

Block Size

 512e  4k

Account



## Quality of Service

Policy

Custom Settings

IO Size	Min IOPS	Max IOPS	Burst IOPS
4 KB	50	15000	15000
8 KB	31 IOPS	9375 IOPS	9375 IOPS
16 KB	19 IOPS	5556 IOPS	5556 IOPS
262 KB	1 IOPS	385 IOPS	385 IOPS

Max Bandwidth	104.86 MB/sec	104.86 MB/sec
---------------	---------------	---------------

The minimum size for the hosted engine volume is 75GB. In our design, we added additional space to allow for future extents to be added to the RHV-M VM if necessary.

- c. To create the volume for the guest VMs data storage domain, enter the name of your choice, select the account you created in the last step, enter the size of the volume for the data storage domain, configure the QoS setting and click Create Volume.

## Volume Details

Volume Name

Volume Size



Block Size

 512e

4k

Account



## Quality of Service

Policy

Custom Settings

IO Size	Min IOPS	Max IOPS	Burst IOPS
4 KB	50	15000	15000
8 KB	31 IOPS	9375 IOPS	9375 IOPS
16 KB	19 IOPS	5556 IOPS	5556 IOPS
262 KB	1 IOPS	385 IOPS	385 IOPS

Max Bandwidth	104.86 MB/sec	104.86 MB/sec
---------------	------------------	------------------

The size of the data domain depends on the kind of VMs run in the environment and the space required to support them. Adjust the size of this volume to meet the needs of your environment.

3. To create initiators, complete the following steps:

- a. Go to the Initiators sub-tab and click Create Initiator.
- b. Select the Bulk Create Initiators radio button and enter the initiators' details of both the RHV-H nodes with comma separated values. Then click Add Initiators, enter the aliases for the initiators, and click the tick button. Verify the details and click Create Initiators.

## Create a New Initiator



Create a Single Initiator

IQN/WWPN

Alias

Bulk Create Initiators

Initiators	2	
Name	Alias (optional)	
iqn.1994-05.com.redhat:rhv-host-node-01	RHV-H01	
iqn.1994-05.com.redhat:rhv-host-node-02	RHV-H02	

**Create Initiators**

**Cancel**

4. To create access groups, complete the following steps:

- a. Go to the Access Groups sub-tab and click Create Access Groups.
- b. Enter the name of your choice, select the initiators for both RHV-H nodes that were created in the previous step, select the volumes, and click Create Access Group.

## Volume Access Group Details

Name

### Add Initiators

Initiators

[Create Initiator?](#)

Initiators			2
ID	Name	Alias	
3	iqn.1994-05.com.redhat:rhv-host-node-01	RHV-H01	
4	iqn.1994-05.com.redhat:rhv-host-node-02	RHV-H02	

Delete orphan initiators

### Attach Volumes

Volumes

Attached Volumes			2
ID	Name		
1	RHV-HostedEngine		
2	RHV-DataDomain		

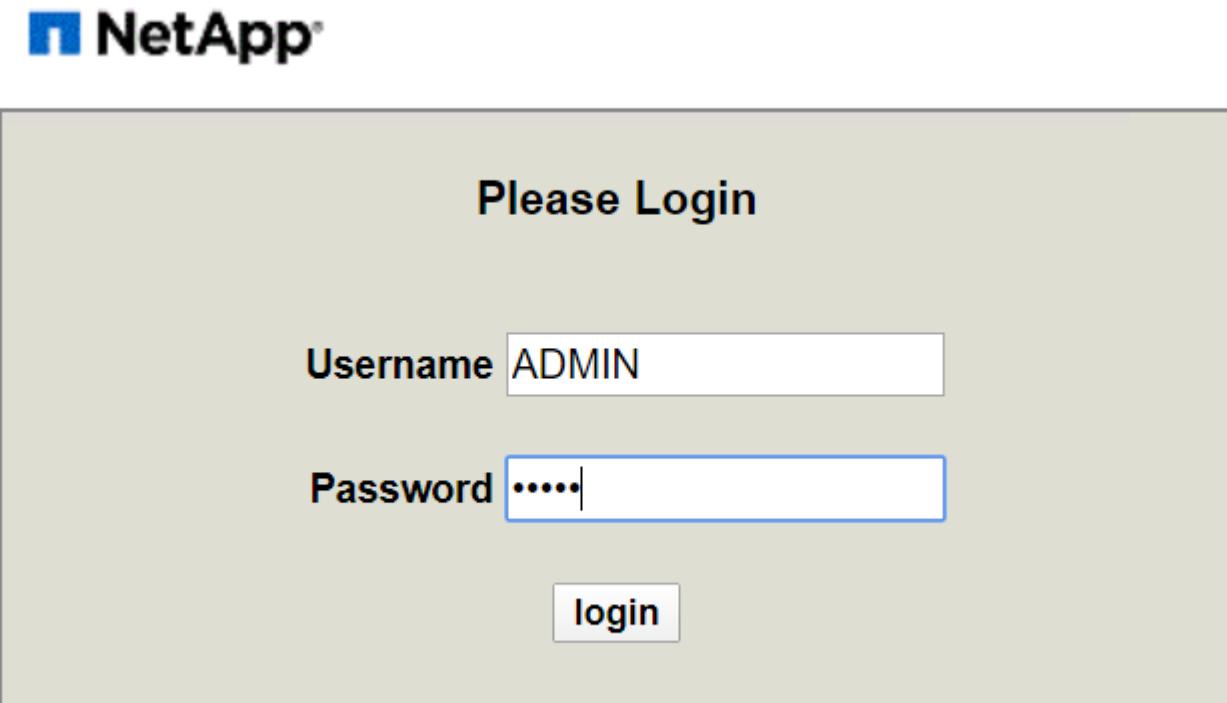
Next: 4. Deploy the RHV-H Hypervisor on the HCI Compute Nodes

#### 4. Deploy the RHV-H Hypervisor on the HCI Compute Nodes: NetApp HCI with RHV

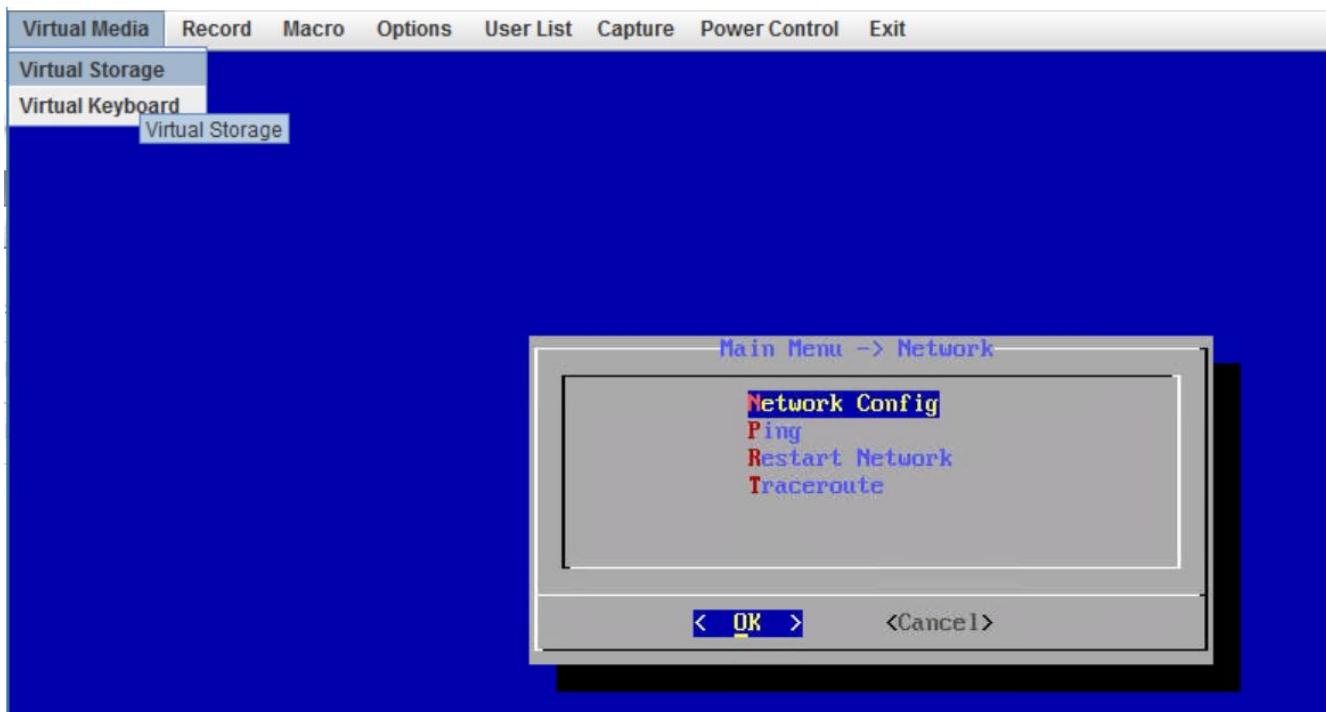
This solution employs the recommended self-hosted engine architecture of RHV deployment with the minimum setup (two self-hosted engine nodes). These steps begin

after the nodes have been racked and cabled and the IPMI port has been configured on each node for using the console. To deploy the RHV-H hypervisor on HCI compute nodes, complete the following steps:

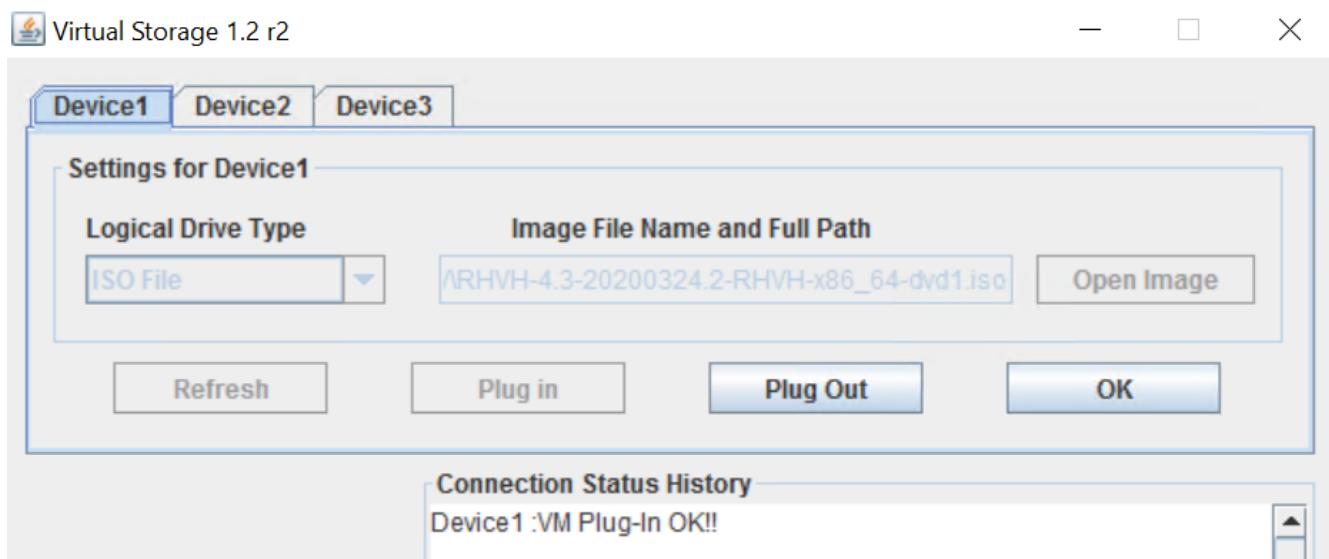
1. Access the out-of-band management console for the compute nodes in the cluster and log in with the default credentials ADMIN/ADMIN.



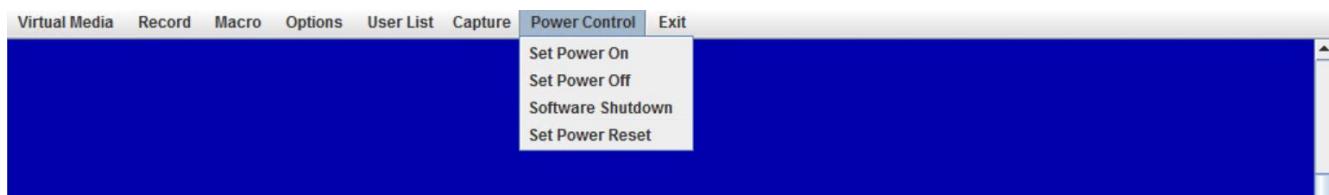
2. Click the Remote Console Preview image in the center of the screen to download a JNLP file launched by Java Web Start, which launches an interactive console to the system.
3. After the virtual console launches, attach the RHV-H 4.3.9 ISO by navigating to and clicking Virtual Media > Virtual Storage.



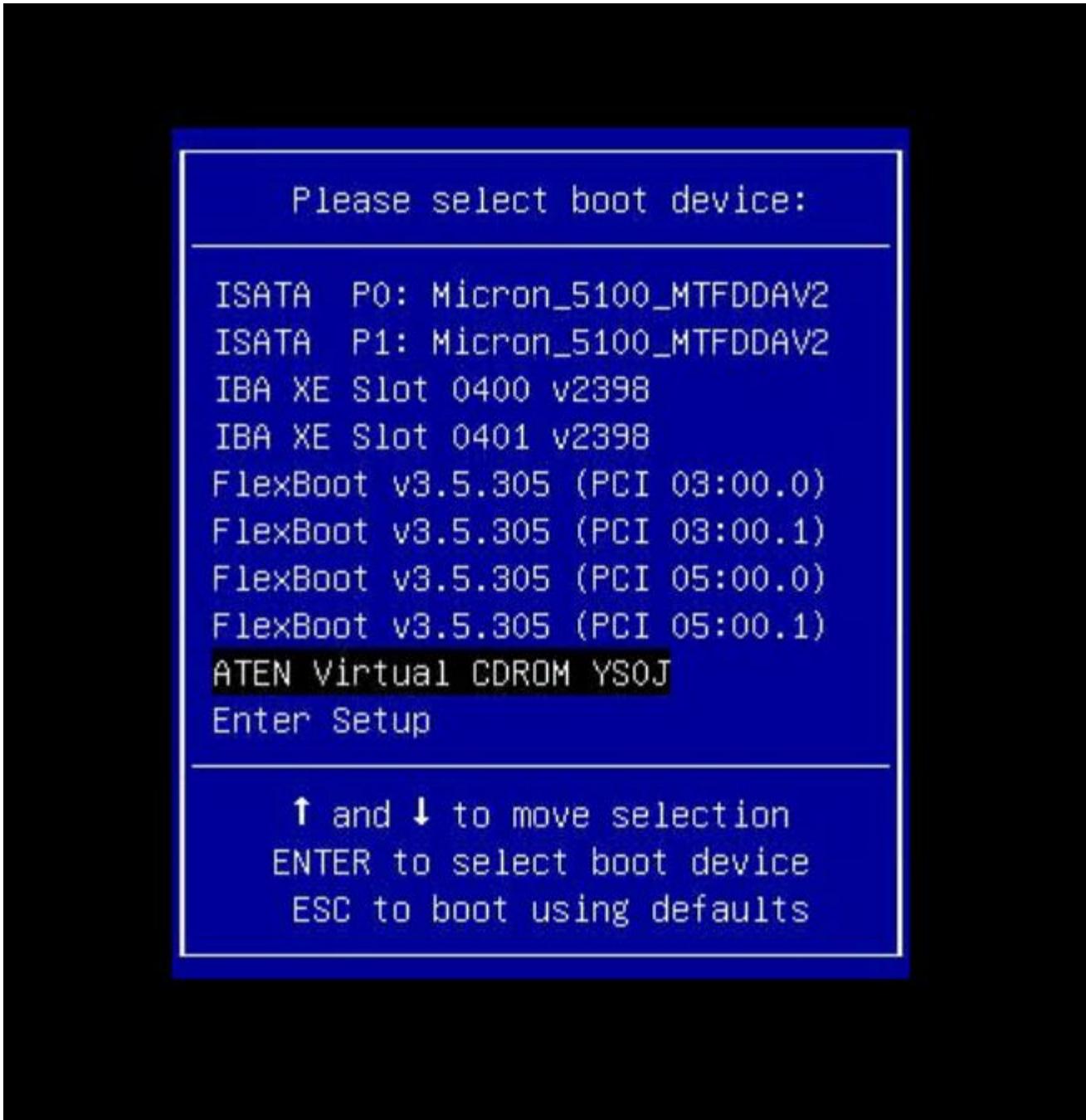
4. For Logical Drive Type, select ISO File from the drop down. Provide the full path and full name of the RHV-H 4.3.9 ISO file or attach it by clicking the Open Image button. Then click Plug In.



5. Reboot the server so that it boots using RHV-H 4.3.9 ISO by navigating and clicking Power Control > Set Power Reset.



6. When the node reboots and the initial screen appears, press F11 to enter the boot menu. From the boot menu, navigate to and click ATEN Virtual CDROM YSOJ.



7. On the next screen, navigate to and click Install RHV 4.3. This loads the image, runs the pre-installation scripts, and starts Anaconda, the Red Hat Enterprise Linux system installer.

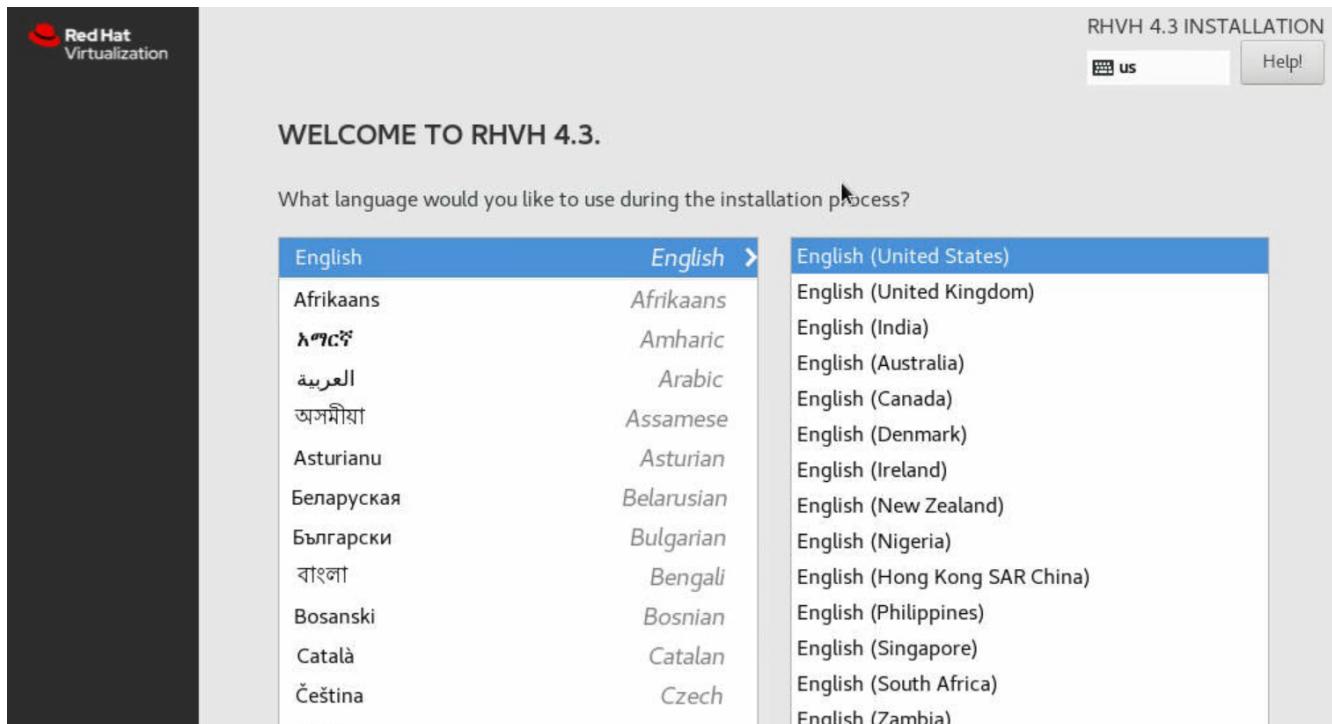
RHVH 4.3

Install RHVH 4.3  
Test this media & install RHVH 4.3

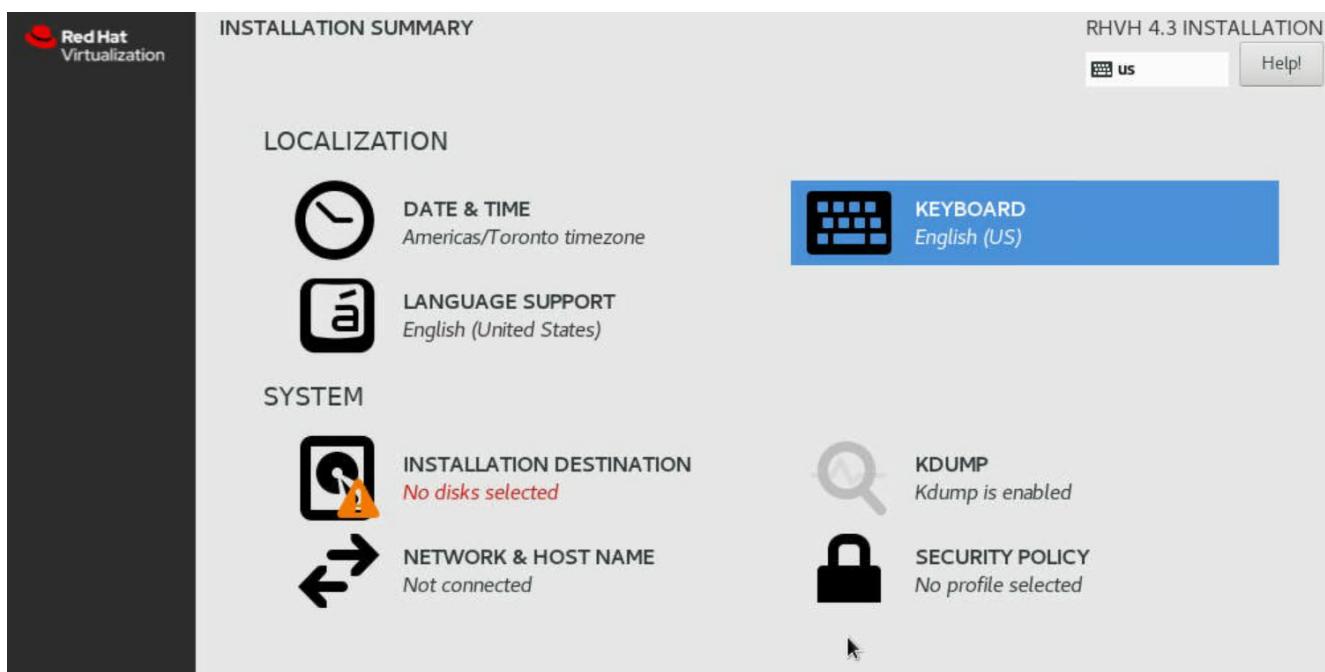
Troubleshooting >

Press Tab for full configuration options on menu items.

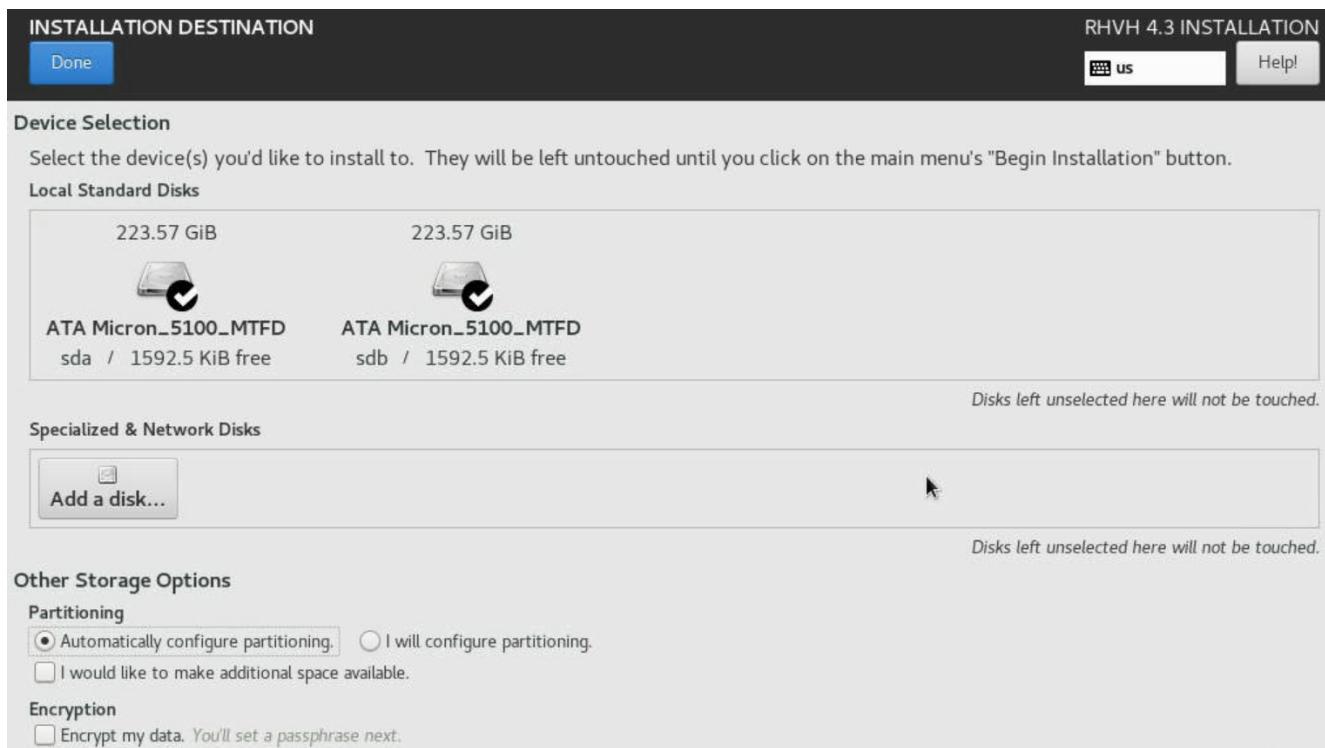
8. The installation welcome screen appears. Select the preferred language and click Next.



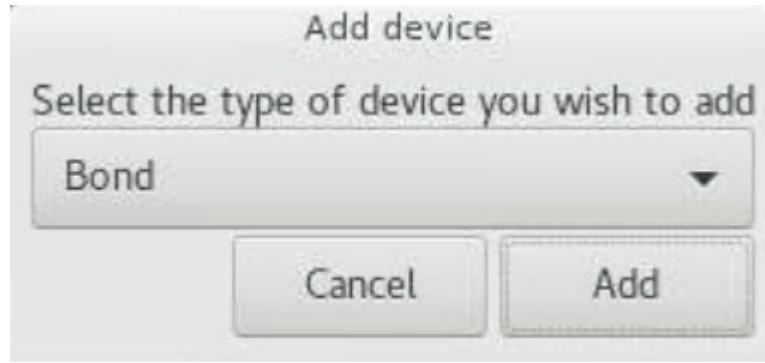
9. In the next screen, select your time zone under Date & Time. The default is UTC. However, NetApp recommends that you configure NTP servers for your environment on this screen. Then select the keyboard language and click Done.



10. Next, click Installation Destination. In the Installation Destination screen, select the drives on which you want to install RHV-H. Verify that Automatically Configure Partitioning is selected in the Partitioning section. Optionally, you can enable encryption by checking the box next to Encrypt My Data. Click Done to confirm the settings.



11. Click Network & Host Name. Provide the desired host name at the bottom of the screen. Then click the (+) button at the bottom. Select the Bond from the drop down and click Add.



12. Next, in the bond configuration screen, click Add to add the member interfaces to the bond interface.

## Editing Bond connection 1

Connection name: **Bond connection 1**

**General      Bond      Proxy      IPv4 Settings      IPv6 Settings**

Interface name: **bond0**

Bonded connections:

	<b>Add</b>
	<b>Edit</b>
	<b>Delete</b>

Mode: **Round-robin**

Link Monitoring: **MII (recommended)**

Monitoring frequency: **1**   ms

Link up delay: **0**   ms

Link down delay: **0**   ms

MTU: **automatic**   bytes

**Cancel** **Save**

13. Select Ethernet from the drop down, indicating that the Ethernet interface is added as a member to the bond interface. Click Create.



## Choose a Connection Type

Select the type of connection you wish to create.

If you are creating a VPN, and the VPN connection you wish to create does not appear in the list, you may not have the correct VPN plugin installed.

Ethernet

Cancel

Create...

- From the Device dropdown in the slave 1 configuration screen, select the Ethernet interface. Verify that the MTU is set to 9000. Click Save.

## Editing bond0 slave 1

Connection name: **bond0 slave 1**

General    **Ethernet**    802.1X Security    DCB

Device:	en01 (AC:1F:6B:8D:85:28)	▼	
Cloned MAC address:	▼		
MTU:	9000	- + bytes	
Wake on LAN:	<input checked="" type="checkbox"/> Default <input type="checkbox"/> Phy <input type="checkbox"/> Unicast <input type="checkbox"/> Multicast <input type="checkbox"/> Ignore <input type="checkbox"/> Broadcast <input type="checkbox"/> Arp <input type="checkbox"/> Magic		
Wake on LAN password:			▼
Link negotiation:	Automatic		▼
Speed:	100 Mb/s		▼
Duplex:	Full		▼

**Cancel**    **Save**

15. Repeat steps 12, 13, and 14 to add the other Ethernet port to the bond0 interface.
16. From the Mode dropdown in the bond configuration screen, select 802.3ad for LACP. Verify that the MTU is set to 9000. Then click Save.

Editing Bond connection 1

Connection name: Bond connection 1

General Bond Proxy IPv4 Settings IPv6 Settings

Interface name: bond0

Bonded connections:

- bond0 slave 1
- bond0 slave 2

Add Edit Delete

Mode: 802.3ad

Link Monitoring: MII (recommended)

Monitoring frequency: 1 ms

Link up delay: 0 ms

Link down delay: 0 ms

MTU: 9000 bytes

Cancel Save

17. Create the VLAN interface for the in-band management network. Click the (+) button again, select VLAN from the dropdown and click Create.



18. In the Editing VLAN connection screen, select bond0 in the Parent Interface dropdown, enter the VLAN ID of the in-band management network. Provide the name of the VLAN interface in `bond 0.< vlan_id >` format.

**Editing VLAN connection 1**

Connection name: **VLAN connection 1**

**VLAN** **General** **Proxy** **IPv4 Settings** **IPv6 Settings**

Parent interface	bond0 (via "Bond connection 1")	▼
VLAN id	1172	- +
VLAN interface name	bond0.1172	
Cloned MAC address		▼
MTU	automatic	- + bytes
Flags	<input checked="" type="checkbox"/> Reorder headers <input type="checkbox"/> GVRP <input type="checkbox"/> Loose binding <input type="checkbox"/> MVRP	

**Cancel** **Save**

19. In the Editing VLAN connection screen, click the IPv4 Settings sub-tab. In the IPv4 Settings sub-tab, configure the network address, netmask, gateway, and DNS servers corresponding to the in-band management network. Click Save to confirm the settings.

**Editing VLAN connection 1**

Connection name: **VLAN connection 1**

General    VLAN    Proxy    **IPv4 Settings**    IPv6 Settings

Method: **Manual**

**Addresses**

Address	Netmask	Gateway
10.63.172.151	24	10.63.172.1

Add    Delete

DNS servers: **10.61.184.251, 10.61.184.252**

Search domains: **cie.netapp.com**

DHCP client ID:

Require IPv4 addressing for this connection to complete

Routes...

Cancel    Save

The screenshot shows the 'Editing VLAN connection 1' dialog box. The 'IPv4 Settings' tab is active. The 'Method' dropdown is set to 'Manual'. The 'Addresses' section contains one entry: Address 10.63.172.151, Netmask 24, and Gateway 10.63.172.1. There are 'Add' and 'Delete' buttons for managing addresses. Below the table are fields for 'DNS servers' (10.61.184.251, 10.61.184.252) and 'Search domains' (cie.netapp.com). A 'DHCP client ID' field is present. A checkbox 'Require IPv4 addressing for this connection to complete' is checked. At the bottom are 'Routes...', 'Cancel', and 'Save' buttons.

20. Create the VLAN interface for the storage network. Click the (+) button again, select VLAN from the dropdown, and click Create. In the Editing VLAN Connection screen, select bond0 in the Parent Interface dropdown, enter the VLAN ID of the storage network, provide the name of the VLAN interface in the `bond 0.<vlan_id>` format. Adjust the MTU to 9000 to allow jumbo frame support. Click Save.

Editing VLAN connection 2

Connection name: **VLAN connection 2**

**General    VLAN    Proxy    IPv4 Settings    IPv6 Settings**

Parent interface: bond0 (via "Bond connection 1")

VLAN id: 3343

VLAN interface name: bond0.3343

Cloned MAC address:

MTU: 9000

Flags:  Reorder headers  GVRP  Loose binding  MVRP

**Cancel**    **Save**

21. In the Editing VLAN Connection screen, click the IPv4 Settings sub-tab. In the IPv4 Settings sub-tab, configure the network address and the netmask corresponding to the storage network. Click Save to confirm the settings.

Editing VLAN connection 2 (on localhost.localdomain) X

Connection name: VLAN connection 2

General VLAN Proxy **IPv4 Settings** IPv6 Settings

Method: **Manual** ▾

**Addresses**

Address	Netmask	Gateway	
172.21.87.31	255.255.255.0		<b>Add</b>
			<b>Delete</b>

DNS servers:

Search domains:

DHCP client ID:

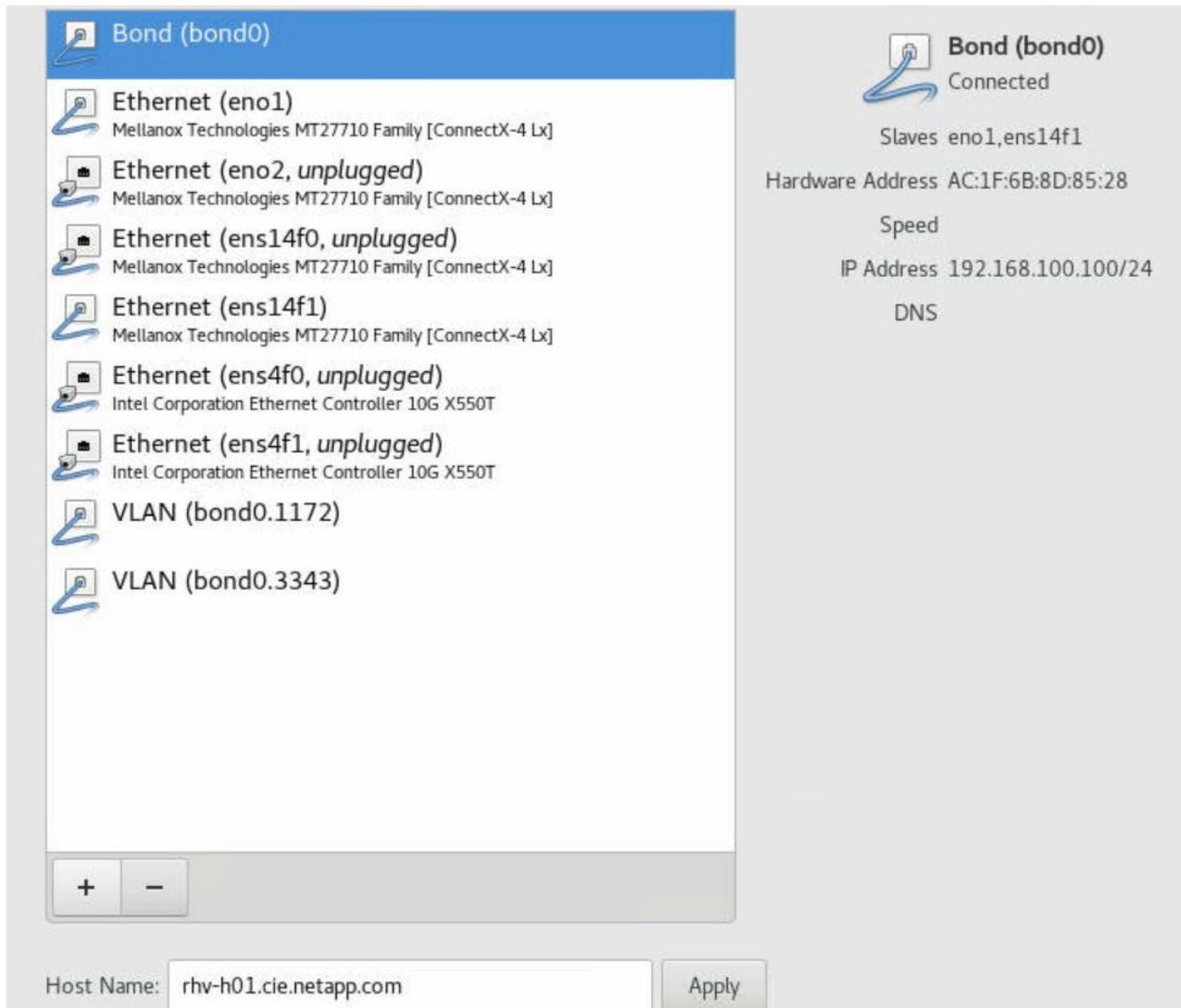
**Require IPv4 addressing for this connection to complete**

**Routes...**

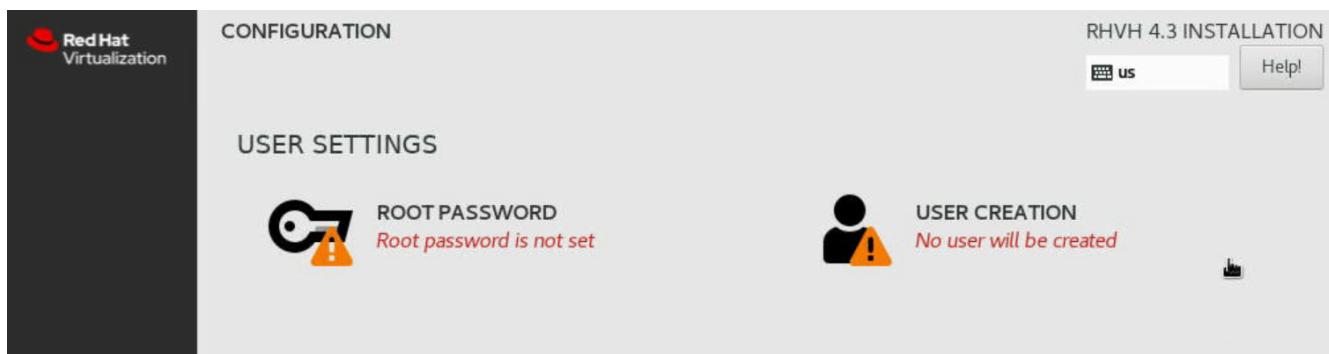
**Cancel** **Save**

The screenshot shows the 'Editing VLAN connection 2' dialog box. The 'IPv4 Settings' tab is active. The 'Method' dropdown is set to 'Manual'. In the 'Addresses' section, there is one entry: Address 172.21.87.31 and Netmask 255.255.255.0. There are 'Add' and 'Delete' buttons next to the table. Below the table are fields for 'DNS servers', 'Search domains', and 'DHCP client ID'. A checkbox labeled 'Require IPv4 addressing for this connection to complete' is checked. At the bottom are 'Cancel' and 'Save' buttons.

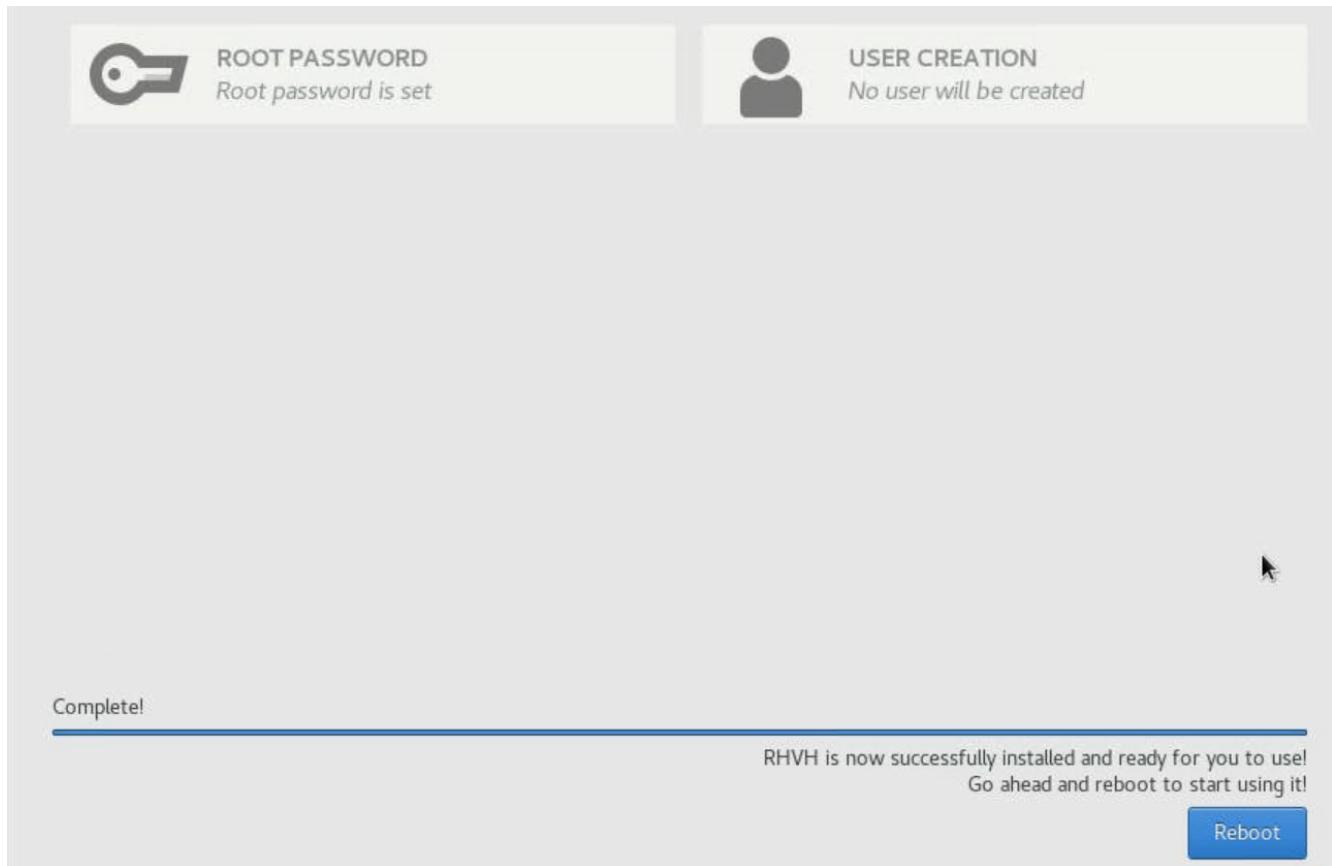
22. Confirm that the network interfaces are up and click Done.



23. After the wizard navigates back to the configuration page, click Begin Installation. The next screen prompts you to configure the root password and optionally to create another user for logging into RHV-H.



24. After the installation completes, unmount the ISO file by navigating to Virtual media > Virtual Storage in the virtual console and click Plug Out. Then click Reboot on the Anaconda GUI to complete the installation process. The node then reboots.



After the node comes up, it displays the login screen.

```
Red Hat Virtualization Host 4.3.9 (el7.8)
Kernel 3.10.0-1127.el7.x86_64 on an x86_64
```

```
rhv-h01 login:
```

25. Now that the installation is complete, you must then register RHV-H and enable the required repositories. Open a browser and log in to the Cockpit user interface at <https://<HostFQDN/IP>:9090> using the root credentials provided during the installation.

RED HAT VIRTUALIZATION HOST 4.3.9 (EL7.8)

Virtualization

Node Status

Health **ok** (green circle)

Current Layer [rhvh-4.3.9.2-0.20200324.0+](#) Rollback

System

Networking Information: [View](#)

System Logs: [View](#)

Storage: [View](#)

SSH Host Key: [View](#)

V Hosted Engine

26. Navigate to localhost > Subscriptions and click Register. Enter your Red Hat Portal username and password, click the check box Connect this System to Red Hat Insights, and click Register. The system automatically subscribes to the Red Hat Virtualization Host entitlement.

Red Hat Insights provide continuous analysis of registered systems to proactively recognize threats to availability, security, performance, and stability across physical, virtual, and cloud environments.

### Register system

URL [Default](#)

Proxy  Use proxy server

Login

Password

Activation Key

Organization

Insights  Connect this system to [Red Hat Insights](#).

[Cancel](#) [Register](#)

27. Navigate to localhost > Terminal to display the CLI. Optionally you can use any SSH client to log in to the RHV-H CLI. Confirm that the required subscription is attached, and then enable the Red Hat Virtualization Host 7 repository to allow further updates and make sure that all other repositories are disabled.

```

# subscription-manager list
+-----+
      Installed Product Status
+-----+
Product Name: Red Hat Virtualization Host
Product ID: 328
Version: 4.3
Arch: x86_64
Status: Subscribed
# subscription-manager repos --disable=*
Repository 'rhel-7-server- rhvh-4-source-rpms' is disabled for this
system.
Repository 'rhvh-4-build-beta-for-rhel-8-x86_64-source-rpms' is disabled
for this system.
Repository 'rhel-7-server- rhvh-4-beta-debug-rpms' is disabled for this
system.
Repository 'rhvh-4-beta-for-rhel-8-x86_64-debug-rpms' is disabled for
this system.
Repository 'jb-eap-textonly-1-for-middleware-rpms' is disabled for this
system.
Repository 'rhvh-4-build-beta-for-rhel-8-x86_64-rpms' is disabled for
this system.
Repository 'rhvh-4-beta-for-rhel-8-x86_64-source-rpms' is disabled for
this system.
Repository 'rhel-7-server- rhvh-4-debug-rpms' is disabled for this
system.
Repository 'rhvh-4-build-beta-for-rhel-8-x86_64-debug-rpms' is disabled
for this system.
Repository 'rhel-7-server- rhvh-4-beta-source-rpms' is disabled for this
system.
Repository 'rhel-7-server- rhvh-4-rpms' is disabled for this system.
Repository 'jb-coreservices-textonly-1-for-middleware-rpms' is disabled
for this system.
Repository 'rhvh-4-beta-for-rhel-8-x86_64-rpms' is disabled for this
system.
Repository 'rhel-7-server- rhvh-4-beta-rpms' is disabled for this
system.
# subscription-manager repos --enable=rhel-7-server- rhvh-4-rpms
Repository 'rhel-7-server- rhvh-4-rpms' is enabled for this system.

```

28. From the console, modify the iSCSI initiator ID to match the one you set in the Element access group previously by running the following command.

```
rhv-h01 # echo InitiatorName=iqn.1994-05.com.redhat:rhv-host-node- 01 >
/etc/iscsi/initiatorname.iscsi
```

29. Enable and restart the iscsid service.

```
# systemctl enable iscsid
Created symlink from /etc/systemd/system/multi-
user.target.wants/iscsid.service to
/usr/lib/systemd/system/iscsid.service
# systemctl start iscsid
# systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
   vendor preset: disabled)
     Active: active (running) since Thu 2020-05-14 16:08:52 EDT; 3 days
ago
   Docs: man:iscsid(8)
         man:iscsiuio(8)
         man:iscsiadm(8)
 Main PID: 5422 (iscsid)
   Status: "Syncing existing session(s)"
  CGroupl: /system.slice/iscsid.service
          └─5422 /sbin/iscsid -f
              ├ 5423 /sbin/iscsid -f
```

30. Install and prepare the other RHV host by repeating the steps 1 to 29.

Next: [5. Deploy the RHV Manager as a Self-Hosted Engine](#)

## 5. Deploy the RHV Manager as a Self-Hosted Engine: NetApp HCI with RHV

This section describes the detailed steps for installing the Red Hat Virtualization Manager as a self-hosted engine. These steps begin after the RHV hosts are registered and the Cockpit GUI is accessible.

1. Log in to the Cockpit GUI of one of the RHV hosts at <https://<HostFQDN/IP>:9090> using the root credentials. Navigate to the Virtualization sub-tab and click Hosted Engine. Then click the Start button below the Hosted Engine content to initiate the engine deployment.

RED HAT VIRTUALIZATION HOST 4.3.9 (EL7.8)

Privileged root

Dashboard Hosted Engine

# RED HAT<sup>®</sup> VIRTUALIZATION

## Hosted Engine Setup

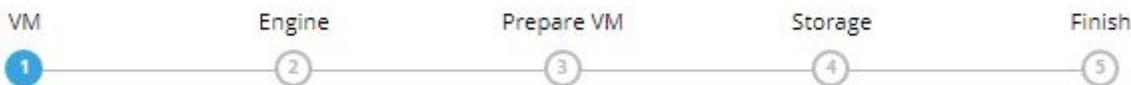
Configure and install a highly-available virtual machine that will run oVirt Engine to manage multiple compute nodes, or add this system to an existing hosted engine cluster.

**Hosted Engine**  
Deploy oVirt hosted engine on storage that has already been provisioned  
[Start](#)

**Hyperconverged**  
Configure Gluster storage and oVirt hosted engine  
[Start](#)

[Getting Started](#) [Installation Guide](#) [More Information](#) [RHV Documentation](#)

2. In the first screen of engine deployment, configure the RHV-M FQDN, network related configuration, root password, and resources for the engine VM (at least 4 CPUs and 16GB memory). Confirm the other configuration settings as required and click Next.



## VM Settings

Engine VM FQDN	<input type="text" value="rhv-m.cie.netapp.com"/>
MAC Address	<input type="text" value="00:16:3e:4e:6b:05"/>
Network Configuration	<input type="button" value="Static"/>
VM IP Address	<input type="text" value="10.63.172.150"/> / <input type="text" value="24"/>
Gateway Address	<input type="text" value="10.63.172.1"/>
DNS Servers	<input type="text" value="10.61.184.251"/>
	<input type="text" value="10.61.184.252"/>
Bridge Interface	<input type="button" value="bond0.1172"/>
Root Password	<input type="password" value="....."/>
Root SSH Access	<input type="button" value="Yes"/>
Number of Virtual CPUs	<input type="text" value="4"/>
Memory Size (MiB)	<input type="text" value="16384"/> 511,548MB available

› Advanced

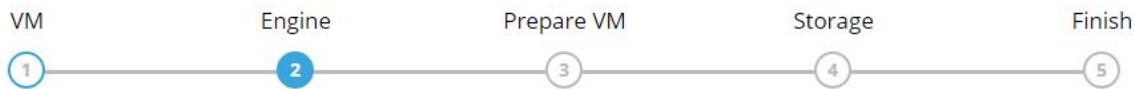


Make sure that the engine VM FQDN is resolvable by the specified DNS servers.

3. In the next screen, enter the admin portal password. Optionally, enter the notification settings for alerts to be sent by email. Then click Next.

## Hosted Engine Deployment

X



### Engine Credentials

Admin Portal Password

### Notification Settings

Server Name

Server Port Number

Sender E-Mail Address

Recipient E-Mail Addresses

[Cancel](#)

[< Back](#)

[Next >](#)

4. In the next screen, review the configuration for the engine VM. If any changes are desired, go back at this point and make them. If the information is correct, click Prepare the VM.



Please review the configuration. Once you click the 'Prepare VM' button, a local virtual machine will be started and used to prepare the management services and their data. This operation may take some time depending on your hardware.

▼ VM

**Engine FQDN:** rhv-m.cie.netapp.com  
**MAC Address:** 00:16:3e:4e:6b:05  
**Network Configuration:** Static  
**VM IP Address:** 10.63.172.150/24  
**Gateway Address:** 10.63.172.1  
**DNS Servers:** 10.61.184.251,10.61.184.252  
**Root User SSH Access:** yes  
**Number of Virtual CPUs:** 4  
**Memory Size (MiB):** 16384  
**Root User SSH Public Key:** (None)  
**Add Lines to /etc/hosts:** yes  
**Bridge Name:** ovirtmgmt  
**Apply OpenSCAP profile:** no

▼ Engine

**SMTP Server Name:** localhost  
**SMTP Server Port Number:** 25  
**Sender E-Mail Address:** root@localhost  
**Recipient E-Mail Addresses:** root@localhost

[Cancel](#)

[◀ Back](#)

[Prepare VM](#)

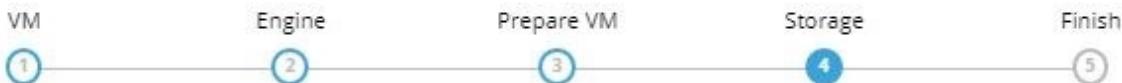
- 
5. The VM installation begins and can take some time to complete as it downloads a machine image and stages the VM locally. After it has completed, it displays the Execution Completed Successfully message. Click Next.



Execution completed successfully. Please proceed to the next step.

[Cancel](#)[< Back](#)[Next >](#)

6. After RHV-M is installed, enter the details of the hosted engine storage domain where it copies the VM from local storage to the shared storage domain to facilitate a high availability engine quorum.
7. Enter the Storage Type as iSCSI, provide the iSCSI portal details, click Retrieve Target List, which fetches the iSCSI target list corresponding to the portal, and select the volume and LUN to be mapped to the hosted engine storage domain. Click Next.



Please configure the storage domain that will be used to host the disk for the management VM. Please note that the management VM needs to be responsive and reliable enough to be able to manage all resources of your deployment, so highly available storage is preferred.

#### Storage Settings

Storage Type	iSCSI
Portal IP Address	172.21.87.140
Portal Port	3260
Portal Username	admin
Portal Password	.....

**Retrieve Target List**

The following targets have been found:

- ④ iqn.2010-01.com.solidfire:nh35.rhv-hostedengine.1, TPGT: 1  
172.21.87.140:3260

The following luns have been found on the requested target:

- ④ ID: 36f47acc1000000006e68333500000003  
Size (GiB): 186.00  
Description: SolidFire SSD SAN  
Status: free  
Number of Paths: 1

› Advanced



If the Hosted Engine setup is unable to discover the storage, open an interactive SSH session to the node and verify that you can reach the SVIP IP address through your node's storage interface. If the network is reachable, you might need to manually discover or log in to the iSCSI LUN intended for the Hosted Engine install.

8. On the next screen, review the storage configuration and, if any changes are desired, go back and make them. If the information is correct, click Finish Deployment. It takes some time as the VM is copied to the storage domain. After deployment is complete, click Close.

## Hosted Engine Deployment



Hosted engine deployment complete!

**Close**

9. The next step is to register and enable the Red Hat Virtualization Manager repositories. Log in to the RHV-M VM with SSH to register it with Subscription Manager.

```
# subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: redhat_user
Password: redhat_password
The system has been registered with ID: 99d06fcb-a3fd74-41230f-bad583-
0ae61264f9a3
The registered system name is: rhv-m.cie.netapp.com
```

10. After registration, list the available subscriptions and record the pool ID for RHV-M.

```
# subscription-manager list --available
<snip>
Subscription Name: Red Hat Virtualization Manager
Provides: Red Hat Beta
Red Hat Enterprise Linux Server
Red Hat CodeReady Linux Builder for x86_64
Red Hat Enterprise Linux for x86_64
Red Hat Virtualization Manager
Red Hat OpenShift Container Platform
Red Hat Ansible Engine
Red Hat Enterprise Linux Fast Datapath
Red Hat JBoss Core Services
JBoss Enterprise Application Platform
SKU: RV00045
Contract:
Pool ID: 8a85f9937a1a2a57c0171a366b5682540112a313 ß Pool ID
Provides Management: No
Available: 6
Suggested: 0
Service Type: L1-L3
Roles:
Service Level: Layered
Usage:
Add-ons:
Subscription Type: Stackable
Starts: 04/22/2020
Ends: 04/21/2021
Entitlement Type: Physical
<snip>
```

## 11. Attach the RHV-M subscription using the recorded pool ID.

```
# subscription-manager attach
--pool=8a85f9937a1a2a57c0171a366b5682540112a313
Successfully attached a subscription for: Red Hat Virtualization Manager
```

## 12. Enable the required RHV-M repositories.

```
# subscription-manager repos \
--disable='*' \
--enable=rhel-7-server-rpms \
--enable=rhel-7-server-supplementary-rpms \
--enable=rhel-7-server-rhv-4.3-manager-rpms \
--enable=rhel-7-server-rhv-4-manager-tools-rpms \
--enable=rhel-7-server-ansible-2-rpms \
--enable=jb-eap-7.2-for-rhel-7-server-rpms

Repository 'rhel-7-server-ansible-2-rpms' is enabled for this system.
Repository 'rhel-7-server-rhv-4-manager-tools-rpms' is enabled for this
system.
Repository 'rhel-7-server-rhv-4.3-manager-rpms' is enabled for this
system.
Repository 'rhel-7-server-rpms' is enabled for this system.
Repository 'jb-eap-7.2-for-rhel-7-server-rpms' is enabled for this
system.
Repository 'rhel-7-server-supplementary-rpms' is enabled for this
system.
```

13. Next, create a storage domain to hold the VM disks or OVF files for all VMs in the same datacenter as that of the hosts.
14. To log into the RHV-M Administrative portal using a browser, log into <https://<ManagerFQDN>/ovirt-engine>, select Administrative Portal, and log in as the `admin @ internal` user.
15. Navigate to Storage > Storage Domains and click New Domain.
16. From the dropdown menu, select Data for the Domain Function, select iSCSI for the Storage Type, select the host to map the volume, enter a name of your choice, confirm that the data center is correct, and then expand the data domain iSCSI target and add the LUN. Click OK to create the domain.

New Domain

Data Center	Default (V5)	Name	data_domain
Domain Function	Data	Description	Data Domain for VMs
Storage Type	iSCSI	Comment	
Host	rhv-h01.cie.netapp.com		

- Discover Targets Login All

Target Name	Address	Port	
iqn.2010-01.com.solidfire:nh35.rhv-hostedengine-1.3	172.21.87.140	3260	<span style="color: green;">→</span>
iqn.2010-01.com.solidfire:nh35.rhv-hostedengine.1	172.21.87.140	3260	<span style="color: green;">→</span>
iqn.2010-01.com.solidfire:nh35.data-domain.5	172.21.87.140	3260	<span style="color: green;">→</span>

LUN ID      Size      #path      Vendor ID      Product ID      Serial      Add

36f47acc100000006e68333500000005	1430 GiB	1	SolidFir	SSD SAN	SSolidFirSSD_SAN_6e68333500000005	<span style="background-color: #ccc; border: 1px solid #ccc; padding: 2px;">Add</span>
----------------------------------	----------	---	----------	---------	-----------------------------------	--

↳ Advanced Parameters

OK Cancel



If the Hosted Engine setup is unable to discover the storage, you might need to manually discover or log in to the iSCSI LUN intended for the data domain.

17. Add the second host to the hosted engine quorum. Navigate to Compute > Hosts and click New. In the New Host pane, select the appropriate cluster, provide the details of the second host, and check the Activate Host After Install checkbox.

New Host X

**General** >

Host Cluster Default ▼  
*Data Center: Default*

Use Foreman/Satellite

Name

Comment

Hostname/IP i

SSH Port

Activate host after install

**Authentication**

User Name

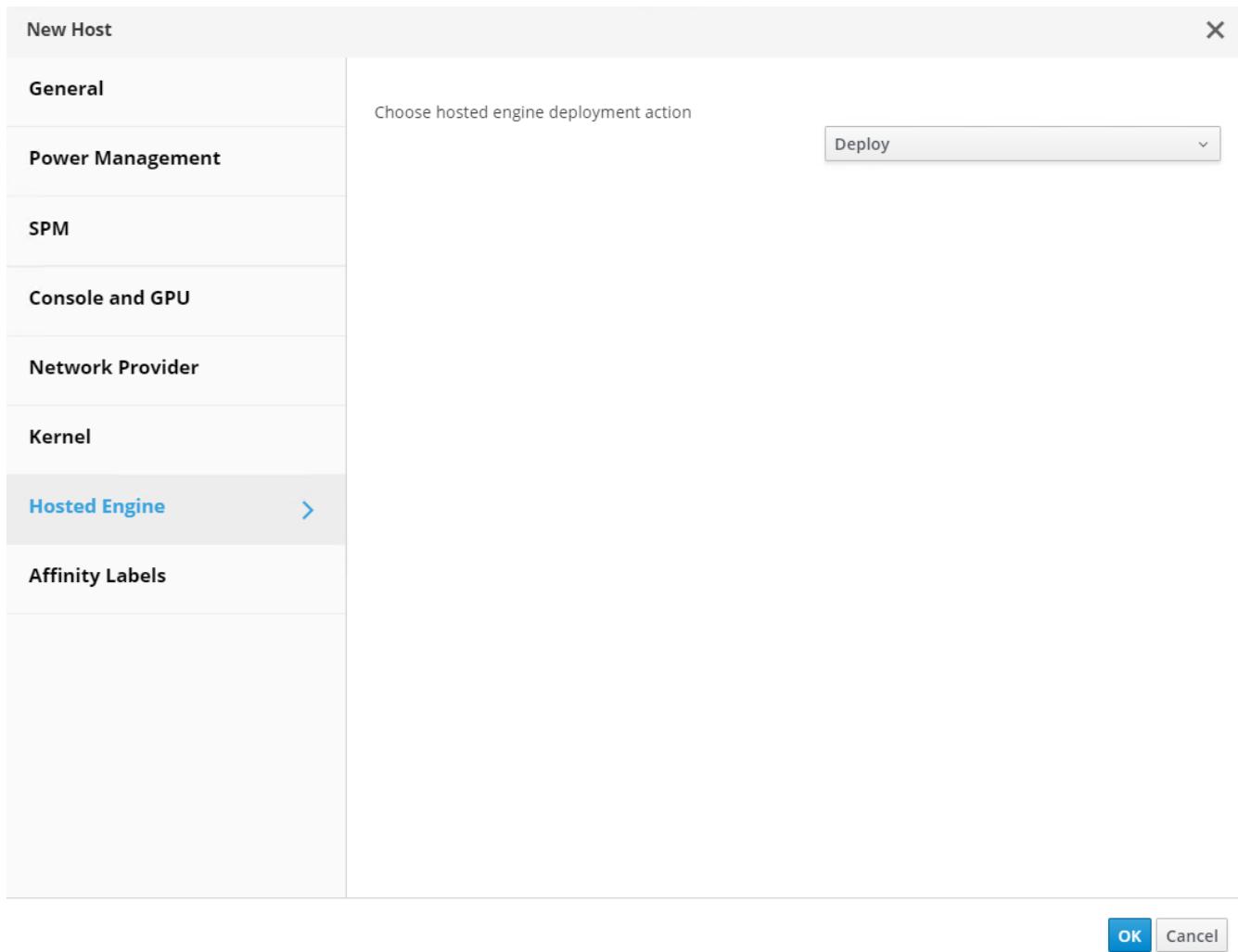
Password

SSH Public Key

Advanced Parameters

OK Cancel

18. Click the Hosted Engine sub-tab in the New Host pane dropdown and select Deploy from the hosted engine deployment action. Click OK to add the host to the quorum. This begins the installation of the necessary packages to support the hosted engine and activate the host. This process might take a while.



19. Next, create a storage virtual network for hosts. Navigate to Network > Networks and click New. Enter the name of your choice, enable VLAN tagging, and enter the VLAN ID for the Storage network. Confirm that the VM Network checkbox is checked and that the MTU is set to 9000. Go to the Cluster sub-tab and make sure that Attach and Require are checked. Then click OK to create the storage network.

New Logical Network

**General**

Data Center	<input type="text" value="Default"/>
Name <small>i</small>	<input type="text" value="storagenet"/>
Description	<input type="text"/>
Comment	<input type="text"/>
<b>Network Parameters</b>	
Network Label	<input type="text"/>
<input checked="" type="checkbox"/> Enable VLAN tagging	<input type="text" value="3343"/>
<input checked="" type="checkbox"/> VM network <small>vm</small>	
MTU	<input type="radio"/> Default (1500) <input checked="" type="radio"/> Custom <input type="text" value="9000"/>
Host Network QoS	<input type="text" value="Unlimited"/>

**vNIC Profiles**

**OK** **Cancel**

20. Assign the storage logical network to the second host in the cluster or to whichever host is not currently hosting the hosted engine VM.
21. Navigate to Compute > Hosts, and click the host that has silver crown in the second column. Then navigate to the Network Interfaces sub-tab, click Setup Host Networks, and drag and drop the storage logical network into the Assigned Logical Networks column to the right of bond0.

## Setup Host rhv-h02.cie.netapp.com Networks

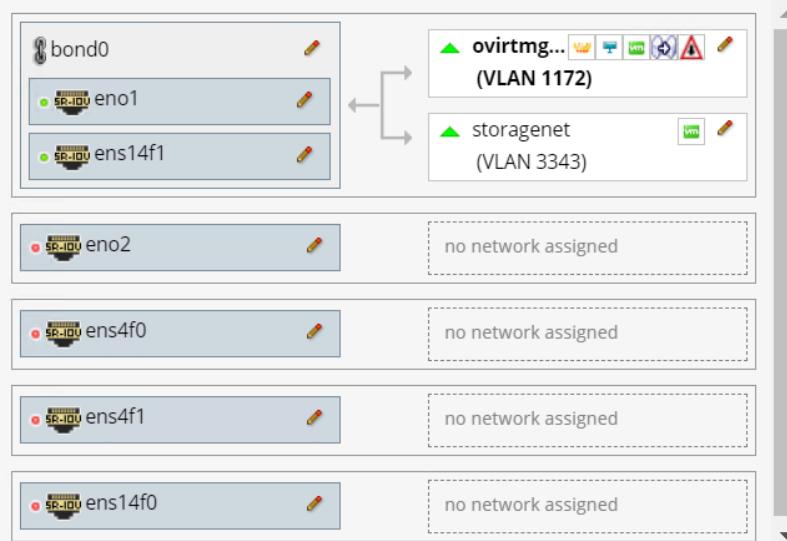
X

Drag to make changes

Interfaces

Assigned Logical Networks

Networks Labels



Unassigned Logical Networks

Required

Non Required

External Logical Networks i

- Verify connectivity between Host and Engine i
- Save network configuration i

**OK** Cancel

22. Click the pen symbol on the storage network interface under bond0. Configure the IP address and the netmask, and then click OK. Click OK again in the Setup Host Networks pane.

Edit Network storagenet

**IPv4**

Sync network [i](#)

**IPv6**

**QoS**

**Custom Properties**

**DNS Configuration**

Boot Protocol  
 None  
 DHCP  
 Static

IP

Netmask / Routing Prefix

Gateway

**OK** **Cancel**

23. Migrate the hosted engine VM to the host that was just configured so that the storage logical network can be configured on the second host. Navigate to Compute > Virtual Machines, click HostedEngine and then click Migrate. Select the second host from the dropdown menu Destination Host and click Migrate.

Migrate VM(s)

Select a host to migrate 1 virtual machine(s) to:

**Destination Host** [i](#)

**Migrate VMs in Affinity** [i](#)  Migrate all VMs in positive enforcing affinity with selected VMs.

**Virtual Machines** HostedEngine

**Cancel** **Migrate**

After the migration is successful and the hosted engine VM is migrated to the second host, repeat steps 21 and 22 for the host that currently possesses the silver crown.

24. After you have completed this process, you should see that both the hosts are up. One of the hosts has a golden crown, indicating that it is hosting the hosted engine VM, and the other host has a silver crown indicating that it is capable of hosting the hosted engine VM.

The screenshot shows the Red Hat Virtualization web interface. The left sidebar has 'Compute' selected. The main area is titled 'Compute > Hosts'. It features a search bar and buttons for 'New' and 'Edit'. A table lists two hosts: 'rhv-h01.cie.netapp.com' and 'rhv-h02.cie.netapp.com'. Both hosts are marked with green up arrows and have their names, hostnames/IPs, clusters, data centers, and statuses listed.

Next: 6. Configure RHV-M Infrastructure

## 6. Configure RHV-M Infrastructure: NetApp HCI with RHV

To configure the RHV-M infrastructure, complete the following steps:

1. By default, the ovirtmgmt network is used for all purposes, including the migration of VMs and virtual guest data.
2. It is a best practice to specify different networks for these purposes. To configure the migration network, navigate to Network > Networks and click New. Enter the name of your choice, enable VLAN tagging, and enter the VLAN ID for the migration network.
3. Make sure that the VM Network checkbox is unchecked. Go to the Cluster sub-tab and make sure that Attach and Require are checked. Then click OK to create the network.

**New Logical Network**

**General**

Data Center	Default
Name <small>i</small>	migration_net
Description	
Comment	

**Cluster**

**Network Parameters**

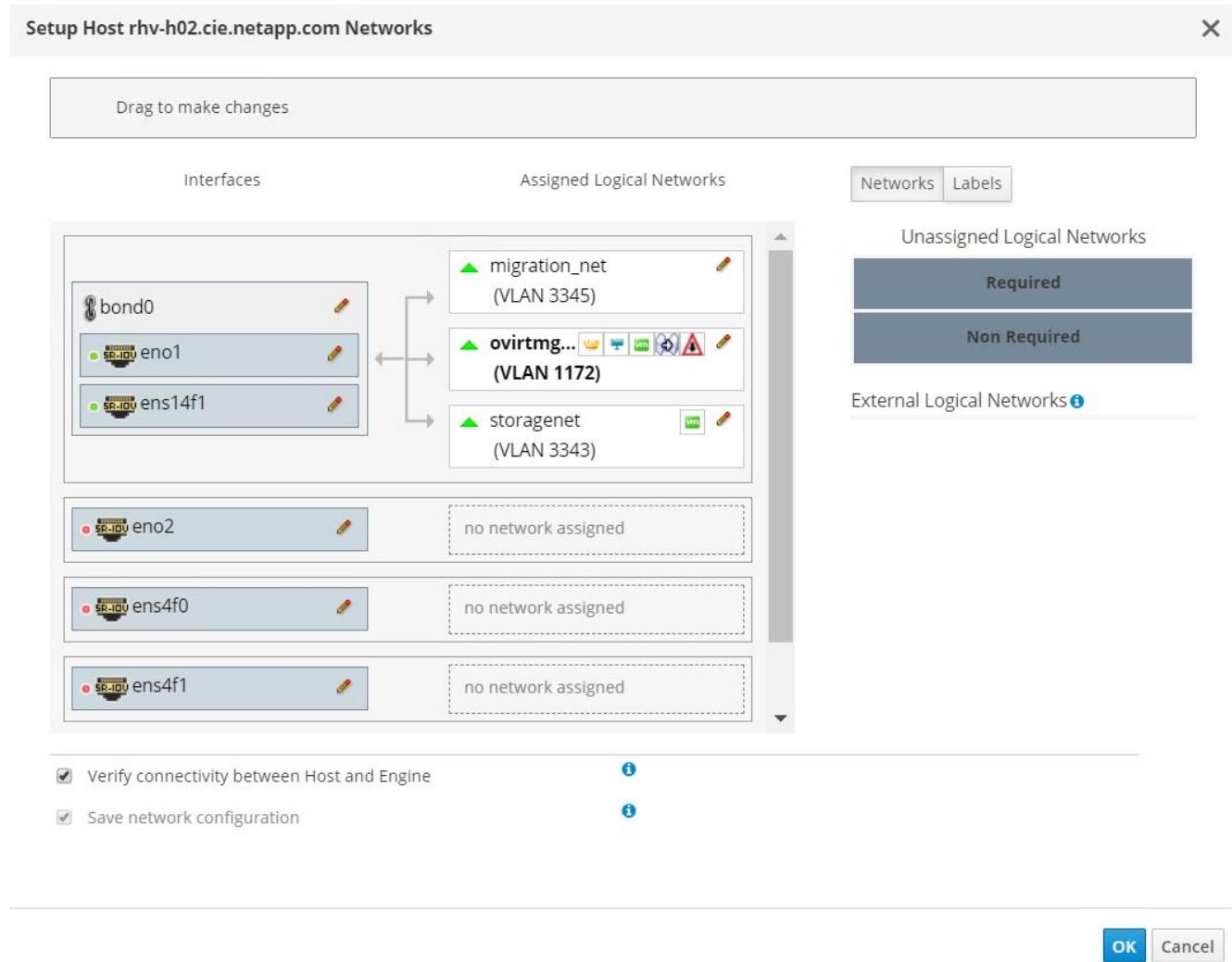
Network Label	
Enable VLAN tagging	<input checked="" type="checkbox"/> 3345
VM network	<input type="checkbox"/>
MTU	<input type="radio"/> Default (1500) <input type="radio"/> Custom <input type="text"/>
Host Network QoS	[Unlimited]

**OK** **Cancel**

4. To assign the migration logical network to both the hosts, navigate to Compute > Hosts, click the hosts, and

navigate to the Network Interfaces sub-tab.

5. Then click Setup Host Networks and drag and drop the migration logical network into the Assigned Logical Networks column to the right of bond0.



6. Click the pen symbol on the migration network interface under bond0. Configure the IP address details and click OK. Then click OK again in the Setup Host Networks pane.

Edit Network migration\_net

**IPv4**

Sync network i

**IPv6**

**QoS**

**Custom Properties**

**DNS Configuration**

Boot Protocol  
 None  
 DHCP  
 Static

IP: 172.21.89.10

Netmask / Routing Prefix: 24

Gateway:

**OK** **Cancel**

7. Repeat steps 4 through 6 for the other host as well.
8. The newly created network must be assigned the role of the migration network. Navigate to Compute > Clusters and click the cluster that the RHV hosts belong to, click the Logical Networks sub-tab, and click Manage Networks. For the migration network, enable the checkbox under Migration Network column. Click OK.

Manage Networks

Name	<input checked="" type="checkbox"/> Assign All	<input checked="" type="checkbox"/> Require All	VM Network	Management	Display Network	Migration Network
ovirtmgmt	<input checked="" type="checkbox"/> Assign	<input checked="" type="checkbox"/> Require		<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
migration_net	<input checked="" type="checkbox"/> Assign	<input checked="" type="checkbox"/> Require		<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
storageenet	<input checked="" type="checkbox"/> Assign	<input checked="" type="checkbox"/> Require		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. Next, as a best practice, create a separate VM network rather than using the ovirtmgmt network for VMs.
10. Navigate to Network > Networks and click New. Enter the name of your choice, enable VLAN tagging, and enter the VLAN ID for the VM guest network. Make sure that the checkbox VM Network is checked. Go to the Cluster's sub-tab and make sure that Attach and Require are checked. Then click OK to create the VM guest network.

New Logical Network

General >

Cluster

vNIC Profiles

Data Center Default

Name vGuest

Description

Comment

**Network Parameters**

Network Label

Enable VLAN tagging 3346

VM network

MTU  Default (1500)  Custom

Host Network QoS [Unlimited]

OK Cancel

- Assign the VM guest logical network to both the hosts. Navigate to Compute > Hosts, click the host names and navigate to the Network Interfaces sub-tab. Then click Setup Host Networks and drag and drop the VM guest logical network into the Assigned Logical Networks column to the right of bond0. There is no need to assign an IP to this logical network, because it provides passthrough networking for the VMs.

The VM guest network should be able to reach the internet to allow guests to register with Red Hat Subscription Manager.

Next: [7. Deploy the NetApp mNode](#)

## 7. Deploy the NetApp mNode: NetApp HCI with RHV

The management node (mNode) is a VM that runs in parallel with one or more Element software-based storage clusters. It is used for the following purposes:

- Providing system services including monitoring and telemetry
- Managing cluster assets and settings
- Running system diagnostic tests and utilities
- Enabling callhome for NetApp ActiveIQ for additional support

To install the NetApp mNode on Red Hat Virtualization, complete the following steps:

- Upload the mNode ISO as a disk to the storage domain. Navigate to Storage > Disks > Upload and click Start. Then click Upload Image and select the downloaded mNode ISO image. Verify the storage domain, the host to perform the upload, and additional details. Then click OK to upload the image to the domain. A progress bar indicates when the upload is complete and the ISO is usable.

2. Create a VM disk by navigating to Storage > Disks and click New. The mNode disk must be at least 400 GB in size but can be thin-provisioned. In the wizard, enter the name of your choice, select the proper data center, make sure that the proper storage domain is selected, select Thin Provisioning for the allocation policy, and check the Wipe After Delete checkbox. Click OK.

**New Virtual Disk**

Image		Direct LUN	Cinder	Managed Block
Size (GiB)	400			
Alias	mNode_disk			
Description				
Data Center	Default			
Storage Domain	data_domain (1784 GiB free of 1907 GiB)			
Allocation Policy	Thin Provision			
Disk Profile	data_domain			

3. Next, navigate to Compute > Virtual Machines and click New. In the General sub-tab, select the appropriate cluster, enter the name of your choice, click attach, and select the disk created in the previous step. Check the box below OS to emphasize that it is a bootable drive. Click OK.

**Attach Virtual Disks**

Image		Direct LUN	Cinder	Managed Block							
<input checked="" type="radio"/>	mNode_disk		0438434a-9...	400 GiB	1 GiB	data_domain	VirtIO ▾	R/O	<input checked="" type="checkbox"/> OS	<input checked="" type="checkbox"/> Boot	

4. Select ovirtmgmt from the dropdown for nic1. Click the (+) sign and select the storage network interface from the dropdown list for nic2.

New Virtual Machine X

General >	
System	Cluster <input type="text" value="Default"/> <small>Data Center: Default</small>
Initial Run	Template <input type="text" value="Blank   (0)"/>
Console	Operating System <input type="text" value="Other OS"/>
Host	Instance Type <input type="text" value="Custom"/> <small>Custom</small>
High Availability	Optimized for <input type="text" value="Server"/>
Resource Allocation	Name <input type="text" value="NetApp mNode"/>
Boot Options	Description <input type="text"/>
Random Generator	Comment <input type="text"/>
Custom Properties	VM ID <input type="text"/>
Icon	<input type="checkbox"/> Stateless <input type="checkbox"/> Start in Pause Mode <input type="checkbox"/> Delete Protection
Foreman/Satellite	Instance Images mNode_disk: (400 GB) attaching (boot) <span style="float: right;"><input type="button" value="Edit"/> <input type="button" value="+"/> <input type="button" value="-"/></span>
Affinity Labels	Instantiate VM network interfaces by picking a vNIC profile. nic1 <input type="text" value="ovirtmgmt/ovirtmgmt"/> <input type="button" value="-"/> nic2 <input type="text" value="storagenet/storagenet"/> <input type="button" value="+"/> <input type="button" value="-"/>

Hide Advanced Options

- Click the System sub-tab and make sure that it has at least 12GB of memory and 6 virtual CPUs as recommended.

New Virtual Machine X

<b>General</b>	Cluster <input type="text" value="Default"/> <span style="font-size: small;">Data Center: Default</span>
<b>System</b> >	Template <input type="text" value="Blank   (0)"/>
<b>Initial Run</b>	Operating System <input type="text" value="Other OS"/>
<b>Console</b>	Instance Type <input type="text" value="Custom"/> <span style="font-size: small;">Custom</span>
<b>Host</b>	Optimized for <input type="text" value="Server"/>
<b>High Availability</b>	Memory Size <input type="text" value="12288 MB"/>
<b>Resource Allocation</b>	Maximum memory <input type="text" value="49152 MB"/>
<b>Boot Options</b>	Physical Memory Guaranteed <input type="text" value="12288 MB"/>
<b>Random Generator</b>	Total Virtual CPUs <input type="text" value="6"/>
<b>Custom Properties</b>	<input type="checkbox"/> Advanced Parameters
<b>General</b>	Hardware Clock Time Offset <input type="text" value="default: (GMT+00:00) GMT Standard Time"/>
<b>Icon</b>	<input type="checkbox"/> Provide custom serial number policy
<b>Foreman/Satellite</b>	
<b>Affinity Labels</b>	

Hide Advanced Options **OK** **Cancel**

6. Click the Boot Options sub-tab, select CD-ROM as the first device in the boot sequence, select Hard Drive as the second device. Enable Attach CD and attach the mNode ISO. Then click OK.

New Virtual Machine X

<b>General</b>	Cluster <input type="text" value="Default"/> <small>Data Center: Default</small>	
<b>System</b>	Template <input type="text" value="Blank   (0)"/>	
<b>Initial Run</b>	Operating System <input type="text" value="Other OS"/>	
<b>Console</b>	Instance Type <input type="text" value="Custom"/>	
<b>Host</b>	Optimized for <input type="text" value="Server"/>	
<b>High Availability</b>	Boot Sequence:	
	First Device <input type="text" value="CD-ROM"/>	
	Second Device <input type="text" value="Hard Disk"/>	
<b>Boot Options</b> >	<input checked="" type="checkbox"/> Attach CD <input type="text" value="solidfire-fdva-sodium-patch5-11.5.0."/> <span style="font-size: small; color: blue;">e</span>	
<b>Random Generator</b>	<input type="checkbox"/> Enable menu to select boot device	
<b>Custom Properties</b>		
<b>Icon</b>		
<b>Foreman/Satellite</b>		
<b>Affinity Labels</b>		

Hide Advanced Options **OK** **Cancel**

The VM is created.

- After the VM becomes available, power it on, and open a console to it. It begins to load the NetApp Solidfire mNode installer. When the installer is loaded, you are prompted to start the RTFI magnesium installation; type `yes` and press Enter. The installation process begins, and after it is complete, it automatically powers off the VM.



.....

Starting SolidFire RTFI magnesium

Proceed (Yes,No)

yes

8. Next, click the mNode VM and click Edit. In the Boot Options sub-tab, uncheck the Attach CD checkbox and click the OK button.

**Edit Virtual Machine**

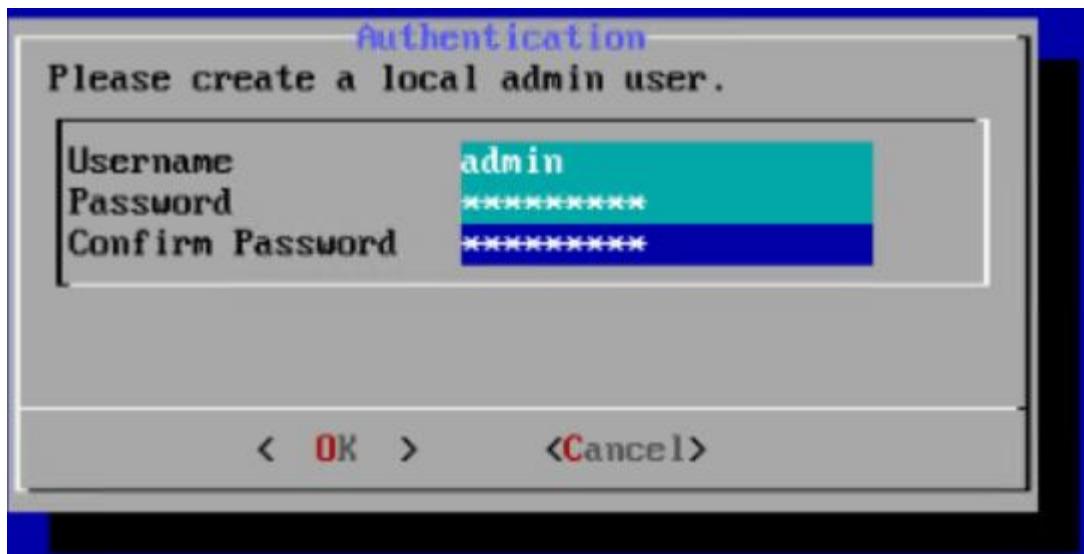
<b>General</b>	Cluster	<input type="text" value="Default"/>
	Data Center: Default	
<b>System</b>	Template	<input type="text" value="Blank   (0)"/>
<b>Initial Run</b>	Operating System	<input type="text" value="Other OS"/>
<b>Console</b>	Instance Type	<input type="text" value="Custom"/>
<b>Host</b>	Optimized for	<input type="text" value="Server"/>
<b>High Availability</b>	<b>Boot Sequence:</b>	
	First Device	<input type="text" value="CD-ROM"/>
	Second Device	<input type="text" value="Hard Disk"/>
<b>Resource Allocation</b>	<input type="checkbox"/> Attach CD	<input type="text" value="solidfire-fdva-magnesium-12.0.0.333"/>
<b>Boot Options</b> >	<input type="checkbox"/> Enable menu to select boot device	
<b>Random Generator</b>		
<b>Custom Properties</b>		
<b>Icon</b>		
<b>Foreman/Satellite</b>		
<b>Affinity Labels</b>		

**Hide Advanced Options** **OK** **Cancel**

- Power on the mNode VM. Using the terminal user interface (TUI), create a management node admin user.



To move through the menu options, press the Up or Down arrow keys. To move through the buttons, press Tab. To move from the buttons to the fields, press Tab. To navigate between fields, press the Up or Down arrow keys.



- After the user is created, you are returned to a login screen. Log in with the credentials that were just created.
- To configure the network interfaces starting with the management interface, navigate to Network > Network Config > eth0 and enter the IP address, netmask, gateway, DNS servers, and search domain for your environment. Click OK.

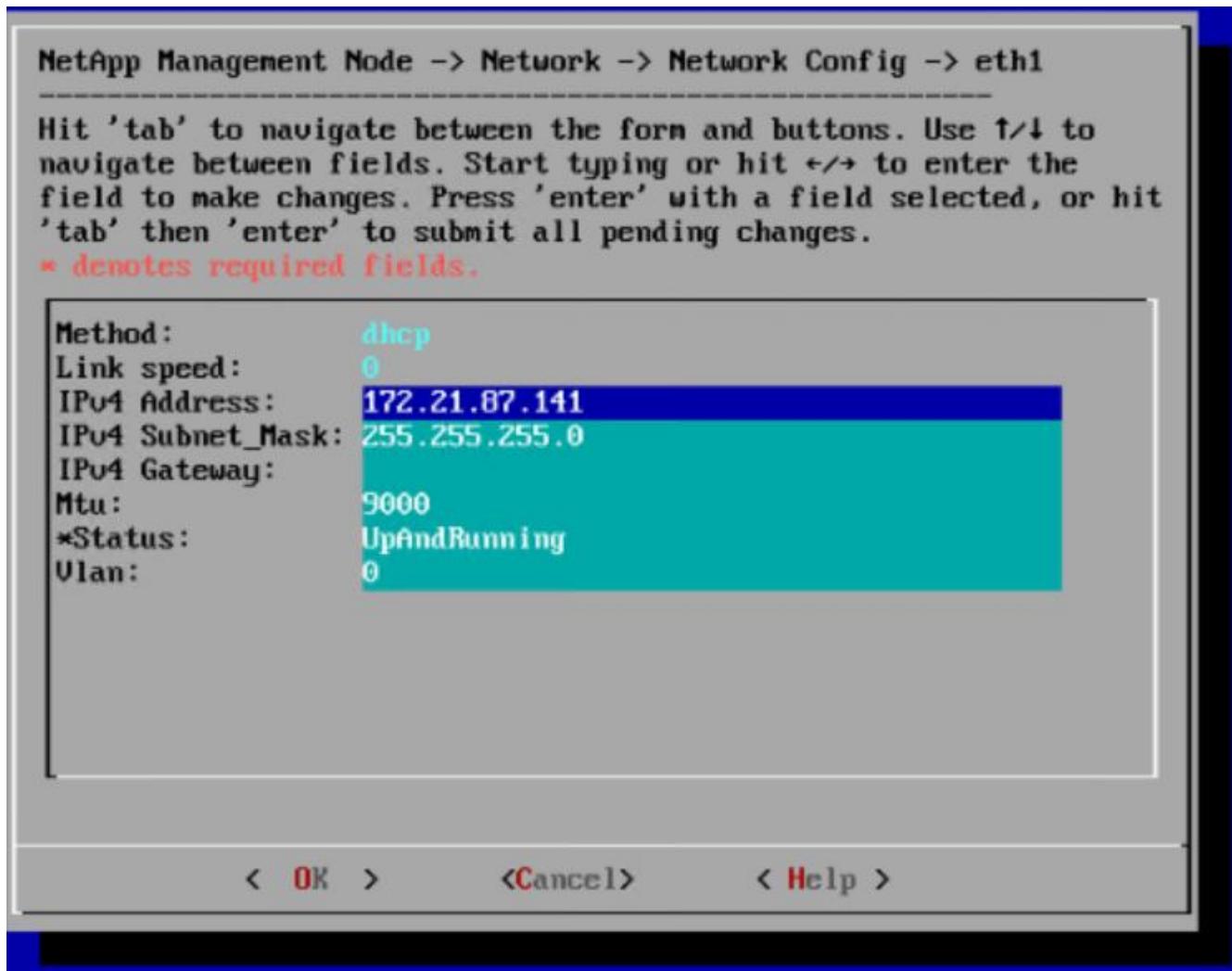
NetApp Management Node -> Network -> Network Config -> eth0

Hit 'tab' to navigate between the form and buttons. Use ↑↓ to navigate between fields. Start typing or hit ←→ to enter the field to make changes. Press 'enter' with a field selected, or hit 'tab' then 'enter' to submit all pending changes.  
\* denotes required fields.

Method:	static
Link speed:	0
*IPv4 Address:	10.63.172.141
*IPv4 Subnet_Mask:	255.255.255.0
IPv4 Gateway:	10.63.172.1
Mtu:	1500
Dns:	10.61.184.251, 10.61.184.252
Domains:	cie.netapp.com
IPv6 Address:	
IPv6 Gateway:	
*Status:	UpAndRunning
Ulan:	0

< OK >      <Cancel>      < Help >

12. Next, configure eth1 to access the storage network. Navigate to Network > Network Config > eth1 and enter the IP address and netmask. Verify that the MTU is 9000. Then click OK.



You can now close the TUI interface.

13. SSH into the management node using the management IP, escalate to root and register the mNode with the HCI storage cluster.

```
admin@SF-3D1C ~ $ sudo su

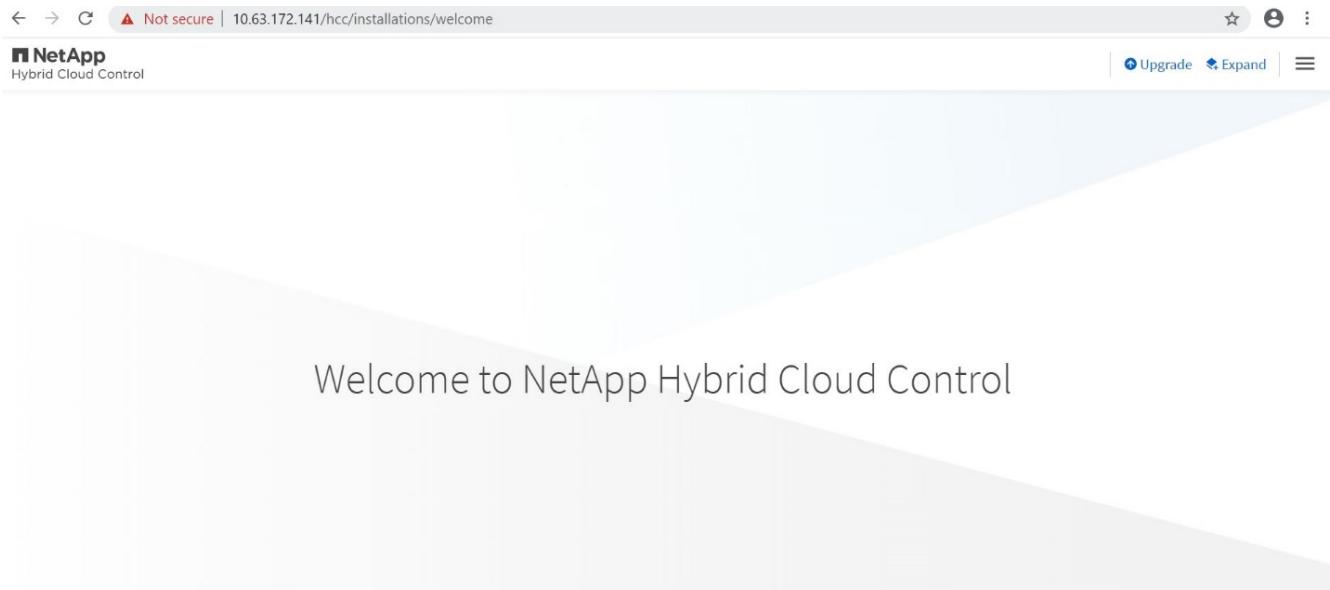
SF-3D1C /home/admin # /sf/packages/mnode/setup-mnode --mnode_admin_user
admin --storage_mvip 10.63.172.140 --storage_username admin
--telemetry_active true

Enter the password for storage user admin:
Enter password for mNode user admin:
[2020-05-21T17:19:53.281657Z]:[setup_mnode:296] INFO:Starting mNode
deployment
[2020-05-21T17:19:53.286153Z]:[config_util:1313] INFO:No previously
running mNode. Continuing with deployment.
```

```
[2020-05-21T17:19:53.286687Z]:[config_util:1320] INFO:Validating
credentials for mNode host.
[2020-05-21T17:19:53.316270Z]:[config_util:1232] INFO:Checking Cluster
information.
[2020-05-21T17:19:53.380168Z]:[config_util:112] INFO:Cluster credentials
verification successful.
[2020-05-21T17:19:53.380665Z]:[config_util:1252] INFO:Cluster version
check successful.
[2020-05-21T17:19:53.458271Z]:[config_util:112] INFO:Successfully
queried system configuration
[2020-05-21T17:19:53.463611Z]:[config_util:497] INFO:CIDR range
172.16.0.0/22 open. Using for docker ingress.
[2020-05-21T17:19:53.464179Z]:[mnodecfg:141] INFO:Configuring mNode
[2020-05-21T17:19:53.464687Z]:[config_util:194] INFO:Wait for ping of
127.0.0.1 to succeed
[2020-05-21T17:19:53.475619Z]:[mnodecfg:145] INFO:Validating the
supplied MNode network configuration
[2020-05-21T17:19:53.476119Z]:[mnodecfg:155] INFO:Testing the MNode
network configuration
[2020-05-21T17:19:53.476687Z]:[config_util:353] INFO:Testing network
connection to storage MVIP: 10.63.172.140
[2020-05-21T17:19:53.477165Z]:[config_util:194] INFO:Wait for ping of
10.63.172.140 to succeed
[2020-05-21T17:19:53.488045Z]:[config_util:356] INFO:Successfully
reached storage MVIP: 10.63.172.140
[2020-05-21T17:19:53.488569Z]:[mnodecfg:158] INFO:Configuring MNode
storage (this can take several minutes)
[2020-05-21T17:19:57.057435Z]:[config_util:536] INFO:Configuring MNode
storage succeeded.
[2020-05-21T17:19:57.057938Z]:[config_util:445] INFO:Replacing default
ingress network.
[2020-05-21T17:19:57.078685Z]:[mnodecfg:163] INFO:Extracting services
tar (this can take several minutes)
[2020-05-21T17:20:36.066185Z]:[config_util:1282] INFO:Extracting
services tar succeeded
[2020-05-21T17:20:36.066808Z]:[mnodecfg:166] INFO:Configuring MNode
authentication
[2020-05-21T17:20:36.067950Z]:[config_util:1485] INFO:Updating element-
auth configuration
[2020-05-21T17:20:41.581716Z]:[mnodecfg:169] INFO:Deploying MNode
services (this can take several minutes)
[2020-05-21T17:20:41.810264Z]:[config_util:557] INFO:Deploying MNode
services succeeded
[2020-05-21T17:20:41.810768Z]:[mnodecfg:172] INFO:Deploying MNode Assets
[2020-05-21T17:20:42.162081Z]:[config_util:122] INFO:Retrying 1/45
time...
```

```
[2020-05-21T17:20:42.162640Z] :[config_util:125] INFO:Waiting 10 seconds before next attempt.  
[2020-05-21T17:20:52.199224Z] :[config_util:112] INFO:Mnode is up!  
[2020-05-21T17:20:52.280329Z] :[config_util:112] INFO:Root asset created.  
[2020-05-21T17:20:52.280859Z] :[config_util:122] INFO:Retrying 1/5 time...  
[2020-05-21T17:20:52.281280Z] :[config_util:125] INFO:Waiting 10 seconds before next attempt.  
[2020-05-21T17:21:02.299565Z] :[config_util:112] INFO:Successfully queried storage assets  
[2020-05-21T17:21:02.696930Z] :[config_util:112] INFO:Storage asset created.  
[2020-05-21T17:21:03.238455Z] :[config_util:112] INFO:Storage asset registered.  
[2020-05-21T17:21:03.241966Z] :[mnodedcfg:175] INFO:Attempting to set up VCP-SIOC credentials  
[2020-05-21T17:21:03.242659Z] :[config_util:953] INFO>No VCP-SIOC credential given from NDE. Using default credentials for VCP-SIOC service.  
[2020-05-21T17:21:03.243117Z] :[mnodedcfg:185] INFO:Configuration Successfully Completed
```

14. Using a browser, log into the management node GUI using <https://<mNodeIP>>. mNode or Hybrid Cloud Control facilitates expansion, monitoring, and upgrading the Element cluster.



15. Click the three parallel lines on the top right and click View Active IQ. Search for the HCI storage cluster by filtering the cluster name and make sure that it is logging the most recent updates.

[Next: Best Practices - Updating RHV Manager and RHV-H Hosts](#)

## Best Practices for Production Deployments

### Updating RHV Manager and RHV-H Hosts: NetApp HCI with RHV

It is a recommended best practice to make sure that both the RHV Manager and the RHV-H hosts have the latest security and stability updates applied to make sure that the environment is protected and continues to run as expected. To apply the updates to the hosts in the deployment, they must first be subscribed to either the Red Hat Content Delivery Network or a local Red Hat Satellite repository. The tasks involved in updating the platform include updating the manager VM and afterward updating each physical host non-disruptively after ensuring virtual guests are migrated to another node in the cluster.

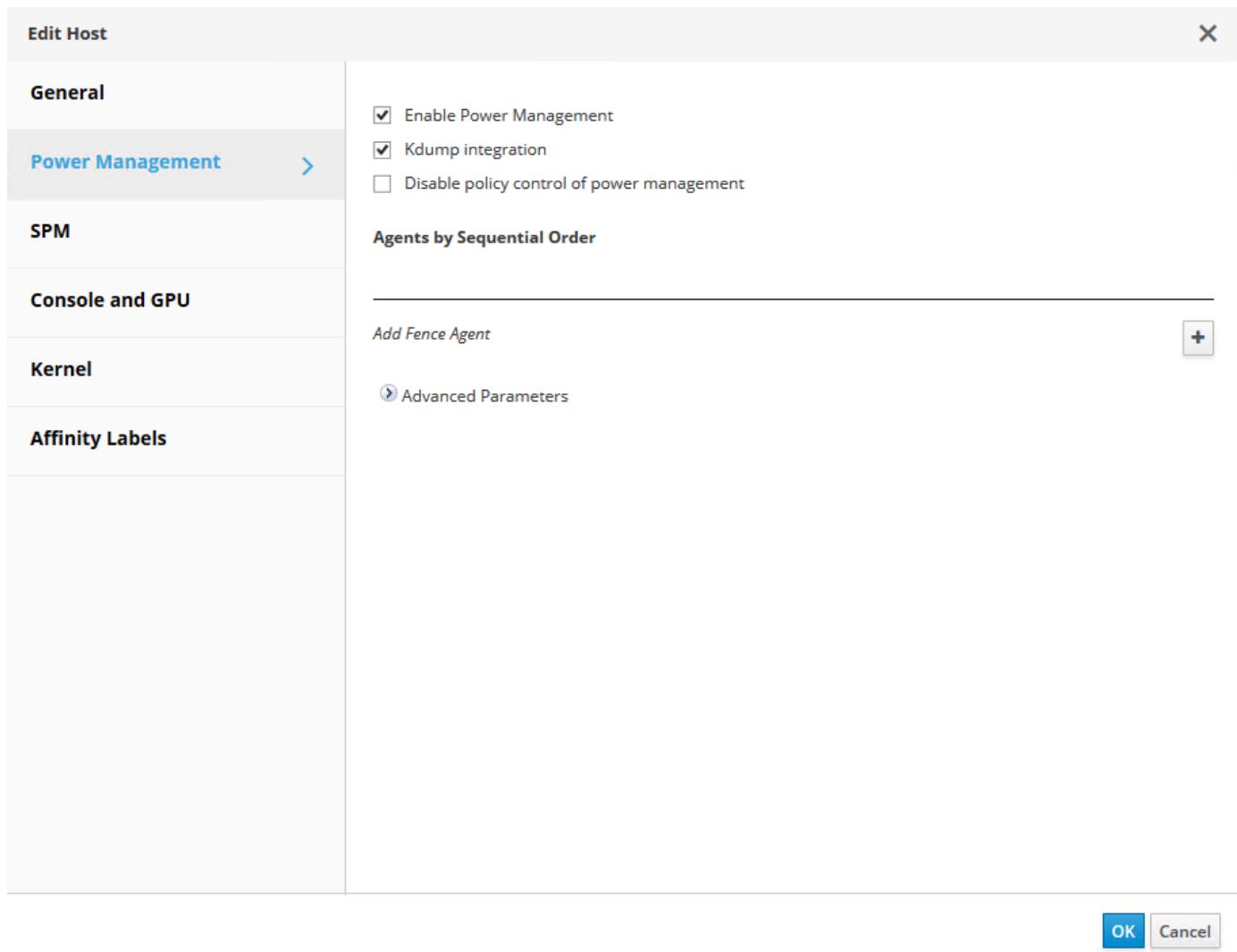
Official documentation to support the upgrade of RHV 4.3 between minor releases can be found [here](#).

[Next: Best Practices - Enabling Fencing for RHV-H Hosts](#)

### Enabling Fencing for RHV-H Hosts: NetApp HCI with RHV

Fencing is a process by which the RHV Manager can provide high availability of the VMs in the environment by automatically shutting down a non-responsive hypervisor host. It does this by sending commands to a fencing agent, which in the case of NetApp HCI is available through the IPMI out-of-band management interface on the compute nodes and rebooting the host. This action releases the locks that the non-responsive hypervisor node has on VM disks and allows for those virtual guests to be restarted on another node in the cluster without risking data corruption. After the host completes its boot process, it automatically attempts to rejoin the cluster it was a part of prior to the shutdown. If it is successful, it is once again allowed to host VMs.

To enable fencing, each host must have power management enabled; this can be found by highlighting the host and clicking the Edit button in the upper right-hand corner or by right-clicking on the host and selecting Edit.



After power management is enabled, the next step involves configuring a fencing agent. Click on the plus sign (+) near the Add Fence Agent, and a new window pops up that must be filled out with the information for the IPMI connection on the NetApp HCI compute nodes. The type of connection is IPMILAN, and the agent needs the IP address, username, and password for the console login. After you have provided this information, you can click test to validate the configuration. If properly configured, it should report the current power status of the node.

Edit fence agent

X

Address	<input type="text" value="172.16.14.31"/>
User Name	<input type="text" value="ADMIN"/>
Password	<input type="password" value="*****"/>
Type	<input style="width: 100px;" type="text" value="ipmilan"/> ▼
Options	<input type="text"/>

Please use a comma-separated list of 'key=value'

Test successful: power on

---

With fencing enabled, the RHV environment is configured to support a highly available deployment should one of the hypervisor nodes become nonresponsive.

[Next: Best Practices - Optimizing Memory for Red Hat Virtualization](#)

#### Optimizing Memory for Red Hat Virtualization: NetApp HCI with RHV

One of the primary benefits for deploying a virtual infrastructure is to enable the more efficient use of physical resources in the environment. In a case in which the guest VMs underutilize the memory allotted, you can use memory overcommitment to optimize memory usage. With this feature, the sum of the memory allocated to guest VMs on a host is allowed to exceed the amount of physical memory on that host.

The concept behind memory overcommitment is similar to thin provisioning of storage resources. At any given moment, every VM on the host does not use the total amount of memory allocated to it. When one VM has excess memory, its unused memory is available for other VMs to use. Therefore, an end user can deploy more VMs than the physical infrastructure would normally allow. Memory overcommitment on the hosts in the cluster is handled by Memory Overcommit Manager (MoM). Techniques like memory ballooning and Kernel Same-page Merging (KSM) can improve memory overcommitment depending on the kind of workload.

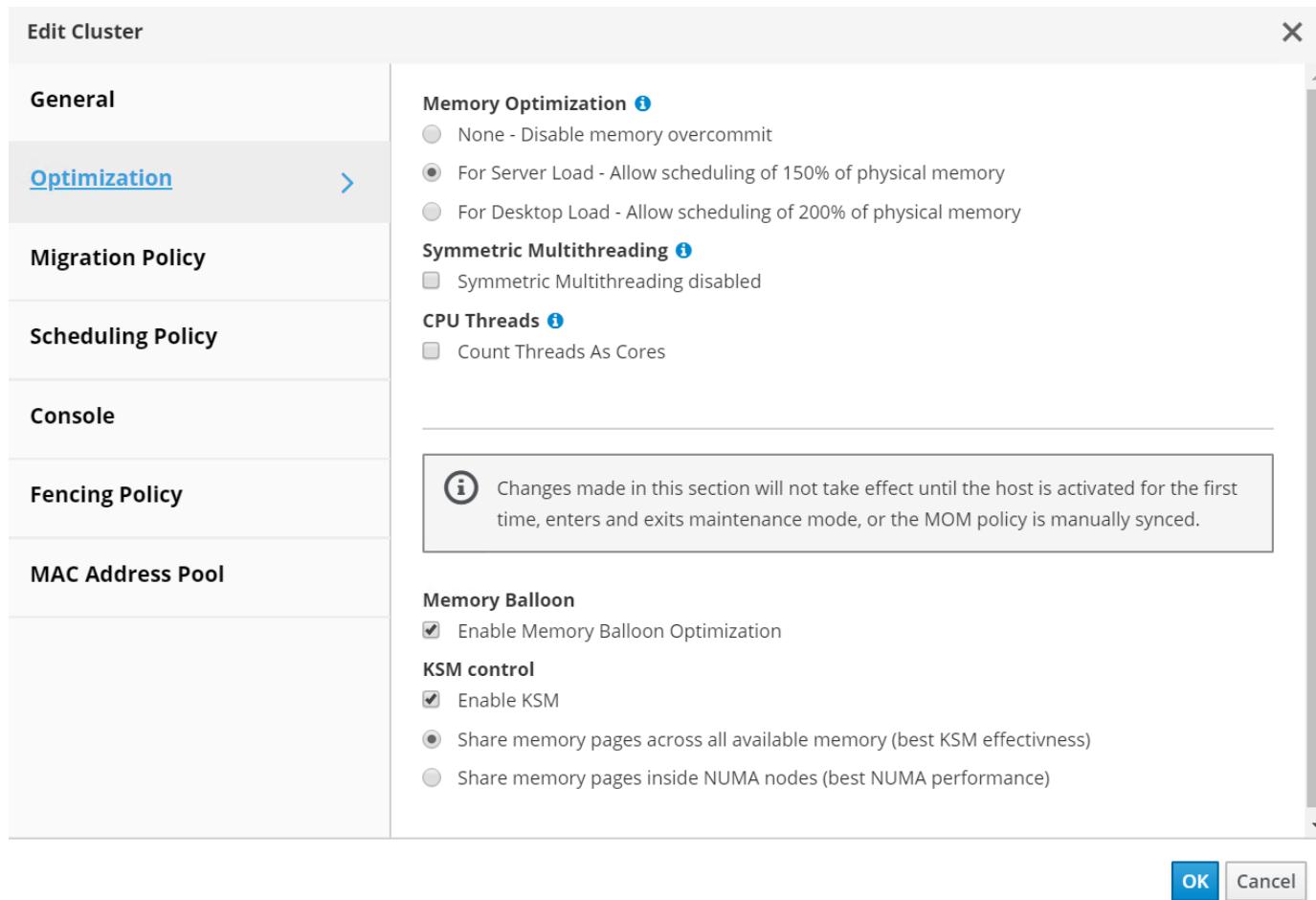
Memory ballooning is a memory management technique which allows a host to artificially expand its memory by reclaiming unused memory that was previously allocated to various VMs, with a limitation of the guaranteed memory size of every VM. For memory ballooning to work, each VM by default has a balloon device with the necessary drivers. Ballooning essentially is a cooperative operation between the VM driver and the host.

Depending on the memory needs of the host, it instructs the guest OS to inflate (provide memory to host) or deflate (regain the memory) the balloon which is controlled by the balloon device.

Kernel Same-page Merging (KSM) allows the host kernel to examine two or more running VMs and compare their image and memory. If any memory regions or pages are identical, KSM reduces multiple identical memory pages to a single page. This page is then marked ‘copy on write’ and a new page is created for that guest VM if the contents of the page are modified by a guest VM.

Both features can be enabled at a cluster level to apply to all hosts in that cluster. To enable these features, navigate to Compute > Clusters, select the desired cluster and click Edit. Then click the Optimization sub-tab and perform the following steps based on your requirements:

1. Depending on the use-case and workload, enable Memory Optimization to allow overcommitment of memory to either 150% or 200% of the available physical memory.
2. To enable memory ballooning, check the Enable Memory Balloon Optimization checkbox.
3. To enable KSM, check the Enable KSM checkbox.
4. Click Ok to confirm the changes.



Be aware that after these changes have been applied, they do not take effect until you manually sync the MoM policy. To sync the MoM policy, navigate to Compute > Clusters and click the cluster for which you made the optimization changes. Navigate to the Hosts sub-tab, select all the hosts, and then click Sync MoM Policy.

Compute » Clusters » Default

**Hosts**

Red Hat Documentation

Sync MoM Policy

Name	Hostname/IP	Status	Load	Display Address Overridden
rhv-h01.cie.netapp.com	rhv-h01.cie.netapp.com	Up	3 VMs	No
rhv-h02.cie.netapp.com	rhv-h02.cie.netapp.com	Up	5 VMs	No

KSM and ballooning can free up some memory on the host and facilitate overcommitment, but, if the amount of shareable memory decreases and the use of physical memory increases, it might cause an out-of-memory condition. Therefore, the administrator should be sure to reserve enough memory to avoid out-of-memory conditions if the shareable memory decreases.

In some scenarios, memory ballooning may collide with KSM. In such situations, MoM tries to adjust the balloon size to minimize collisions. Also, there can be scenarios for which ballooning might cause sub-optimal performance. Therefore, depending on the workload requirements, you can consider enabling either or both the techniques.

[Next: Where to Find Additional Information NetApp HCI with RHV](#)

## Where to Find Additional Information: NetApp HCI with RHV

To learn more about the information described in this document, review the following documents and/or websites:

- NetApp HCI Documentation <https://www.netapp.com/us/documentation/hci.aspx>
- Red Hat Virtualization Documentation [https://access.redhat.com/documentation/en-us/red\\_hat\\_virtualization/4.3/](https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.3/)

## TR-4857: NetApp HCI with Cisco ACI

Abhinav Singh, Nikhil M Kulkarni, NetApp

Cisco Application Centric Infrastructure (Cisco ACI) is an industry-leading, secure, open, and comprehensive Software-Defined Networking (SDN) solution. Cisco ACI radically simplifies, optimizes, and accelerates infrastructure deployment and governance, and it expedites the application deployment lifecycle. Cisco ACI deployed in data centers is proven to work with NetApp HCI with full interoperability. You can manage Ethernet networks for compute, storage, and access with Cisco ACI. You can establish and manage secure network segments for server-to-server and virtual machine (VM)-to-VM communications as well as secure storage-network access through iSCSI from server-to-NetApp HCI storage. This level of endpoint-to-endpoint network security allows customers to architect and operate NetApp HCI in a more secure fashion.

[Next: Use Cases](#)

## Use Cases

The NetApp HCI with Cisco ACI solution delivers exceptional value for customers with the following use cases:

- On-premises software-defined compute, storage, and networking infrastructure
- Large enterprise and service-provider environments
- Private cloud (VMware and Red Hat)
- End User Computing and Virtual Desktop Infrastructure
- Mixed-workload and mixed-storage environments

[Next: Architecture](#)

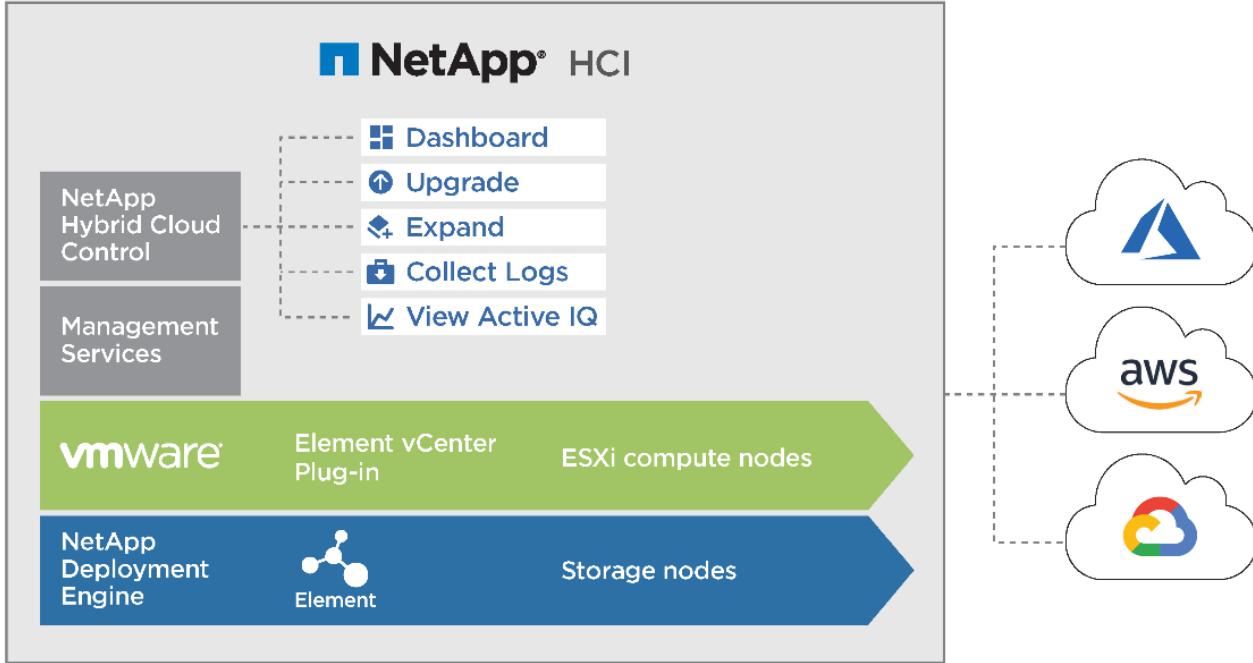
## Architecture

### Solution Technology

This document outlines the best practices to follow for a fully featured on-premises data center or private cloud while interoperating NetApp HCI with Cisco ACI. To demonstrate workload independence, networking best practices are extended to virtualization solutions, including VMware vSphere and Red Hat Virtualization when deployed over NetApp HCI, and to other storage solutions like NetApp ONTAP and StorageGRID. It also emphasizes the interoperability of Cisco ACI switches with different virtual switches, for example, VMware Distributed Switch (VDS), Cisco ACI Virtual Edge (AVE), Linux Bridge, or Open vSwitch.

### NetApp HCI

NetApp HCI is an enterprise-scale, hyper-converged infrastructure solution that delivers compute and storage resources in an agile, scalable, easy-to-manage architecture. Running multiple enterprise-grade workloads can result in resource contention, where one workload interferes with the performance of another. NetApp HCI alleviates this concern with storage quality-of-service (QoS) limits that are available natively within NetApp Element software. Element enables the granular control of every application and volume, helps to eliminate noisy neighbors, and satisfies enterprise performance SLAs. NetApp HCI multitenancy capabilities can help eliminate many traditional performance related problems. See the following graphic for an overview of NetApp HCI.

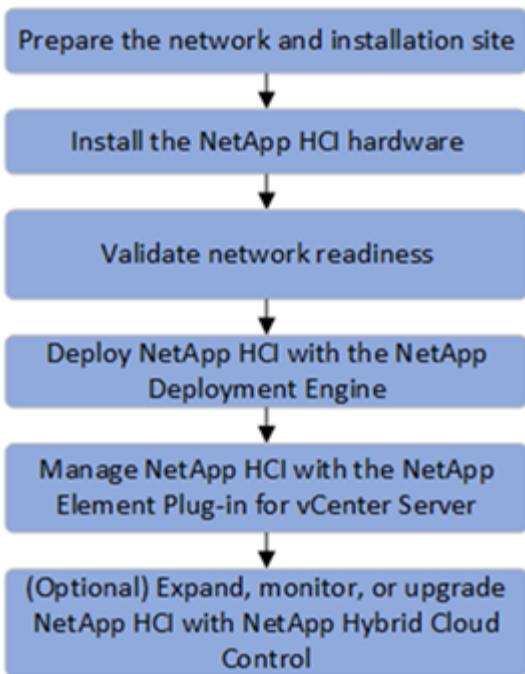


NetApp HCI streamlines installation through the NetApp Deployment Engine (NDE), an intuitive deployment engine that automates more than 400 inputs to fewer than 30 to get your setup running in about 45 minutes. In addition, a robust suite of APIs enables seamless integration into higher-level management, orchestration, backup, and disaster recovery tools. With the NetApp Hybrid Cloud Control management suite, you can manage, monitor, and upgrade your entire infrastructure throughout its lifecycle through a single pane of glass.

#### Software-Defined Architecture

NetApp HCI provides a software-defined approach for deploying and managing data and storage resources. NetApp HCI uses NetApp Element software to provide an easy-to-use GUI-based portal and REST-based API for storage automation, configuration, and management. NetApp Element software provides modular and scalable performance, with each storage node delivering guaranteed capacity and throughput to the environment.

NetApp HCI uses the NetApp Deployment Engine (NDE) to automate the configuration and deployment of physical infrastructure, including the installation and configuration of the VMware vSphere environment and the integration of the NetApp Element Plug-in for vCenter Server. The following figure depicts an overview of the process for deploying NetApp HCI.



### Performance Guarantee

A common challenge is delivering predictable performance when multiple applications are sharing the same infrastructure. An application interfering with other applications creates performance degradation. Mainstream applications have unique I/O patterns that can affect each other's performance when deployed in a shared environment. To address these issues, the NetApp HCI Quality of Service (QoS) feature allows fine-grained control of performance for every application, thereby eliminating noisy neighbors and satisfying performance SLAs. In NetApp HCI, each volume is configured with minimum, maximum, and burst IOPS values. The minimum IOPS setting guarantees performance, independent of what other applications on the system are doing. The maximum and burst values control allocation, enabling the system to deliver consistent performance to all workloads.

NetApp Element software uses the iSCSI storage protocol, a standard way to encapsulate SCSI commands on a traditional TCP/IP network. Element uses a technique called iSCSI login redirection for better performance. iSCSI login redirection is a key part of the NetApp Element software cluster. When a host login request is received, the node decides which member of the cluster should handle the traffic based on IOPS and the capacity requirements for the volume. Volumes are distributed across the NetApp Element software cluster and are redistributed if a single node is handling too much traffic for its volumes or if a new node is added. Multiple copies of a given volume are allocated across the array. In this manner, if a node failure is followed by volume redistribution, there is no effect on host connectivity beyond a logout and login with redirection to the new location. With iSCSI login redirection, a NetApp Element software cluster is a self-healing, scale-out architecture that is capable of nondisruptive upgrades and operations.

### Interoperability

Previous generations of hyperconverged infrastructure typically required fixed resource ratios, limiting deployments to four-node and eight-node configurations. NetApp HCI is a disaggregated hyper-converged infrastructure that can scale compute and storage resources independently. Independent scaling prevents costly and inefficient overprovisioning and simplifies capacity and performance planning.

The architectural design choices offered enables you to confidently scale on your terms, making HCI viable for core Tier-1 data center applications and platforms. It is architected in building blocks at either the chassis or the node level. Each chassis can hold four nodes in a mixed configuration of storage or compute nodes. NetApp HCI is available in mix-and-match, small, medium, and large storage and compute configurations.

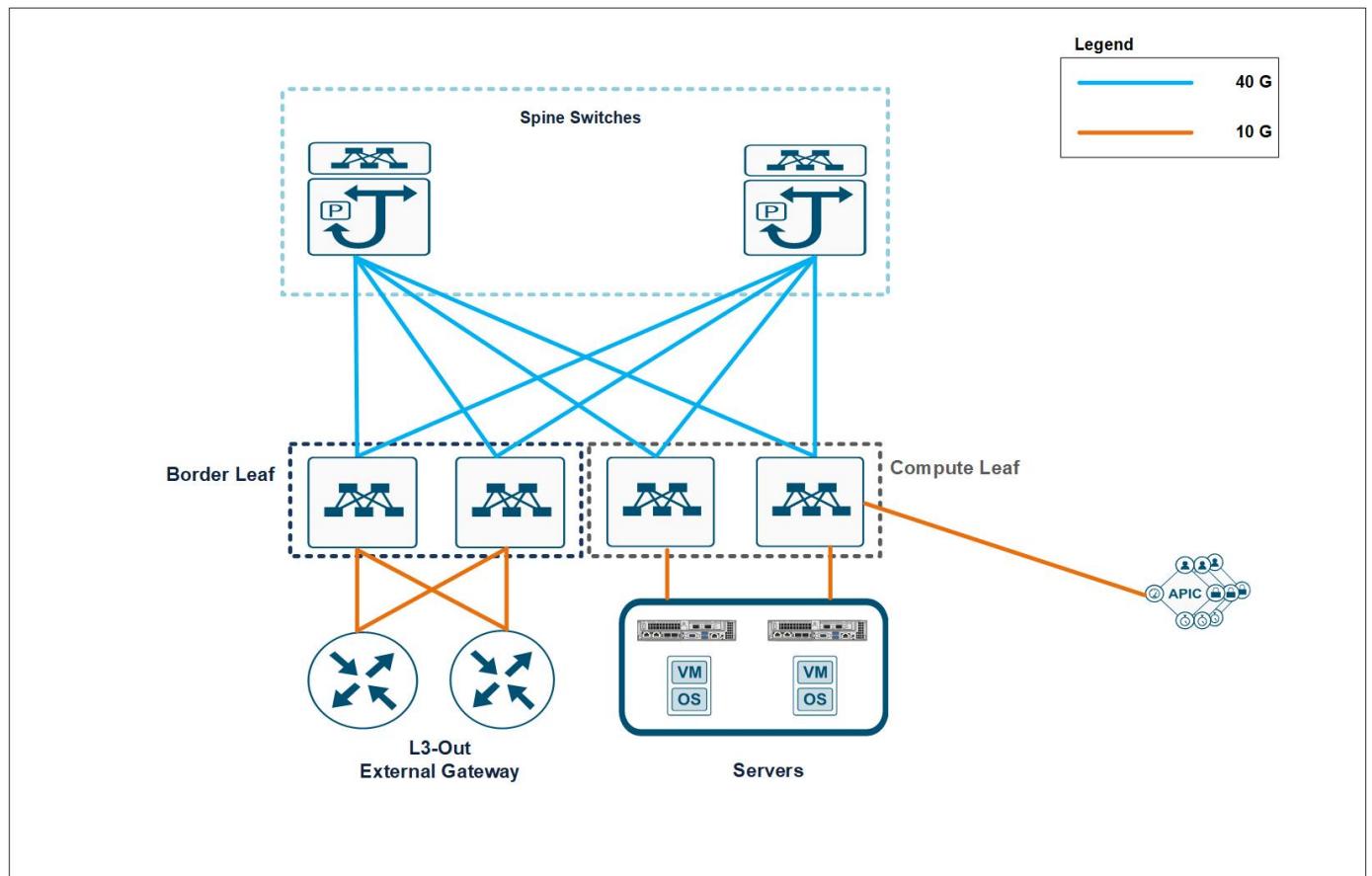
NetApp HCI provides proven multiprotocol and hybrid- cloud support with enterprise grade features. It also offers easy interoperability with multiple different host virtualization technologies and storage solutions. Deploying ONTAP Select and StorageGRID as appliances expands NetApp HCI storage capabilities to include file, block, and object storage services. NetApp HCI provides an agile infrastructure platform for virtual data centers of different flavors. VMware vSphere, Red Hat Virtualization, KVM, Citrix Hypervisor, and so on are supported platforms that can use the NetApp HCI infrastructure to provide a scalable, enterprise-grade on-premises virtual environment.

For more details, see the [NetApp HCI documentation](#).

## Cisco ACI

Cisco ACI is an industry leading software-defined networking solution that facilitates application agility and data center automation. Cisco ACI has a holistic architecture with a centralized policy-driven management. It implements a programmable data center Virtual Extensible LAN (VXLAN) fabric that delivers distributed networking and security for any workload, regardless of its nature (virtual, physical, container, and so on).

Cisco pioneered the introduction of intent-based networking with Cisco ACI in the data center. It combines the high- performance hardware and robust software integrated with two important SDN features—overlays and centralized control. The ACI fabric consists of Cisco Nexus 9000 series switches running in ACI mode and a cluster of at least three centrally managed Application Policy Infrastructure Controllers (APIC) servers. The following figure provides an overview of Cisco ACI.



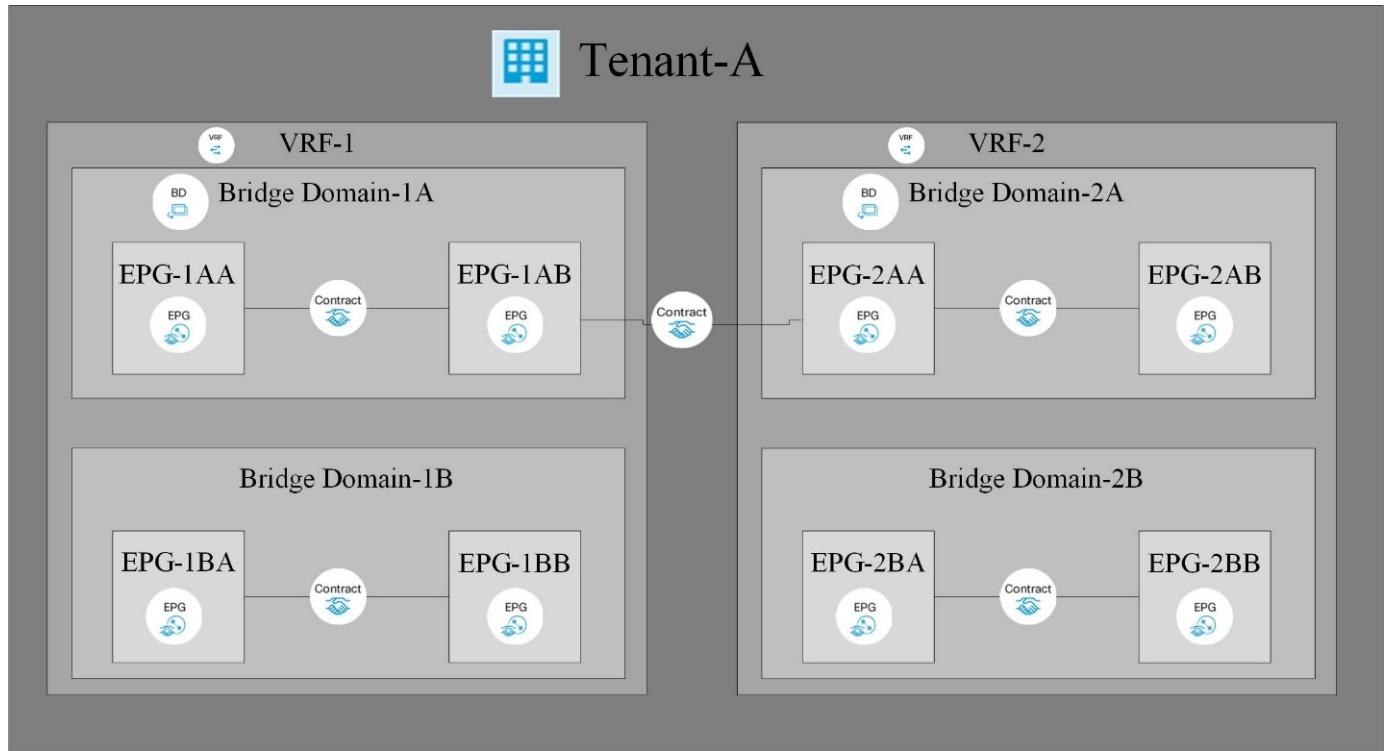
## Policy-Driven Networking

Cisco ACI, with its policy driven model, makes network hardware stateless. The Application Policy Infrastructure Controller (APIC) acts as the central controller managing and configuring all the switches in the ACI fabric. The Cisco ACI fabric consists of Cisco Nexus 9000 series switches which are centrally configured

and managed by the cluster of APICs using the declarative policy model.

Cisco ACI uses logical constructs to form a layered policy architecture to define and manage the different functions of the entire fabric, including infrastructure, authentication, security, services, applications, and diagnostics.

The following figure depicts the categorization and relation between different logical constructs in Cisco ACI.



Tenants are logical containers with administrative boundaries that exercise domain-based access control. It is a logical policy isolation and does not equate to a real network construct.

Within the tenant, a context is a unique layer-3 forwarding policy domain. A context can be directly mapped to the Virtual Routing and Forwarding (VRF) concept of traditional networks. In fact, a context is also called VRF. Because each context is a separate layer- 3 domain, two different contexts can have overlapping IP spaces.

Within a context, a bridge domain (BD) represents a unique layer-2 forwarding construct. The bridge domain defines the unique layer-2 MAC address space and can be equated to a layer-2 flood domain or to a layer-3 gateway. A bridge domain can have zero subnets, but it must have at least one subnet if it is to perform routing for the hosts residing in the BD.

In ACI, an endpoint is anything that communicates on the network, be it a compute host, a storage device, a network entity that is not part of the ACI fabric, a VM, and so on. A group of endpoints that have the same policy requirements are categorized into an Endpoint Group (EPG). An EPG is used to configure and manage multiple endpoints together. An EPG is a member of a bridge domain. One EPG cannot be a member of multiple bridge domains, but multiple EPGs can be members of a single bridge domain.

All the endpoints that belong to the same EPG can communicate with each other. However, endpoints in different EPGs cannot communicate by default, but they can communicate if a contract exists between the two EPGs allowing that communication. Contracts can be equated to ACLs in traditional networking. However, it differs from an ACL in the way that it doesn't involve specifying specific IP addresses as source and destination and that contracts are applied to an EPG as a whole.

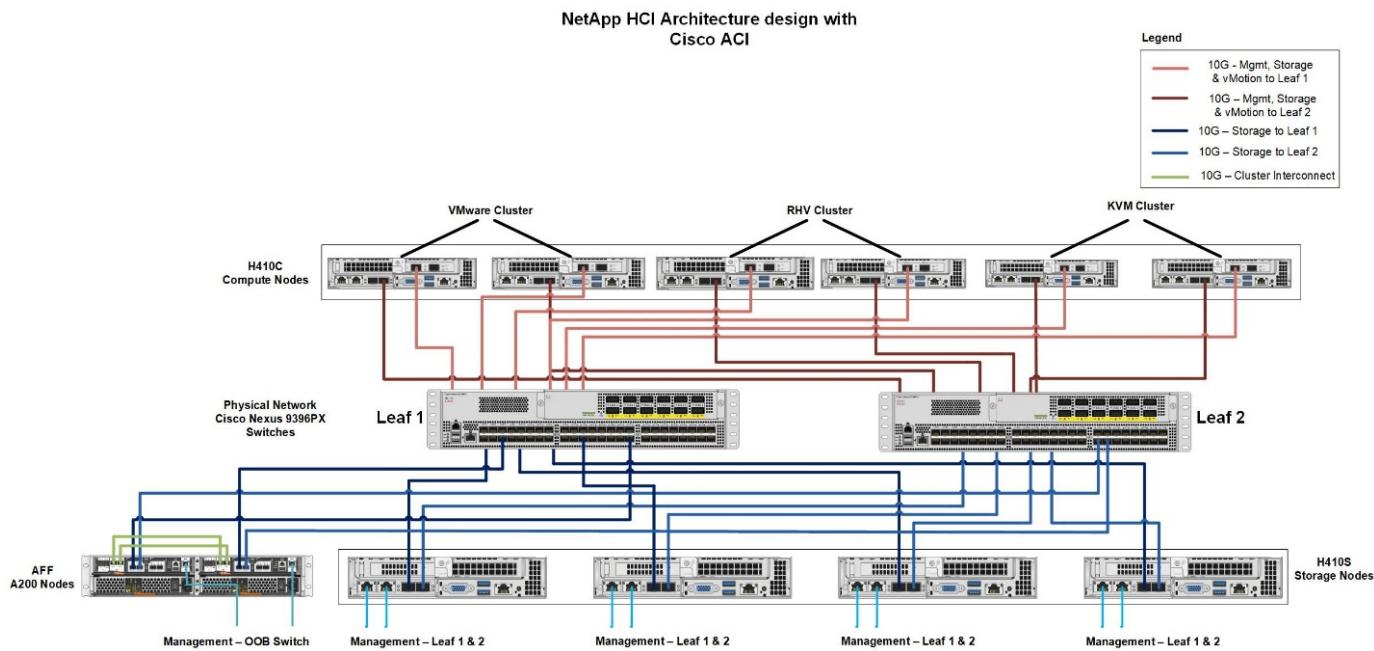
See the [Cisco ACI documentation](#) for more information.

## Networking Advantages

Cisco ACI provides many advantages over traditional networking. Programmability and automation are critical features of a scalable data center virtualization infrastructure and the policy driven mechanism of Cisco ACI opens a lot of opportunities for providing optimal physical and virtual networking.

- **Virtual Machine Manager (VMM) Integration.** With the Cisco ACI open REST API features, integration with virtualized environments is easy. Cisco ACI supports VMM integration with multiple hypervisors and provides automated access and control over the hypervisor virtual switches to the networking constructs in ACI. VMM integration in ACI seamlessly extends the ACI policy framework to virtual workloads. In other words, VMM integration allows Cisco ACI to control the virtual switches running on virtualization hosts and to extend the ACI fabric access policies to virtual workloads. The integration also automates the hypervisor's virtual switch deployment and configuration tasks. Cisco ACI VMM integration provides the following benefits:
  - Single point of policy management for physical and virtual environments through APIC
  - Faster application deployment, with transparent instantiation of applications in virtual environments
  - Full integrated visibility into the health of the application through holistic aggregation of information across physical and virtual environments
  - Simplified networking configuration for virtual workloads because the port-group or VM NIC profiles required to attach to the VMs are created automatically. For more information on Cisco ACI VMM integration, see the [Cisco documentation](#). In addition, see the Cisco ACI [virtualization compatibility matrix](#) for version compatibility details.
- **Micro-segmentation.** Micro-segmentation in Cisco ACI allows you to classify the endpoints in existing application EPGs into microsegment (uSeg) EPGs using network-based or VM-based attributes. This helps for filtering the endpoints more granularly and apply specific dynamic policies on those endpoints. Micro-segmentation can be applied to any endpoints within the tenant. Cisco supports micro-segmentation on a variety of virtual switches - Cisco ACI Virtual Edge, VMware VDS and Microsoft vSwitch. uSeg EPGs can be configured with multiple attributes but an endpoint can be assigned to only one EPG. For more details, see the [Cisco ACI Virtualization guide](#) for the specific version.
- **Intra-EPG Isolation.** By default, all endpoints belonging to the same EPG can communicate with each other. Intra-EPG Isolation in Cisco ACI is a feature to prevent endpoints in the same EPG communicate with each other. It achieves isolation by using different VLANs for traffic from ACI leaf to hypervisor hosts and from hypervisor hosts to ACI leaf. Intra-EPG isolation can be enforced on both application EPGs and microsegment EPGs. See the specific version of the [Cisco ACI virtualization guide](#) for more information.

## Architectural Diagram



This diagram represents the physical architecture of NetApp HCI with Cisco ACI that was designed for this solution. Two leaf switches connected via spines and managed by a cluster of three APICs forms the ACI fabric. The leaf switches are connected to upstream routers for external connectivity. Three pairs of NetApp HCI compute nodes (each pair dedicated for a hypervisor) are configured with a two-cable option. Four storage nodes were configured with four-cable option to form the Element cluster. A pair of AFF A200 nodes are used to provide the ONTAP capabilities to the system.

## Hardware and Software Requirements

### Compute

The following tables list the hardware and software compute resources utilized in the solution. The components that are used in any implementation of the solution might vary based on customer requirements.

Hardware	Model	Quantity
NetApp HCI compute nodes	NetApp H410C	6

Software	Purpose	Version
VMware ESXi	Virtualization	6.7
VMware vCenter Server Appliance	Virtualization management	6.7
Red Hat Enterprise Linux	Operating system	7.7
KVM	Virtualization	1.5.3-167
Red Hat Virtualization	Virtualization	4.3.9

### Storage

The following tables list the hardware and software storage resources used in this solution. The components that are used in any particular implementation of the solution might vary based on customer requirements.

<b>Hardware</b>	<b>Model</b>	<b>Quantity</b>
NetApp HCI storage nodes	NetApp H410S	4
AFF	A200	2

<b>Software</b>	<b>Purpose</b>	<b>Version</b>
NetApp HCI	Infrastructure	1.8
NetApp Element	Storage	12.0
ONTAP	Storage	9.7P6
ONTAP Select	Storage	9.7
Storage Grid	Storage	11.3

## Networking

The following tables list the hardware and software network resources used in this solution. The components that are used in any particular implementation of the solution might vary based on customer requirements.

<b>Hardware</b>	<b>Model</b>	<b>Quantity</b>
Cisco UCS server	UCS C-220 M3	3
Cisco Nexus	N9K-C9336-PQ	2
Cisco Nexus	N9K-C9396-PX	2

<b>Software</b>	<b>Purpose</b>	<b>Version</b>
Cisco APIC	Network Management	3.2(9h)
Cisco Nexus ACI-mode Switch	Network	13.2(9h)
Cisco AVE	Network	1.2.9
Open vSwitch (OVS)	Network	2.9.2
VMware Virtual Distributed Switch	Network	6.6

[Next: Design Considerations](#)

## Design Considerations

### Network Design

The minimum configuration of a Cisco ACI fabric consists of two leaf switches and two spine switches with a cluster at least three APICs managing and controlling the whole fabric. All the workloads connect to leaf switches. Spine switches are the backbone of the network and are responsible for interconnecting all leaf switches. No two leaf switches can be interconnected. Each leaf switch is connected to each of the spine switches in a full-mesh topology.

With this two-tier spine-and-leaf architecture, no matter which leaf switch the server is connected to, its traffic always crosses the same number of devices to get to another server attached to the fabric (unless the other server is located on the same leaf). This approach keeps latency at a predictable level.

## Compute Design

The minimum number of compute nodes required for a highly available infrastructure using NetApp HCI is two. NetApp HCI provides two options for cabling: two-cable and six-cable. NetApp HCI H410C compute nodes are available with two 1GbE ports (ports A and B) and four 10/25GbE ports (ports C, D, E, and F) on board. For a two-cable option, ports D and E are used for connectivity to uplink switches, and, for a six-cable option, all ports from A to F are used. Each node also has an additional out-of-band management port that supports Intelligent Platform Management Interface (IPMI) functionality. This solution utilizes the two-cable option for compute nodes.

For VMware deployments, NetApp HCI comes with an automated deployment tool called the NetApp Deployment Engine (NDE). For non-VMware deployments, manual installation of hypervisors or operating systems is required on the compute nodes.

## Storage Design

NetApp HCI uses four-cable option for storage nodes. NetApp HCI H410S storage nodes are available with two 1GbE ports (ports A and B) and two 10/25GbE ports (ports C and D) on board. The two 1GbE ports are bundled as Bond1G (active/passive mode) used for management traffic and the two 10/25GbE ports are bundled as Bond10G (LACP active mode) used for storage data traffic.

For non-VMware deployments, the minimum configuration of NetApp HCI storage cluster is four nodes. For NetApp HCI versions earlier than 1.8 with VMware deployments, the minimum configuration is four storage nodes. However, for HCI version 1.8 with VMware deployments, the minimum configuration for NetApp HCI storage cluster is two nodes. For more information on NetApp HCI two-node storage cluster, see the documentation [here](#).

Next: [VMware vSphere: NetApp HCI with Cisco ACI](#)

## Deploying NetApp HCI with Cisco ACI

### VMware vSphere: NetApp HCI with Cisco ACI

VMware vSphere is an industry-leading virtualization platform that provides a way to build a resilient and reliable virtual infrastructure. vSphere contains virtualization, management, and interface layers. The two core components of VMware vSphere are ESXi server and the vCenter Server. VMware ESXi is hypervisor software installed on a physical machine that facilitates hosting of VMs and virtual appliances. vCenter Server is the service through which you manage multiple ESXi hosts connected in a network and pool host resources. For more information on VMware vSphere, see the documentation [here](#).

### Workflow

The following workflow was used to up the virtual environment. Each of these steps might involve several individual tasks.

1. Install and configure Nexus 9000 switches in ACI mode and APIC software on the UCS C-series server. See the Install and Upgrade [documentation](#) for detailed steps.
2. Configure and setup ACI fabric by referring to the [documentation](#).
3. Configure the tenants, application profiles, bridge domains, and EPGs required for NetApp HCI nodes. NetApp recommends using one BD to one EPG framework, except for iSCSI. See the documentation [here](#) for more details. The minimum set of EPGs required are in-band management, iSCSI, iSCSI-A, iSCSI-B,

VM motion, VM-data network, and native.



iSCSI multipathing requires two iSCSI EPGs: iSCSI-A and iSCSI-B, each with one active uplink.



NetApp mNode requires an iSCSI EPG with both uplinks active.

4. Create the VLAN pool, physical domain, and AEP based on the requirements. Create the switch and interface profiles for individual ports. Then attach the physical domain and configure the static paths to the EPGs. See the [configuration guide](#) for more details.

#### VLAN Pool - HCI-Internal-Phys-Dom-VLAN (Static Allocation)



Policy   Operational   Faults   History

Properties

Name: HCI-Internal-Phys-Dom-VLAN

Description: optional

Alias:

Allocation Mode: Static Allocation

Encap Blocks:

VLAN Range	Allocation Mode	Role
[2]	Inherit allocMode from parent	External or On the wire encapsulations
[3201-3250]	Inherit allocMode from parent	External or On the wire encapsulations

Domains: ▲ Name      Type

Name	Type
HCI-Internal-Phys-Dom	Physical Domain

Show Usage   Close   Submit

# Leaf Access Port Policy Group - HCI-Compute-ESX

The screenshot shows a software interface for managing network policy groups. At the top, there are four small circular icons: a red one with an 'X', an orange one with a downward arrow, a yellow one with an upward arrow, and a green one with a right-pointing arrow. Below these is a section titled 'Properties'.

**Name:** HCI-Compute-ESX

**Description:** optional

**Alias:** (empty field)

**Link Level Policy:** 10G-Auto (dropdown menu with a blue refresh icon)

**CDP Policy:** CDP-Disabled (dropdown menu with a blue refresh icon)

**MCP Policy:** select a value (dropdown menu)

**CoPP Policy:** select a value (dropdown menu)

**LLDP Policy:** LLDP-Enabled (dropdown menu with a blue refresh icon)



Use an access port policy group for interfaces connecting to NetApp HCI compute nodes, and use vPC policy group for interfaces to NetApp HCI storage nodes.

5. Create and assign contracts for tightly-controlled access between workloads. For more information on configuring the contracts, see the guide [here](#).
6. Install and configure NetApp HCI using NDE. NDE configures all the required parameters, including VDS port groups for networking, and also installs the mNode VM. See the [deployment guide](#) for more information.
7. Though VMM integration of Cisco ACI with VMware VDS is optional, using the VMM integration feature is a best practice. When not using VMM integration, an NDE-installed VDS can be used for networking with physical domain attachment on Cisco ACI.
8. If you are using VMM integration, NDE-installed VDS cannot be fully managed by ACI and can be added as read-only VMM domain. To avoid that scenario and make efficient use of Cisco ACI's VMM networking feature, create a new VMware VMM domain in ACI with a new VMware vSphere Distributed Switch (vDS) and an explicit dynamic VLAN pool. The VMM domain created can integrate with any supported virtual switch.
  - a. **Integrate with VDS.** If you wish to integrate ACI with VDS, select the virtual switch type to be VMware Distributed Switch. Consider the configuration best practices noted in the following table. See the [configuration guide](#) for more details.

## Properties

Name: hci-aci-vds-02

Virtual Switch: Distributed Switch

Associated Attachable Entity ▲ Name

Profiles:

HCI-Internal

---

Encapsulation: vlan

Delimiter:

Enable Tag Collection:

Enable VM Folder Data Retrieval:

Access Mode:

Endpoint Retention Time (seconds):

VLAN Pool: hci-aci-vmware(dynamic)  

- b. **Integrate with Cisco AVE.** If you are integrating Cisco AVE with Cisco ACI, select the virtual switch type to be Cisco AVE. Cisco AVE requires a unique VLAN pool of type Internal for communicating between internal and external port groups. Follow the configuration best practices noted in this table. See the [installation guide](#) to install and configure Cisco AVE.

## Properties

Name: hci-vmware-ave

Virtual Switch: Cisco AVE

AVE Time-out Time (seconds):  ^ ▼

Host Availability Assurance:

Associated Attachable Entity ▲ Name  
Profiles:

---

Switching Preference:  No Local Switching  Local Switching

Enhanced Lag Policy:  ▼

Encapsulation: vxlan

Default Encap Mode:  Unspecified  VLAN  VXLAN

---

Enable Tag Collection:

Enable VM Folder Data Retrieval:

Endpoint Retention Time (seconds):  ^ ▼

VLAN Pool:  ▼ ✚

AVE Fabric-Wide Multicast   
Address: Must Use a Multicast Address different  
from the Multicast Address Ranges.

Pool of Multicast Addresses (one  
per-EPG):  ▼ ✚

9. Attach the VMM domain to the EPGs using Pre-Provision Resolution Immediacy. Then migrate all the VMNICs, VMkernel ports, and VNICs from the NDE-created VDS to ACI-created VDS or AVE and so on. Configure the uplink failover and teaming policy for iSCSI-A and iSCSI-B to have one active uplink each. VMs can now attach their VMNICs to ACI-created port groups to access network resources. The port groups on VDS that are managed by Cisco ACI are in the format of <tenant-name> | <application-profile-name> | <epg-name>.



Pre-Provision Resolution Immediacy is required to ensure the port policies are downloaded to the leaf switch even before the VMM controller is attached to the virtual switch.

Summary    Monitor    Configure    Permissions    Ports    Hosts    VMs    Networks

Manufacturer: VMware, Inc.  
Version: 6.6.0  
[Upgrades available](#)

**Switch Details**

**Notes**  
APIC Virtual Switch  
[Edit Notes...](#)

## VMkernel adapters

Device	Network Label	Switch	IP Address	TCP/IP Stack	vMotion	Provisioning
vmk0	HCI-InfraHClH...	hcl-vmware-ave	172.22.9.60	Default	Disabled	Disabled
vmk1	HCI-InfraHClIS...	hcl-vmware-ave	172.22.10.60	Default	Disabled	Disabled
vmk2	HCI-InfraHClIS...	hcl-vmware-ave	172.22.10.58	Default	Disabled	Disabled
vmk3	HCI-InfraHClH...	hcl-vmware-ave	172.22.13.60	Default	Enabled	Disabled
vmk4	HCI-InfraAFF-A...	hcl-vmware-ave	172.22.15.60	Default	Disabled	Disabled

- If you intend to use micro-segmentation, then create micro-segment (uSeg) EPGs attaching to the right BD. Create attributes in VMware vSphere and attach them to the required VMs. Ensure the VMM domain has Enable Tag Collection enabled. Configure the uSeg EPGs with the corresponding attribute and attach the VMM domain to it. This provides more granular control of communication on the endpoint VMs.

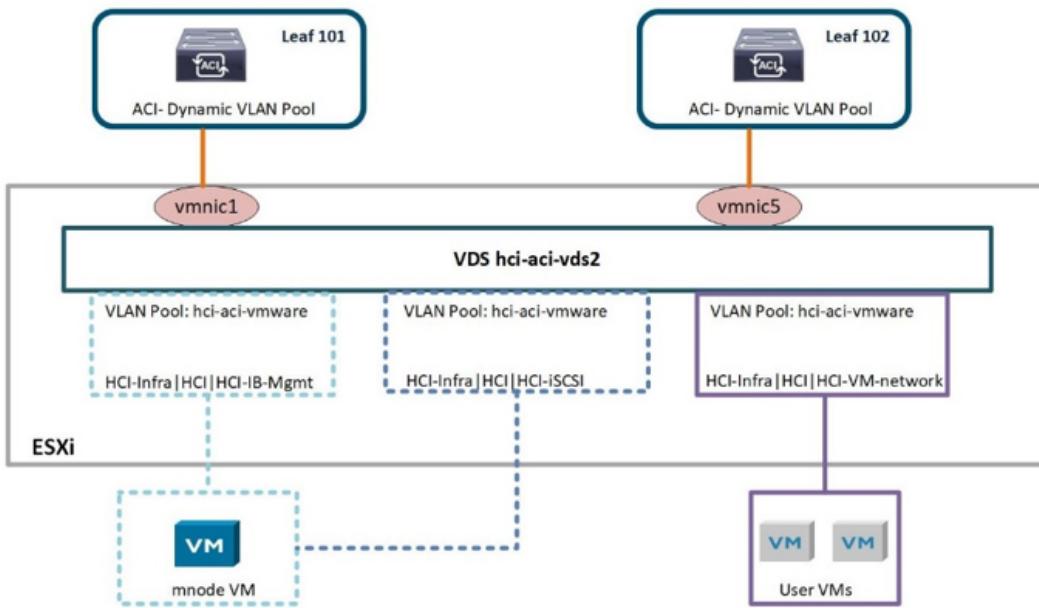
The networking functionality for VMware vSphere on NetApp HCI in this solution is provided either using VMware VDS or Cisco AVE.

## VMware VDS

VMware vSphere Distributed Switch (VDS) is a virtual switch that connects to multiple ESXi hosts in the cluster or set of clusters allowing virtual machines to maintain consistent network configuration as they migrate across multiple hosts. VDS also provides for centralized management of network configurations in a vSphere environment. For more details, see the [VDS documentation](#).

### Legends

	EPG:HCI-IB-Mgmt (VMKernel)
	EPG:HCI-iSCSI (VMKernel)
	EPG:HCI-VM-network



The following table outlines the necessary parameters and best practices for configuring and integrating Cisco ACI with VMware VDS.

Resource	Configuration Considerations	Best Practices
Endpoint groups	<ul style="list-style-type: none"> <li>Separate EPG for native VLANs</li> <li>Static binding of interfaces to HCI storage and compute nodes in native VLAN EPG uses 802.1P mode. This is required for node discovery to run NDE.</li> <li>Separate EPGs for iSCSI, iSCSI-A, and iSCSI-B with a common BD</li> <li>iSCSI-A and iSCSI-B are for iSCSI multipathing and are used for VMkernel ports on ESXi hosts</li> <li>Physical domain to be attached to iSCSI EPG before running NDE</li> <li>VMM domain to be attached to iSCSI, iSCSI-A, and iSCSI-B EPGs</li> </ul>	<ul style="list-style-type: none"> <li>Contracts between EPGs to be well defined. Allow only required ports for communication.</li> <li>Use unique native VLAN for NDE node discovery</li> <li>For EPGs corresponding to port-groups being attached to VMkernel ports, VMM domain to be attached with Pre-Provision for Resolution Immediacy</li> </ul>
Interface policy	<ul style="list-style-type: none"> <li>A common leaf access port policy group for all ESXi hosts</li> <li>One vPC policy group per NetApp HCI storage node</li> <li>LLDP enabled, CDP disabled</li> </ul>	<ul style="list-style-type: none"> <li>Separate VLAN pool for VMM domain with dynamic allocation turned on</li> <li>Recommended to use vPC with LACP Active port-channel policy for interfaces towards NetApp HCI storage nodes</li> <li>Recommended to use individual interfaces for compute nodes, no LACP.</li> </ul>
VMM Integration	<ul style="list-style-type: none"> <li>Local switching preference</li> <li>Access mode is Read Write.</li> </ul>	<ul style="list-style-type: none"> <li>MAC-Pinning-Physical-NIC-Load for vSwitch policy</li> <li>LLDP for discovery policy</li> <li>Enable Tag collection if micro-segmentation is used</li> </ul>
VDS	<ul style="list-style-type: none"> <li>Both uplinks active for iSCSI port-group</li> <li>One uplink each for iSCSI-A and iSCSI-B</li> </ul>	<ul style="list-style-type: none"> <li>Load balancing method for all port-groups to be 'Route based on physical NIC load'</li> <li>iSCSI VMkernel port migration to be done one at a time from NDE deployed VDS to ACI integrated VDS</li> </ul>

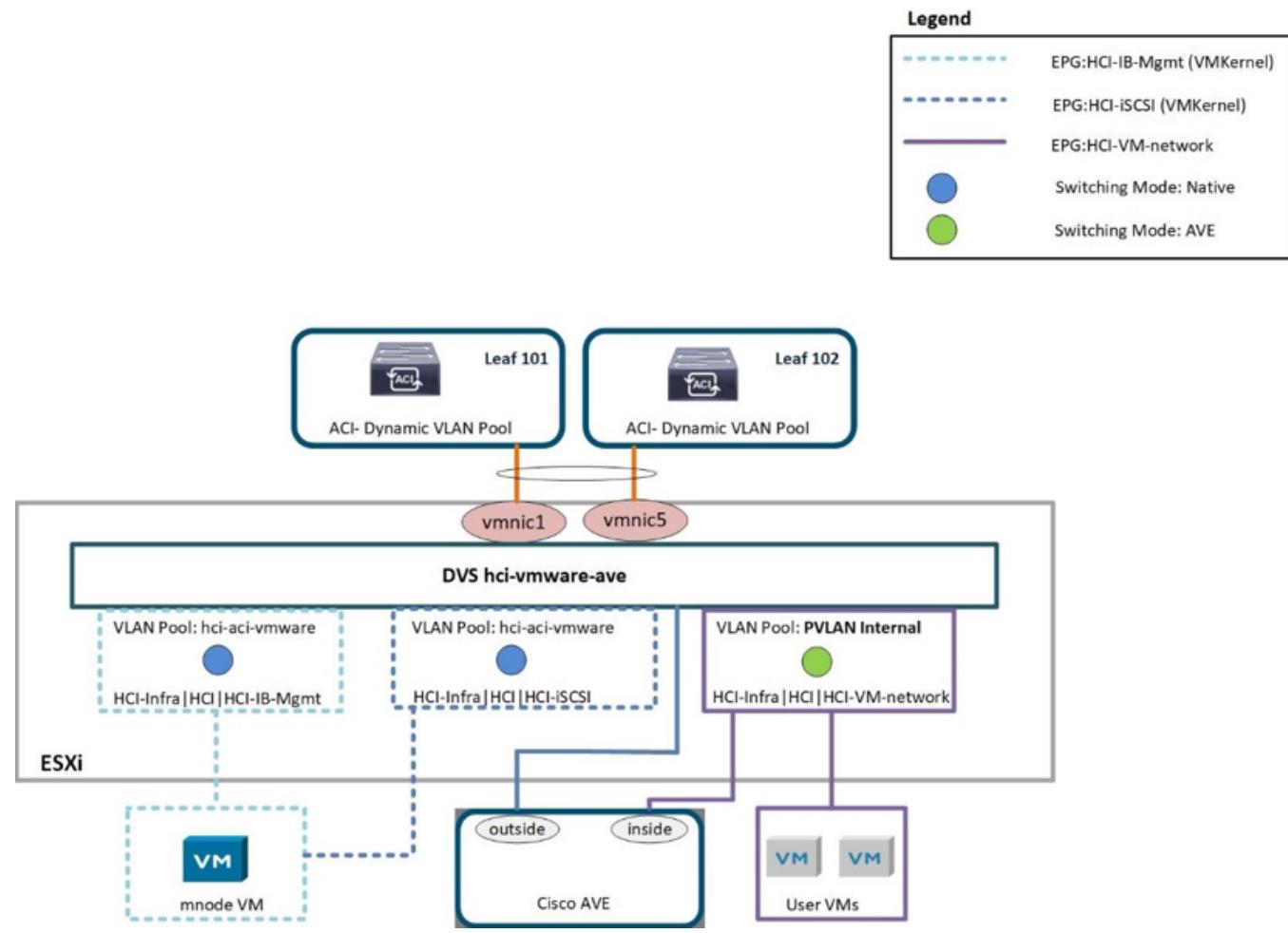
Resource	Configuration Considerations	Best Practices
Easy Scale	<ul style="list-style-type: none"> <li>• Run NDE scale by attaching the same leaf access port policy group for ESXi hosts to be added</li> <li>• One vPC policy group per NetApp HCI storage node</li> <li>• Individual interfaces (for ESXi hosts) and vPCs (for storage nodes) should be attached to native, in-band management, iSCSI, VM motion EPGs for successful NDE scale</li> <li>• LLDP enabled, CDP disabled</li> </ul>	<ul style="list-style-type: none"> <li>• Recommended to use vPC with LACP Active port-channel policy for interfaces towards NetApp HCI storage nodes</li> <li>• Recommended to use individual interfaces for compute nodes, no LACP.</li> </ul>



For traffic load-balancing, port channels with vPCs can be used on Cisco ACI along with LAGs on VDS with LACP in active mode. However, using LACP can affect storage performance when compared to iSCSI multipathing.

## Cisco AVE

Cisco ACI Virtual Edge (AVE) is a virtual switch offering by Cisco that extends the Cisco ACI policy model to virtual infrastructure. It is a hypervisor-independent distributed network service that sits on top of the native virtual switch of the hypervisor. It leverages the underlying virtual switch using a VM-based solution to provide network visibility into the virtual environments. For more details on Cisco AVE, see the [documentation](#). The following figure depicts the internal networking of Cisco AVE on an ESXi host (as tested).



The following table lists the necessary parameters and best practices for configuring and integrating Cisco ACI with Cisco AVE on VMware ESXi. Cisco AVE is currently only supported with VMware vSphere.

Resource	Configuration Considerations	Best Practices
Endpoint Groups	<ul style="list-style-type: none"> <li>Separate EPG for native VLANs</li> <li>Static binding of interfaces towards HCI storage and compute nodes in native VLAN EPG uses 802.1P mode. This is required for node discovery to run NDE.</li> <li>Separate EPGs for iSCSI, iSCSI-A and iSCSI-B with a common BD</li> <li>iSCSI-A and iSCSI-B are for iSCSI multipathing and are used for VMkernel ports on ESXi hosts</li> <li>Physical domain to be attached to iSCSI EPG before running NDE</li> <li>VMM domain is attached to iSCSI, iSCSI-A, and iSCSI-B EPGs</li> </ul>	<ul style="list-style-type: none"> <li>Separate VLAN pool for VMM domain with dynamic allocation turned on</li> <li>Contracts between EPGs to be well defined. Allow only required ports for communication.</li> <li>Use unique native VLAN for NDE node discovery</li> <li>Use native switching mode in VMM domain for EPGs that correspond to port groups being attached to host's VMkernel adapters</li> <li>Use AVE switching mode in VMM domain for EPGs corresponding to port groups carrying user VM traffic</li> <li>For EPGs corresponding to port-groups being attached to VMkernel ports, VMM domain is attached with Pre-Provision for Resolution Immediacy</li> </ul>
Interface Policy	<ul style="list-style-type: none"> <li>One vPC policy group per NetApp HCI storage node</li> <li>LLDP enabled, CDP disabled</li> <li>Before running NDE, for NDE discovery: <ul style="list-style-type: none"> <li>Leaf Access port policy group for all ESXi hosts</li> </ul> </li> <li>After running NDE, for Cisco AVE: <ul style="list-style-type: none"> <li>One vPC policy group per ESXi host</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>NetApp recommends using vPCs to ESXi hosts for Cisco AVE</li> <li>Use static mode on port-channel policy for vPCs to ESXi</li> <li>Use Layer-4 SRC port load balancing hashing method for port-channel policy</li> <li>NetApp recommends using vPC with LACP active port-channel policy for interfaces to NetApp HCI storage nodes</li> </ul>

Resource	Configuration Considerations	Best Practices
VMM Integration	<ul style="list-style-type: none"> <li>Create a new VLAN range [or Encap Block] with role Internal and Dynamic allocation' attached to the VLAN pool intended for VMM domain</li> <li>Create a pool of multicast addresses (one address per EPG)</li> <li>Reserve another multicast address different from the pool of multicast addresses intended for AVE fabric-wide multicast address</li> <li>Local switching preference</li> <li>Access mode to be Read Write mode</li> </ul>	<ul style="list-style-type: none"> <li>Static mode on for vSwitch policy</li> <li>Ensure that vSwitch port-channel policy and interface policy group's port-channel policy are using the same mode</li> <li>LLDP for discovery policy</li> <li>Enable Tag collection if using micro-segmentation</li> <li>Recommended option for Default Encap mode is VXLAN</li> </ul>
VDS	<ul style="list-style-type: none"> <li>Both uplinks active for iSCSI port-group</li> <li>One uplink each for iSCSI-A and iSCSI-B</li> </ul>	<ul style="list-style-type: none"> <li>iSCSI VMkernel port migration is done one at a time from NDE deployed VDS to ACI integrated VDS</li> <li>Load balancing method for all port-groups to be Route based on IP hash</li> </ul>
Cisco AVE	<ul style="list-style-type: none"> <li>Run NDE with access port interface policy groups towards ESXi hosts. Individual interfaces towards ESXi hosts should be attached to native, in-band management, iSCSI, VM motion EPGs for successful NDE run.</li> <li>Once the environment is up, place the host in maintenance mode, migrate interface policy group to vPC with static mode on, assign vPC to all required EPGs and remove the host from maintenance mode. Repeat the same process for all hosts.</li> <li>Run the AVE installation process to install AVE control VM on all hosts</li> </ul>	<ul style="list-style-type: none"> <li>Use local datastore on the hosts for installing AVE control VM. Each host should have one AVE control VM installed on it</li> <li>Use network protocol profile on the in-band management VLAN if DHCP is not available on that network</li> </ul>

Resource	Configuration Considerations	Best Practices
Easy Scale	<ul style="list-style-type: none"> <li>• Run NDE scale with access port interface policy group for ESXi hosts to be added. Individual interfaces should be attached to native, in-band management, iSCSI, VM motion EPGs for successful NDE run.</li> <li>• Once the ESXi host is added to the vSphere cluster, place the host in maintenance mode and migrate the interface policy group to vPC with static mode on. Then attach the vPC to required EPGs.</li> <li>• Run AVE installation process on the new host for installing AVE control VM on that host</li> <li>• One vPC policy group per NetApp HCI storage node to be added to the cluster</li> <li>• LLDP enabled, CDP disabled</li> </ul>	<ul style="list-style-type: none"> <li>• Use local datastore on the host for installing AVE control VM</li> <li>• Use network protocol profile on the in-band management VLAN if DHCP is not available on that network</li> <li>• Recommended to use vPC with LACP Active port-channel policy for interfaces towards NetApp HCI storage nodes</li> </ul>

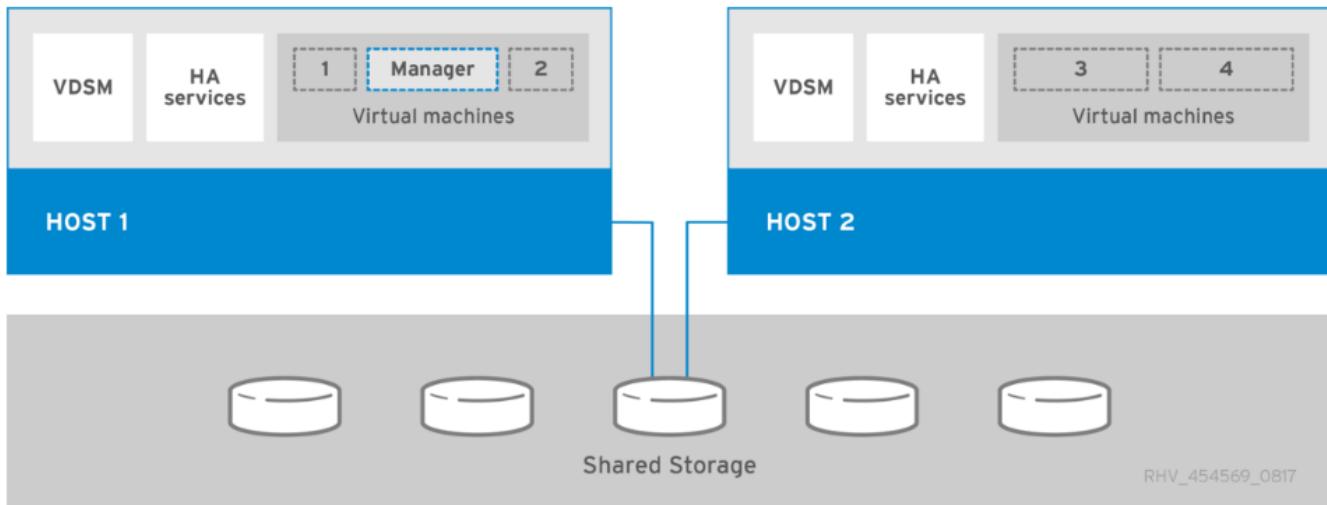


For traffic load balancing, port channel with vPCs can be used on Cisco ACI along with LAGs on ESXi hosts with LACP in active mode. However, using LACP can affect storage performance when compared to iSCSI multipathing.

Next: Red Hat Virtualization: NetApp HCI with Cisco ACI

## Red Hat Virtualization: NetApp HCI with Cisco ACI

Red Hat Virtualization (RHV) is an enterprise virtual data center platform that runs on Red Hat Enterprise Linux using the KVM hypervisor. The key components of RHV include Red Hat Virtualization Hosts (RHV-H) and the Red Hat Virtualization Manager (RHV-M). RHV-M provides centralized, enterprise-grade management for the physical and logical resources within the virtualized RHV environment. RHV-H is a minimal, light-weight operating system based on Red Hat Enterprise Linux that is optimized for the ease of setting up physical servers as RHV hypervisors. For more information on RHV, see the documentation [here](#). The following figure provides an overview of RHV.



Starting with Cisco APIC release 3.1, Cisco ACI supports VMM integration with Red Hat Virtualization environments. The RHV VMM domain in Cisco APIC is connected to RHV-M and directly associated with a data center object. All the RHV-H clusters under this data center are considered part of the VMM domain. Cisco ACI automatically creates logical networks in RHV- M when the EPGs are attached to the RHV VMM domain in ACI. RHV hosts that are part of a Red Hat VMM domain can use Linux bridge or Open vSwitch as its virtual switch. This integration simplifies and automates networking configuration on RHV-M, saving a lot of manual work for system and network administrators.

## Workflow

The following workflow is used to set up the virtual environment. Each of these steps might involve several individual tasks.

1. Install and configure Nexus 9000 switches in ACI mode and APIC software on the UCS C-series server. Refer to the [Install and Upgrade documentation](#) for detailed steps.
2. Configure and setup the ACI fabric by referring to the [documentation](#).
3. Configure tenants, application profiles, bridge domains, and EPGs required for NetApp HCI nodes. NetApp recommends using one BD to one EPG framework, except for iSCSI. See the documentation [here](#) for more details. The minimum set of EPGs required are in-band management, iSCSI, VM motion, VM-data network, and native.
4. Create the VLAN pool, physical domain, and AEP based on the requirements. Create the switch and interface profiles and policies for vPCs and individual ports. Then attach the physical domain and configure the static paths to the EPGs. see the [configuration guide](#) for more details. This table lists best practices for integrating ACI with Linux bridge on RHV.

# PC/VPC Interface Policy Group - HCI-RHVH01

Properties

Name: HCI-RHVH01

Description: optional

Link Aggregation Type: Port Channel **VPC**

Link Level Policy: 10G-Auto  

CDP Policy: CDP-Disabled  

MCP Policy: select a value 

CoPP Policy: select a value 

LLDP Policy: LLDP-Enabled  

STP Interface Policy: select a value 

Egress Data Plane Policing Policy: select a value 

Ingress Data Plane Policing Policy: select a value 

Priority Flow Control Policy: select a value 

Fibre Channel Interface Policy: select a value 

Slow Drain Policy: select a value 

Port Channel Policy: LACP-Active  



Use a vPC policy group for interfaces connecting to NetApp HCI storage and compute nodes.

5. Create and assign contracts for tightly controlled access between workloads. For more information on configuring the contracts, see the guide [here](#).
6. Install and configure the NetApp HCI Element cluster. Do not use NDE for this install; rather, install a standalone Element cluster on the HCI storage nodes. Then configure the required volumes for installation of RHV. Install RHV on NetApp HCI. Refer to [RHV on NetApp HCI NVA](#) for more details.
7. RHV installation creates a default management network called ovirtmgmt. Though VMM integration of Cisco ACI with RHV is optional, leveraging VMM integration is preferred. Do not create other logical networks manually. To use Cisco ACI VMM integration, create a Red Hat VMM domain and attach the VMM domain to all the required EPGs, using Pre- Provision Resolution Immediacy. This process automatically creates corresponding logical networks and vNIC profiles. The vNIC profiles can be directly

used to attach to hosts and VMs for their communication. The networks that are managed by Cisco ACI are in the format <tenant-name>|<application-profile-name>|<epg-name> tagged with a label of format `aci_<rhv-vmm-domain-name>`. See [Cisco's whitepaper](#) for creating and configuring a VMM domain for RHV. Also, see this table for best practices when integrating RHV on NetApp HCI with Cisco ACI.



Except for ovirtmgmt, all other logical networks can be managed by Cisco ACI.

Network > Networks

Network:									<input type="button" value="New"/>	<input type="button" value="Import"/>	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
Name	Comment	Data Center	Description	Role	VLAN Tag	QoS	Nam	Label	Provider	MTU		
HCI-Infra AFF-A200 Aff-NFS		Default		■	1569	-	ad_hci-aci-rhv			9000		
HCI-Infra HCI HCI-IB-Mgmt		Default		■	1567	-	aci_hci-aci-rhv			Default (1500)		
HCI-Infra HCI HCI-iSCSI		Default		■	1568	-	aci_hci-aci-rhv			9000		
HCI-Infra HCI HCI-VM-motion		Default		■	1634	-	aci_hci-aci-rhv			Default (1500)		
HCI-Infra HCI HCI-VM-network		Default		■	1570	-	aci_hci-aci-rhv			Default (1500)		
ovirtmgmt		Default	Management Network	■	3201	-	-			Default (1500)		
quarantine		Default		■	666	-	aci_hci-aci-rhv			Default (1500)		
uplinkNetwork		Default	uplinkNetwork	■	-	-	-			Default (1500)		

Setup Host hci-aci-rtp-rvhv01.cie.netapp.com Networks X

Drag to make changes

Interfaces

- bond0
- eno1
- ens14f1
- eno2

Assigned Logical Networks

- HCI-Infra|AFF-A200|... (VLAN 1569)
- HCI-Infra|HCI|HCI-iS... (VLAN 1568)
- HCI-Infra|HCI|HCI-V... (VLAN 1634)
- HCI-Infra|HCI|HCI-V... (VLAN 1570)
- ovirtmgmt (VLAN 3201)

Networks Labels

[New Label]

- aci\_hci-aci-rhv
- HCI-Infra|AFF-A200|... (VLAN 1569)
- HCI-Infra|HCI|HCI-IB-Mg... (VLAN 1567)
- HCI-Infra|HCI|HCI-i... (VLAN 1568)
- HCI-Infra|HCI|HCI-V... (VLAN 1634)

Verify connectivity between Host and Engine i

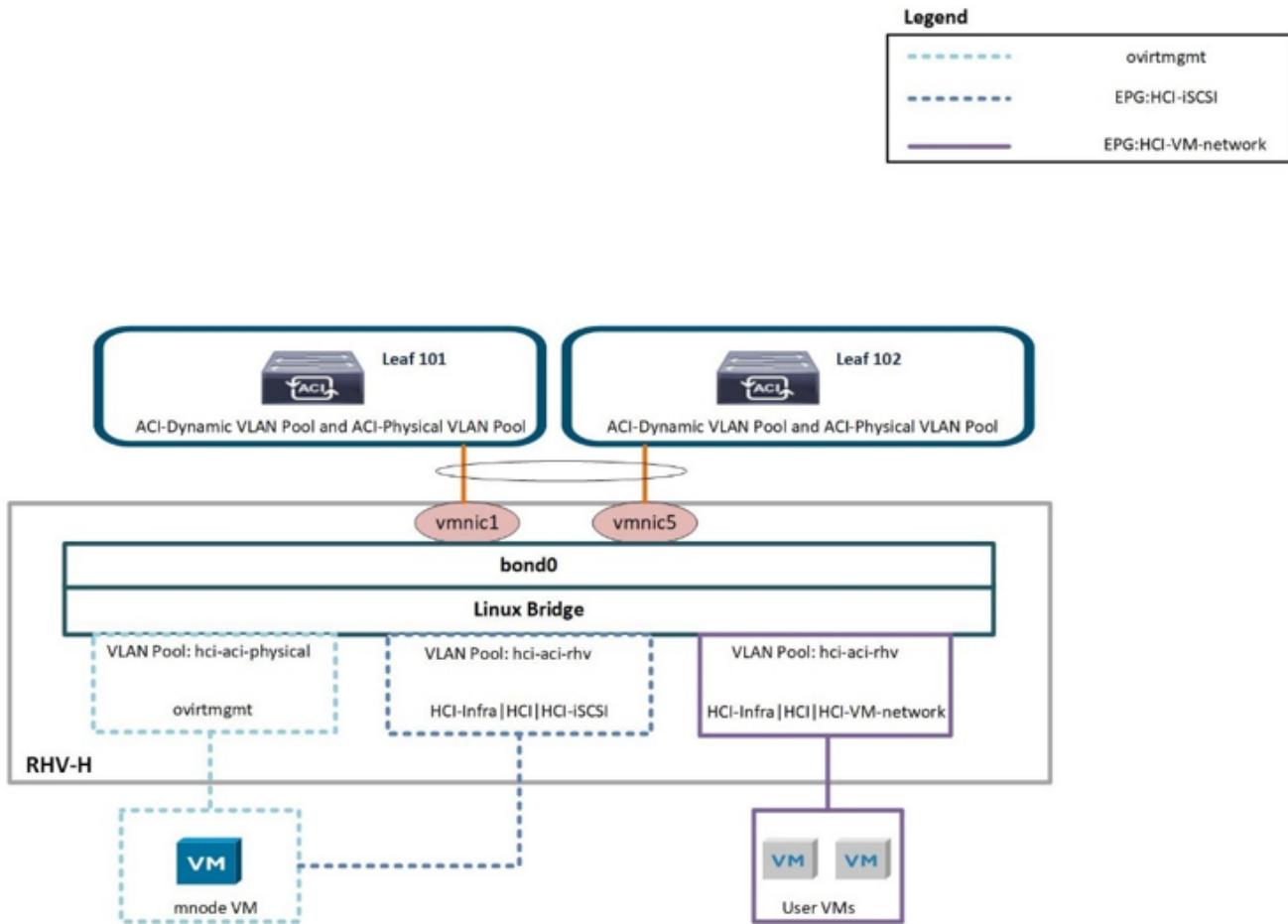
Save network configuration i

OK Cancel

The networking functionality for RHVH hosts in this solution is provided by Linux bridge.

## Linux Bridge

Linux Bridge is a default virtual switch on all Linux distributions that is usually used with KVM/QEMU-based hypervisors. It is designated to forward traffic between networks based on MAC addresses and thus is regarded as a layer-2 virtual switch. For more information, see the documentation [here](#). The following figure depicts the internal networking of Linux Bridge on RHV-H (as tested).



The following table outlines the necessary parameters and best practices for configuring and integrating Cisco ACI with Linux Bridge on RHV hosts.

Resource	Configuration considerations	Best Practices
Endpoint groups	<ul style="list-style-type: none"> <li>Separate EPG for native VLAN</li> <li>Static binding of interfaces towards HCI storage and compute nodes in native VLAN EPG to be on 802.1P mode</li> <li>Static binding of vPCs required on In-band management EPG and iSCSI EPG before RHV installation</li> </ul>	<ul style="list-style-type: none"> <li>Separate VLAN pool for VMM domain with dynamic allocation turned on</li> <li>Contracts between EPGs to be well defined. Allow only required ports for communication.</li> <li>Use unique native VLAN for discovery during Element cluster formation</li> <li>For EPGs corresponding to port-groups being attached to VMkernel ports, VMM domain to be attached with 'Pre-Provision' for Resolution Immediacy</li> </ul>
Interface policy	<ul style="list-style-type: none"> <li>One vPC policy group per RHV-H host</li> <li>One vPC policy group per NetApp HCI storage node</li> <li>LLDP enabled, CDP disabled</li> </ul>	<ul style="list-style-type: none"> <li>Recommended to use vPC towards RHV-H hosts</li> <li>Use 'LACP Active' for the port-channel policy</li> <li>Use only 'Graceful Convergence' and 'Symmetric Hashing' control bits for port-channel policy</li> <li>Use 'Layer4 Src-port' load balancing hashing method for port-channel policy</li> <li>Recommended to use vPC with LACP Active port-channel policy for interfaces towards NetApp HCI storage nodes</li> </ul>
VMM Integration	<ul style="list-style-type: none"> <li>Do not migrate host management logical interfaces from ovirtmgmt to any other logical network</li> </ul>	<ul style="list-style-type: none"> <li>iSCSI host logical interface to be migrated to iSCSI logical network managed by ACI VMM integration</li> </ul>



Except for the ovirtmgmt logical network, it is possible to create all other infrastructure logical networks on Cisco APIC and map them to the VMM domain. 'ovirtmgmt' logical network uses the static path binding on the In-band management EPG attached with the physical domain.

[Next: KVM on RHEL: NetApp HCI with Cisco ACI](#)

### KVM on RHEL: NetApp HCI with Cisco ACI

KVM (for Kernel-based Virtual Machine) is an open-source full virtualization solution for

Linux on x86 hardware such as Intel VT or AMD-V. In other words, KVM lets you turn a Linux machine into a hypervisor that allows the host to run multiple, isolated VMs.

KVM converts any Linux machine into a type-1 (bare-metal) hypervisor. KVM can be implemented on any Linux distribution, but implementing KVM on a supported Linux distribution—like Red Hat Enterprise Linux—expands KVM’s capabilities. You can swap resources among guests, share common libraries, and optimize system performance.

## Workflow

The following high-level workflow was used to set up the virtual environment. Each of these steps might involve several individual tasks.

1. Install and configure Nexus 9000 switches in ACI mode, and install and configure APIC software on a UCS C-series server. See the [Install and Upgrade documentation](#) for detailed steps.
2. Configure and set up the ACI fabric by referring to the [documentation](#).
3. Configure the tenants, application profiles, bridge domains, and EPGs required for NetApp HCI nodes. NetApp recommends using a one-BD-to-one-EPG framework except for iSCSI. See the documentation [here](#) for more details. The minimum set of EPGs required are in-band management, iSCSI, VM Motion, VM-data network, and native.
4. Create the VLAN pool, physical domain, and AEP based on the requirements. Create the switch and interface profiles and policies for vPCs and individual ports. Then attach the physical domain and configure the static paths to the EPGs. See the [configuration guide](#) for more details. Also see this table <link> for best practices for integrating ACI with Open vSwitch on the RHEL–KVM hypervisor.

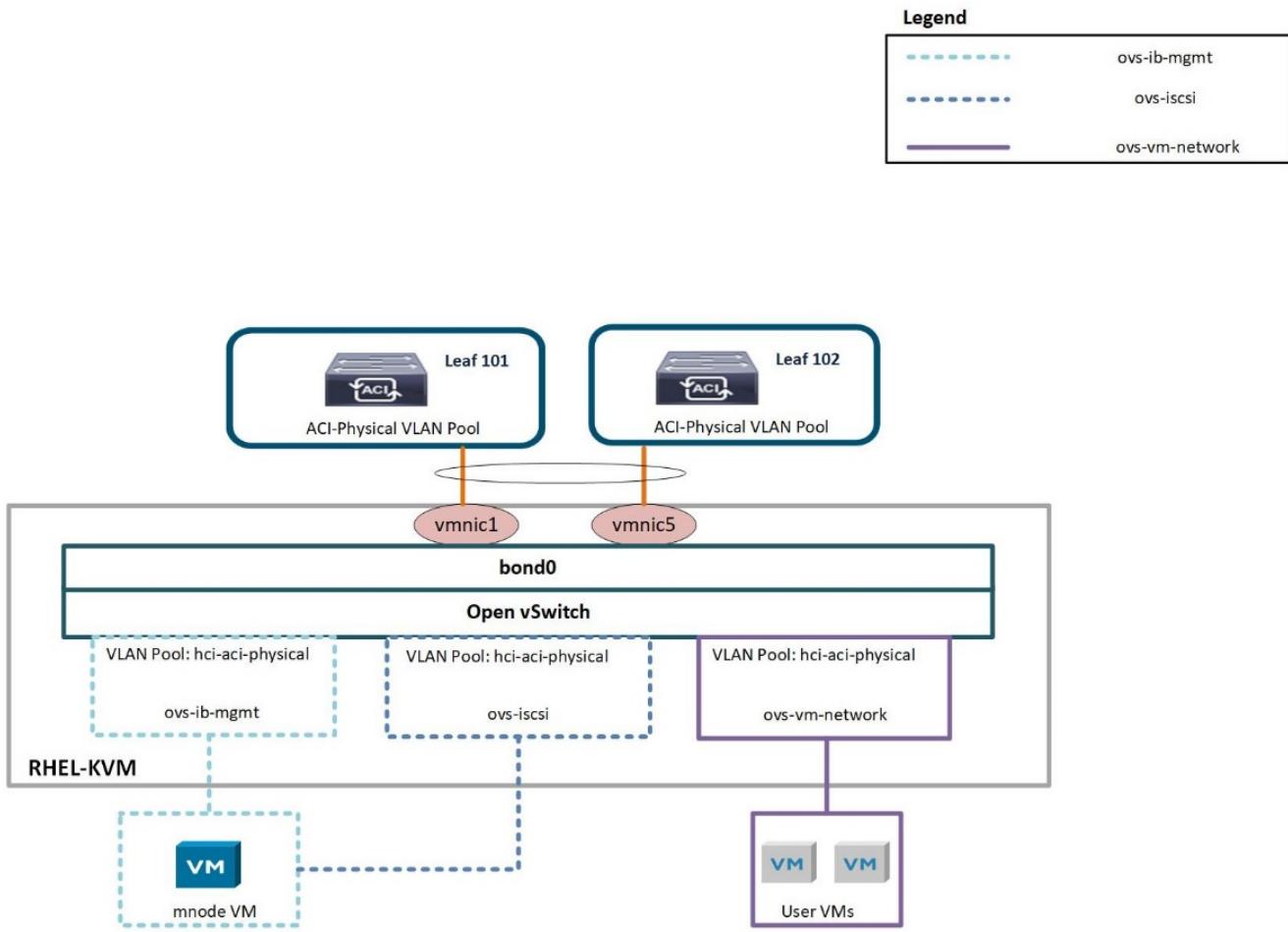


Use a vPC policy group for interfaces connecting to NetApp HCI storage and compute nodes.

5. Create and assign contracts for tightly-controlled access between workloads. For more details on configuring the contracts, see the guide [here](#).
6. Install and configure a NetApp HCI Element cluster. Do not use NDE for this installation; rather, install a standalone Element cluster on HCI storage nodes. Then configure the required volumes for the installation of RHEL. Install RHEL, KVM, and Open vSwitch on the NetApp HCI compute nodes. Configure storage pools on the hypervisor using Element volumes for a shared storage service for hosts and VMs. For more details on installation and configuration of KVM on RHEL, see the [Red Hat documentation](#). See the [OVS documentation](#) for details on configuring Open vSwitch.
7. RHEL KVM hypervisor’s Open vSwitch cannot be VMM integrated with Cisco ACI. Physical domain and static paths must be configured on all required EPGs to allow the required VLANs on the interfaces connecting the ACI leaf switches and RHEL hosts. Also configure the corresponding OVS bridges on RHEL hosts and configure VMs to use those bridges. The networking functionality for the RHEL KVM hosts in this solution is achieved using Open vSwitch virtual switch.

## Open vSwitch

Open vSwitch is an open-source, enterprise-grade virtual switch platform. It uses virtual network bridges and flow rules to forward packets between hosts. Programming flow rules work differently in OVS than in the standard Linux Bridge. The OVS plugin does not use VLANs to tag traffic. Instead, it programs flow rules on the virtual switches that dictate how traffic should be manipulated before forwarded to the exit interface. Flow rules determine how inbound and outbound traffic should be treated. The following figure depicts the internal networking of Open vSwitch on an RHEL-based KVM host.



The following table outlines the necessary parameters and best practices for configuring Cisco ACI and Open vSwitch on RHEL based KVM hosts.

Resource	Configuration Considerations	Best Practices
Endpoint groups	<ul style="list-style-type: none"> <li>Separate EPG for native VLAN</li> <li>Static binding of interfaces towards HCI storage and compute nodes in native VLAN EPG to be on 802.1P mode</li> <li>Static binding of vPCs required on in-band management EPG and iSCSI EPG before KVM installation</li> </ul>	<ul style="list-style-type: none"> <li>Separate VLAN pool for physical domain with static allocation turned on</li> <li>Contracts between EPGs to be well defined. Allow only required ports for communication.</li> <li>Use unique native VLAN for discovery during Element cluster formation</li> </ul>

Resource	Configuration Considerations	Best Practices
Interface Policy	<ul style="list-style-type: none"> <li>• One vPC policy group per RHEL host</li> <li>• One vPC policy group per NetApp HCI storage node</li> <li>• LLDP enabled, CDP disabled</li> </ul>	<ul style="list-style-type: none"> <li>• NetApp recommends using vPC towards RHV-H hosts</li> <li>• Use LACP Active for the port-channel policy</li> <li>• Use only Graceful Convergence and Symmetric Hashing control bits for port-channel policy</li> <li>• Use Layer4 Src-Port load-balancing hashing method for port-channel policy</li> <li>• NetApp recommends using vPC with LACP Active port-channel policy for interfaces towards NetApp HCI storage nodes</li> </ul>

Next: [ONTAP on AFF: NetApp HCI and Cisco ACI](#)

## ONTAP on AFF: NetApp HCI and Cisco ACI

NetApp AFF is a robust storage platform that provides low-latency performance, integrated data protection, multiprotocol support, and nondisruptive operations. Powered by NetApp ONTAP data management software, NetApp AFF ensures nondisruptive operations, from maintenance to upgrades to complete replacement of your storage system.

NetApp ONTAP is a powerful storage operating system with capabilities like inline compression, nondisruptive hardware upgrades, and cross-storage import. A NetApp ONTAP cluster provides a unified storage system with simultaneous data access and management of Network File System (NFS), Common Internet File System (CIFS), iSCSI, Fibre Channel (FC), Fibre Channel over Ethernet (FCoE), and NVMe/FC protocols. ONTAP provides robust data protection capabilities, such as NetApp MetroCluster, SnapLock, Snapshot copies, SnapVault, SnapMirror, SyncMirror technologies and more. For more information, see the [ONTAP documentation](#).

To extend the capabilities of storage to file services and add many more data protection abilities, ONTAP can be used in conjunction with NetApp HCI. If NetApp ONTAP already exists in your environment, you can easily integrate it with NetApp HCI and Cisco ACI.

### Workflow

The following high-level workflow was used to set up the environment. Each of these steps might involve several individual tasks.

1. Create a separate bridge domain and EPG on ACI for NFS and/or other protocols with the corresponding subnets. You can use the same HCI-related iSCSI EPGs.
2. Make sure you have proper contracts in place to allow inter-EPG communication for only the required ports.

3. Configure the interface policy group and selector for interfaces towards AFF controllers. Create a vPC policy group with the LACP Active mode for port-channel policy.

## PC/VPC Interface Policy Group - Storage-AFF-01

Name: Storage-AFF-01

Description: optional

Link Aggregation Type: Port Channel **VPC**

Link Level Policy: 10G-Auto

CDP Policy: CDP-Enabled

MCP Policy: select a value

CoPP Policy: select a value

LLDP Policy: LLDP-Enabled

STP Interface Policy: select a value

Egress Data Plane Policing Policy: select a value

Ingress Data Plane Policing Policy: select a value

Priority Flow Control Policy: select a value

Fibre Channel Interface Policy: select a value

Slow Drain Policy: select a value

Port Channel Policy: LACP-Active

4. Attach both a physical and VMM domain to the EPGs created. Attach the vPC policy as static paths and, in the case of the Cisco AVE virtual switch, use Native switching mode when you attach the VMM domain.

VMware/hci-vmware-ave

VMM Domain: On Demand

native

VLAN: 1

Update Cancel

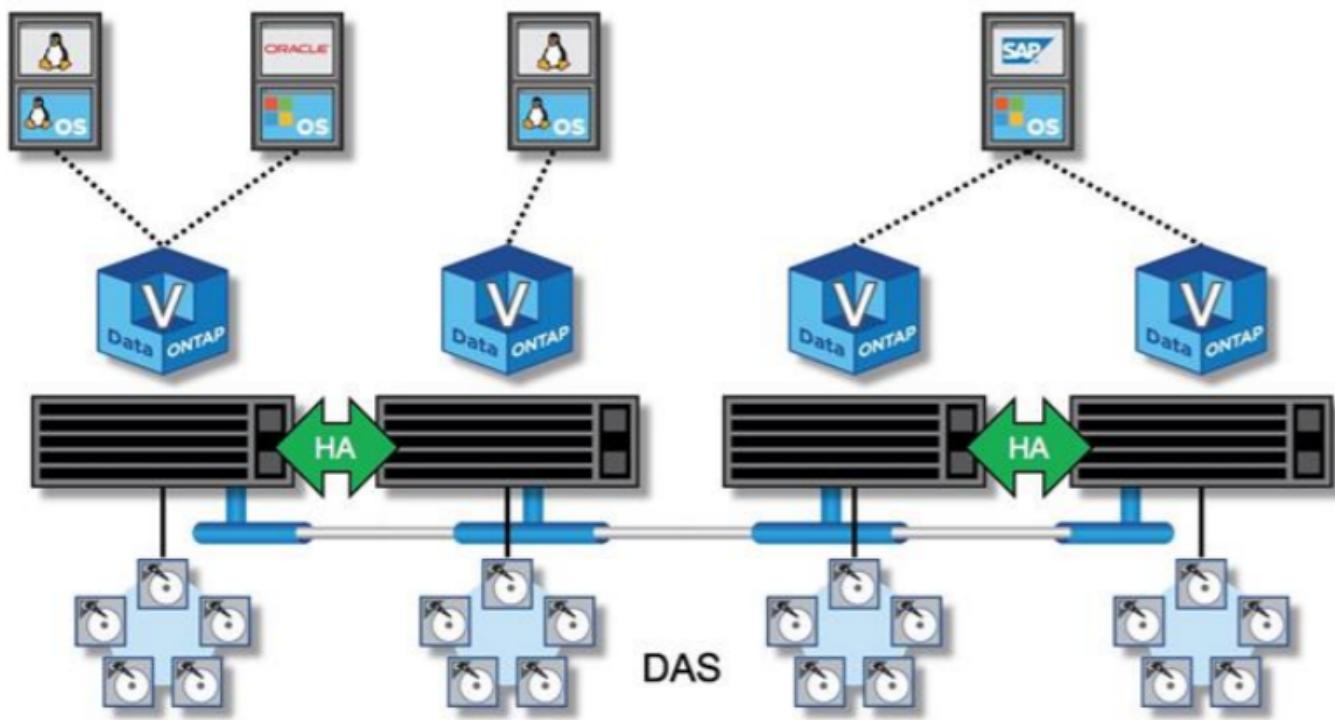
5. Install and configure an ONTAP cluster on the AFF controllers. Then create and configure NFS and/or iSCSI volumes/LUNs. See the [AFF and ONTAP documentation](#) for more information.
6. Create a VMkernel adapter (in the case of VMware ESXi) or a logical interface (in the case of RHV-H and RHEL-KVM hosts) attaching the NFS (or other protocols) port group or logical network.
7. Create additional datastores, storage domains, or storage pools on hypervisors (VMware, RHV, or KVM) using AFF storage.

Next: ONTAP Select with VMware vSphere: NetApp HCI and Cisco ACI

## ONTAP Select with VMware vSphere: NetApp HCI and Cisco ACI

NetApp ONTAP Select is the NetApp solution for software-defined storage (SDS), bringing enterprise-class storage management features to the software-defined data center. ONTAP Select extends ONTAP functionality to extreme edge use cases including IoT and tactical servers as a software-defined storage appliance that acts as a full storage system. It can run as a simple VM on top of a virtual environment to provide a flexible and scalable storage solution.

Running ONTAP as software on top of another software application allows you to leverage much of the qualification work done by the hypervisor. This capability is critical for helping us to rapidly expand our list of supported platforms. Also, positioning ONTAP as a virtual machine (VM) allows customers to plug into existing management and orchestration frameworks, which allows rapid provisioning and end-to-end automation from deployment to sunsetting. The following figure provides an overview of a four-node ONTAP Select instance.



Deploying ONTAP Select in the environment to use the storage offered by NetApp HCI extends the capabilities of NetApp Element.

### Workflow

The following workflow was used to set up the environment. In this solution, we deployed a two-node ONTAP Select cluster. Each of these steps might involve several individual tasks.

1. Create an L2 BD and EPG for the OTS cluster's internal communication and attach the VMM domain to the EPG in the Native switching mode (in case of a Cisco AVE virtual switch) with Pre-Provision Resolution Immediacy.

# EPG - HCI-Select-Internal



## Properties

Contract Exception Tag:

QoS class: Unspecified

Custom QoS: select a value

Data-Plane Policer: select a value

Intra EPG Isolation: Enforced

Preferred Group Member:

Flood on Encapsulation:

Configuration Status: applied

Configuration Issues:

Label Match Criteria: AtleastOne

Bridge Domain: SELECT-Internal

Resolved Bridge Domain: HCI-Infra/SELECT-Internal

Monitoring Policy: select a value

FHS Trust Control Policy: select a value

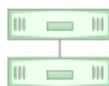
2. Verify that you have a VMware vSphere license.
3. Create a datastore that hosts OTS.
4. Deploy and configure ONTAP Select according to the [ONTAP Select documentation](#).

## **i** Cluster Details

Name	hci-aci-ontap-select	Cluster Size	2 node cluster (1 HA Pairs)
ONTAP Image Version	9.7	Licensing	evaluation
IPv4 Address	172.22.9.81	Cluster MTU	9000
Netmask	255.255.255.0	Domain Names	cie.netapp.com
Gateway	172.22.9.1	Server IP Addresses	10.61.184.251, 10.61.184.252
Mediator Status	HA Active	NTP Server	10.61.184.48
Last Refresh	-		

## **i** Node Details

### > HA Pair 1



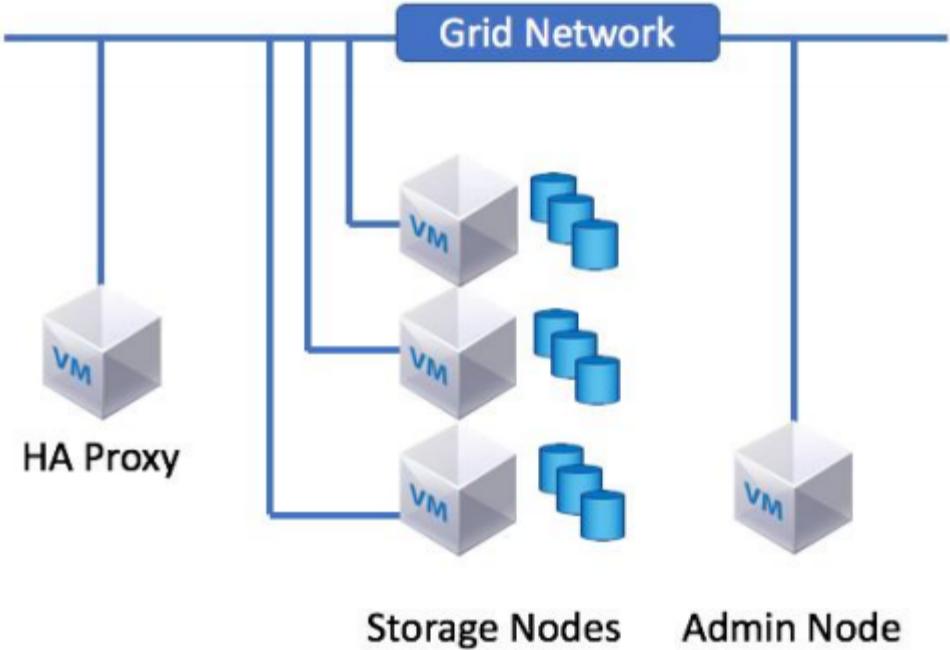
Node 1	hci-aci-ontap-select... — 2 TB +	Host 1	172.22.9.61 — (Small (4 CPU, 16 GB Memory))
Node 2	hci-aci-ontap-select... — 2 TB +	Host 2	172.22.9.60 — (Small (4 CPU, 16 GB Memory))

5. Create additional datastores using ONTAP Select to make use of additional capabilities.

Next: [StorageGRID with VMware vSphere: NetApp HCI and Cisco ACI](#)

## StorageGRID with VMware vSphere: NetApp HCI and Cisco ACI

StorageGRID is a robust software-defined, object-based storage platform that stores and manages unstructured data with a tiered approach along with intelligent policy-driven management. It allows you to manage data while optimizing durability, protection, and performance. StorageGRID can also be deployed as hardware or as an appliance on top of a virtual environment that decouples storage management software from the underlying hardware. StorageGRID opens a new realm of supported storage platforms, increasing flexibility and scalability. StorageGRID platform services are also the foundation for realizing the promise of the hybrid cloud, letting you tier and replicate data to public or other S3-compatible clouds. See the [StorageGRID](#) documentation for more details. The following figure provides an overview of StorageGRID nodes.



#### Workflow

The following workflow was used to set up the environment. Each of these steps might involve several individual tasks.

1. Create an L2 BD and EPG for the grid network used for internal communication between the nodes in the StorageGRID system. However, if your network design for StorageGRID consists of multiple grid networks, then create an L3 BD instead of an L2 BD. Attach the VMM domain to the EPG with the Native switching mode (in the case of a Cisco AVE virtual switch) and with Pre-Provision Resolution Immediacy. The corresponding port group is used for the grid network on StorageGRID nodes.

# EPG - GridNetwork



Properties

QoS class: Unspecified

Custom QoS: select a value

Data-Plane Policer: select a value

Intra EPG Isolation: Enforced **Unenforced**

Preferred Group Member: **Exclude** Include

Flood on Encapsulation: **Disabled** Enabled

Configuration Status: applied

Configuration Issues:

Label Match Criteria: AtleastOne

Bridge Domain: GridNetwork-BD 

Resolved Bridge Domain: HCI-Infra/GridNetwork-BD

Monitoring Policy: select a value

FHS Trust Control Policy: select a value

EPG Contract Master: [empty]

2. Create a datastore to host the StorageGRID nodes.
3. Deploy and configure StorageGRID. For more details on installation and configuration, see the [StorageGRID documentation](#). If the environment already has ONTAP or ONTAP Select, then you can use the NetApp Fabric Pool feature. Fabric Pool is an automated storage tiering feature in which active data resides on local high-performance solid-state drives (SSDs) and inactive data is tiered to low-cost object storage. It was first made available in NetApp ONTAP 9.2. For more information on Fabric Pool, see the documentation [here](#).

Next: [Validation Results](#)

## Validation Results

We used the iPerf tool for testing network throughput, and the baseline expectation was

that the test systems should achieve throughput within 10% of the maximum line rate. Test results for different virtual switches is indicated in the following table.

For storage IOPS subsystem measurement, we used the IOmeter tool. The baseline expectation was that the test systems should achieve read/write throughput within 10% of the maximum. Test results for different hypervisors is indicated in the following table.

We considered the following scenarios for the network line rate and storage IOPS testing:

### **VMware**

- VMs on a NetApp HCI datastore (with and without micro-segmentation)
- VMs on a NetApp ONTAP datastore
- VMs on a NetApp ONTAP Select datastore

### **Red Hat Virtualization**

- VMs on a NetApp HCI datastore
- VMs on a NetApp ONTAP datastore

### **KVM (RHEL)**

- VMs on a NetApp HCI datastore

### **Miscellaneous**

- One VM on RHV with a NetApp HCI datastore and one VM on VMware vSphere with a NetApp ONTAP datastore.

Hypervisor	Virtual Switch	iPerf	IOmeter	Micro-segmentation
VMware	VDS	Pass	Pass	Pass
RHV	Linux Bridge	Pass	Pass	N/A
RHEL-KVM	Open vSwitch	Pass	Pass	N/A

[Next: Where to Find Additional Information](#)

## **Where to Find Additional Information**

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp HCI Documentation

<https://www.netapp.com/us/documentation/hci.aspx>

- Cisco ACI Documentation

<https://www.cisco.com/c/en/us/solutions/data-center-virtualization/application-centric-infrastructure/index.html>

- Cisco Nexus 9000 Series Switches

<http://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html>

- NetApp AFF A-series

<http://www.netapp.com/us/products/storage-systems/all-flash-array/aff-a-series.aspx>

- ONTAP Documentation

<https://docs.netapp.com/ontap-9/index.jsp>

- ONTAP Select Documentation

<https://docs.netapp.com/us-en/ontap-select/>

- StorageGRID Documentation

<https://docs.netapp.com/sgws-113/index.jsp>

- Red Hat Virtualization

[https://access.redhat.com/documentation/en-us/red\\_hat\\_virtualization/4.3/](https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.3/)

- VMware vSphere

<https://docs.vmware.com/en/VMware-vSphere/index.html>

- VMware vCenter Server

<http://www.vmware.com/products/vcenter-server/overview.html>

- NetApp Interoperability Matrix Tool

<http://now.netapp.com/matrix>

- Cisco ACI Virtualization Compatibility Matrix

<https://www.cisco.com/c/dam/en/us/td/docs/Website/datacenter/aci/virtualization/matrix/virtmatrix.html>

- VMware Compatibility Guide

<http://www.vmware.com/resources/compatibility>

# Solution Automation

## NetApp Solution Automation

### Introduction

One of the objectives of validating and architecting solutions is to make the solution easily consumable. Therefore, it is paramount that the deployment and configuration of infrastructure and/or applications delivered through our solutions is simplified through automation. NetApp is committed to simplifying solution consumption through automation using RedHat Ansible.

Ansible is an open-source automation engine that helps IT teams automate application deployment, cloud provisioning, configuration management, and many other IT needs. Ansible is agentless and does not require a custom security infrastructure. You can manage the automation of multiple systems from your control system remotely via SSH making it a robust solution for IT teams looking to automate their tedious and repetitive IT needs.

If you are new to NetApp solution automation, you can use the following sections to set up your Ansible controller.

For more information about RedHat Ansible, see the documentation [here](#).

## Setup the Ansible control node (For CLI based deployments)

### NetApp Solution Automation

#### Procedure

1. Requirements for the Ansible control node,:
  - a. A RHEL/CentOS machine with the following packages installed:
    - i. Python3
    - ii. Pip3
    - iii. Ansible (version greater than 2.10.0)
    - iv. Git

If you have a fresh RHEL/CentOS machine without the above requirements installed, follow the below steps to setup that machine as the Ansible control node:

1. Enable the Ansible repository for RHEL-8/RHEL-7
  - a. For RHEL-8 (run the below command as root)

```
subscription-manager repos --enable ansible-2.9-for-rhel-8-x86_64-
rpms
```

- b. For RHEL-7 (run the below command as root)

```
subscription-manager repos --enable rhel-7-server-ansible-2.9-rpms
```

## 2. Create a .sh file

```
vi setup.sh
```

## 3. Paste the below content in the file

```
#!/bin/bash
echo "Installing Python ----->"
sudo yum -y install python3 >/dev/null
echo "Installing Python Pip ----->"
sudo yum -y install python3-pip >/dev/null
echo "Installing Ansible ----->"
python3 -W ignore -m pip --disable-pip-version-check install ansible
>/dev/null
echo "Installing git ----->"
sudo yum -y install git >/dev/null
```

## 4. Make the file executable

```
chmod +x setup.sh
```

## 5. Run the script (as root)

```
./setup.sh
```

# NetApp Solution Automation

## Procedure

### 1. Requirements for the Ansible control node,:;

- A Ubuntu/Debian machine with the following packages installed:
  - Python3
  - Pip3
  - Ansible (version greater than 2.10.0)
  - Git

If you have a fresh Ubuntu/Debian machine without the above requirements installed, follow the below steps to setup that machine as the Ansible control node:

1. Create a .sh file

```
vi setup.sh
```

2. Paste the below content in the file

```
#!/bin/bash
echo "Installing Python ----->"
sudo apt-get -y install python3 >/dev/null
echo "Installing Python Pip ----->"
sudo apt-get -y install python3-pip >/dev/null
echo "Installing Ansible ----->"
python3 -W ignore -m pip --disable-pip-version-check install ansible
>/dev/null
echo "Installing git ----->"
sudo apt-get -y install git >/dev/null
```

3. Make the file executable

```
chmod +x setup.sh
```

4. Run the script (as root)

```
./setup.sh
```

## NetApp solution automation

### Procedure

This section describes the steps required to configure the parameters in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.

1. Configure the inventory.

- a. Navigate to Resources → Inventories → Add and click Add Inventory.
- b. Provide name and organization details and click Save.
- c. In the Inventories page, click the inventory resources you just created.
- d. If there are any inventory variables, paste them into the variables field.
- e. Go to the Groups sub-menu and click Add.
- f. Provide the name of the group, copy in the group variables (if necessary), and click Save.
- g. Click the group created, go to the Hosts sub-menu and click Add New Host.

- h. Provide the hostname and IP address of the host, paste in the host variables (if necessary), and click Save.
2. Create credential types. For solutions involving ONTAP, Element, VMware, or any other HTTPS-based transport connection, you must configure the credential type to match the username and password entries.
- Navigate to Administration → Credential Types and click Add.
  - Provide the name and description.
  - Paste the following content into the Input Configuration:

```
fields:
- id: username
type: string
label: Username
- id: password
type: string
label: Password
secret: true
- id: vsadmin_password
type: string
label: vsadmin_password
secret: true
```

- a. Paste the following content into the Injector Configuration:

```
extra_vars:
password: '{{ password }}'
username: '{{ username }}'
vsadmin_password: '{{ vsadmin_password }}'
```

- Configure credentials.
  - Navigate to Resources → Credentials and click Add.
  - Enter the name and organization details.
  - Select the correct credential type; if you intend to use the standard SSH login, select the type Machine or alternatively select the custom credential type that you created.
  - Enter the other corresponding details and click Save.
- Configure the project.
  - Navigate to Resources → Projects and click Add.
  - Enter the name and organization details.
  - Select Git for the Source Control Credential Type.
  - Paste the source control URL (or git clone URL) corresponding to the specific solution.
  - Optionally, if the Git URL is access controlled, create and attach the corresponding credential in Source Control Credential.

- f. Click Save.
- 3. Configure the job template.
  - a. Navigate to Resources → Templates → Add and click Add Job Template.
  - b. Enter the name and description.
  - c. Select the Job type; Run configures the system based on a playbook and Check performs a dry run of the playbook without actually configuring the system.
  - d. Select the corresponding inventory, project, and credentials for the playbook.
  - e. Select the playbook that you would like to run as a part of the job template.
  - f. Usually the variables are pasted during runtime. Therefore, to get the prompt to populate the variables during runtime, make sure to tick the checkbox Prompt on Launch corresponding to the Variable field.
  - g. Provide any other details as required and click Save.
- 4. Launch the job template.
  - a. Navigate to Resources → Templates.
  - b. Click the desired template and then click Launch.
  - c. Fill in any variables if prompted on launch and then click Launch again.

# NetApp Solutions Change Log

Recent changes to the NetApp Solutions collateral. The most recent changes are listed first.

Date	Solution Area	Description of change
08/05/2021	Containers	Added a new technology integration to NVA-1160 - NetApp Astra Control Center on Red Hat OpenShift
06/16/2021	Containers	Added a new video demo: Installing OpenShift Virtualization - Red Hat OpenShift with NetApp
06/16/2021	Containers	Added a new video demo: Deploying a Virtual Machine with OpenShift Virtualization - Red Hat OpenShift with NetApp
06/14/2021	SQL Server	Added solution: Microsoft SQL Server on Azure NetApp Files
06/11/2021	Containers	Added a new video demo: Workload Migration - Red Hat OpenShift with NetApp
06/09/2021	Containers	Added a new use-case to NVA-1160 - Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp
05/28/2021	Containers	Added a new use-case to NVA-1160 - OpenShift Virtualization with NetApp ONTAP
05/27/2021	Containers	Added a new use-case to NVA-1160 - Multitenancy on OpenShift with NetApp ONTAP
05/26/2021	Containers	Added NVA-1160 - Red Hat OpenShift with NetApp
05/25/2021	Containers Blog	Added blog: Installing NetApp Trident on Red Hat OpenShift – How to solve the Docker ‘toomanyrequests’ issue!
05/19/2021	NetApp Solutions	Added link to FlexPod solutions portal
05/19/2021	AI Control Plane	Converted solution from PDF to HTML
05/17/2021	NetApp Solutions	Added Solution Feedback tile to main page
05/11/2021	Oracle Database	Added automated deployment of Oracle 19c for ONTAP on NFS
05/10/2021	VMware Virtualization	Video: How to use vVols with NetApp and VMware Tanzu Basic, part 3
05/06/2021	Enterprise Database	Added link to Oracle 19c RAC Databases on FlexPod DataCenter with Cisco UCS and NetApp AFF A800 over FC
05/05/2021	Enterprise Database	Added FlexPod Oracle NVA (1155) and Automation video
05/03/2021	Desktop Virtualization	Added link to FlexPod Desktop Virtualization solutions
04/30/2021	VMware Virtualization	Video: How to use vVols with NetApp and VMware Tanzu Basic, part 2
04/26/2021	Hybrid Cloud Blogs	Added blog: Using VMware Tanzu with ONTAP to accelerate your Kubernetes journey
04/06/2021	NetApp Solutions	Added "About this Repository"

03/31/2021	AI Use Cases	Added TR-4886: AI Inferencing at the Edge: NetApp ONTAP with Lenovo ThinkSystem Solution Design
03/29/2021	Modern Data Analytics	Added NVA-1157: Apache Spark Workload with NetApp Storage Solution
03/23/2021	VMware Virtualization	Video: How to use vVols with NetApp and VMware Tanzu Basic, part 1
03/09/2021	AI, DB, Data Protection	Added E-Series content; categorized AI content
03/04/2021	Solution Automation	New content: getting started with NetApp solution automation
02/18/2021	VMware Virtualization	Added VMware vSphere for ONTAP TR
02/16/2021	AI Edge Inferencing	Added automated deployment steps
02/03/2021	SAP, SAP HANA	Added landing page for all SAP and SAP HANA content
02/01/2021	VDI with NetApp VDS	Added content for GPU nodes
01/06/2021	NetApp AI	New solution: NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches (Design and Deployment)
01/05/2021	NetApp HCI with Cisco ACI	Update: Outline NDE Easy Scale procedure for VMware deployments
12/22/2020	NetApp Solutions	Initial release of NetApp Solutions repository

# About this Repository

Brief introduction of the NetApp Solutions repository - where to find specific solutions and how to use this repository.

## Navigation of the Repository

Navigation of the repository is managed by the main sidebar which is presented on the left side of the page. Solutions are categorized into higher level technical areas defined as the "technology towers" for NetApp Solutions.

### Overview of Technology Towers

Section	Description
Artificial Intelligence	Collection of AI based solutions. Solutions are sub-classified into one of the following categories: AI Converged Infrastructures Data Pipelines, Data Lakes and Management Use Cases
Modern Data Analytics	Collection of Data Analytics solutions (e.g. Splunk SmartStore, Apache Spark, etc.)
Hybrid Cloud / Virtualization	Collection of hybrid cloud core solutions. Solutions are sub-classified into one of the following categories: VMware Virtualization VMware Private Cloud Red Hat Private Cloud Workload Performance
Virtual Desktops	Collection of end user computing solutions. Solutions are sub-classified into one of the following categories: Virtual Desktop Service (VDS) VMware Horizon Citrix Virtual Apps and Desktops Virtual Desktop Applications
Containers	Collection of container based solutions. Solutions are sub-classified into one of the following categories: Red Hat OpenShift Google Anthos
Business Applications	Collection of business applications solutions. Solutions are sub-classified into one of the following categories: SAP
Enterprise Database	Collection of database solutions. Solutions are sub-classified into one of the following categories: SAP HANA Oracle Microsoft SQL Server
Data Protection & Security	Collection of data protection and security solutions.

Infrastructure	Collection of infrastructure based solutions.
Solution Automation	Overview of getting started with solution automation using Red Hat Ansible.

## PDF Generation

A PDF can be generated for any solution or section of a solution by utilizing the PDF section located above the main sidebar on the left side of the page. The generate PDF capability follows the tree structure from the specified section and generates a PDF from that section of the navigation sidebar and includes all HTML files.

**NOTE:** PDF generation will not include documents which are references to files (e.g. links to PDF content) which are not located in this repository. PDF generation only includes content that is rendered as HTML.

The following options may be present when you expand the PDF section:

Section	Description
Site	Generates a PDF of the entire NetApp Solutions site. The resulting PDF will contain all solutions content in a single PDF file.
Sections	Displays a list of the sections that have been expanded using the navigation menu. In order to generate a PDF for an entire solution, select the section that represents the top level of the specific solution.
This Page	Generates a PDF of the currently displayed web page. It will <b>only</b> generate the currently visible page.

## Change Log

All major changes to the repository (new solutions, major updates, new videos / demos, etc.) are tracked in the [change log](#).

## Feedback

Please use [this link](#) to request changes to content or provide feedback on the content. Please be as specific as possible to ensure that your feedback is addressed appropriately.

# **Legal notices**

Legal notices provide access to copyright statements, trademarks, patents, and more.

## **Copyright**

<http://www.netapp.com/us/legal/copyright.aspx>

## **Trademarks**

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

<http://www.netapp.com/us/legal/netapptmlist.aspx>

## **Patents**

A current list of NetApp owned patents can be found at:

<https://www.netapp.com/us/media/patents-page.pdf>

## **Privacy policy**

<https://www.netapp.com/us/legal/privacypolicy/index.aspx>

## **Open source**

Notice files provide information about third-party copyright and licenses used in NetApp software.

## **Copyright Information**

Copyright © 2021 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

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

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

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

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

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## **Trademark Information**

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