NetApp Verified Architecture

NetApp HCI for Private Cloud with Red Hat

NVA Design

Amit Borulkar, NetApp
Gregory Charot, Red Hat
May 2019 | NVA-1133-DESIGN | Version 1.0

Abstract

NetApp® HCI for Private Cloud with Red Hat is a prevalidated, best-practice data center architecture for deploying an OpenStack-based private cloud environment in a reliable and risk-free manner. This reference architecture also showcases running Red Hat OpenShift Container Platform on OpenStack for microservices-based workloads.

In partnership with

Red Hat

NetApp®

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

To meet growing customer demands and requests for new features, enterprises are moving to cloud-based consumption models that enable organizations to operate in an agile manner. As organizations continue to accelerate their cloud journey, a hybrid-cloud or multicloud strategy is a preferred choice for enterprise IT. Factors such as the nature of the workloads, security and compliance adherence, and costs can often dictate the placement of workloads in the private cloud or public cloud. The on-premises cloud must be interoperable and compatible with the other cloud regions. A cloud model can be described by these five core principles:

- **On-demand self-service.** Resources such as virtual machines (VMs), containers, storage, and networks are easily provisioned and released with minimal service provider interaction.
- **Resource pooling.** The underlying compute, network, and storage resources are abstracted to serve multiple consumers in a multitenant model.
- **Broad network access.** Capabilities are available over the network and accessed through standard mechanisms
- **Rapid elasticity.** Additional resources are added or removed based on the consumer's requirement.
- **Measured service.** Resource usage are monitored, controlled, and reported for both the provider and consumer of the utilized service.

Building the underlying infrastructure to support these characteristics imposes certain design requirements. Special care must be taken to ensure that the operational aspect of managing the cloud infrastructure shouldn’t impact business continuity. Any downtime would impact both financially as well as business trust wise.

To address these challenges, NetApp and Red Hat have partnered together to provide an enterprise-grade platform to enable turnkey private cloud deployment in a reliable manner. NetApp HCI provides an intuitive, API-driven, programmable agile platform with enterprise-class features such as storage efficiencies and self-healing capabilities for complete high availability (HA) and guaranteed performance. Red Hat OpenStack Platform version 13, engineered with Red Hat hardened Queens code delivers a stable release for a production scale environment. Red Hat OpenStack Platform version 13 adopters have an advantage of immediate access to bug fixes and critical security patches, tight integration with Red Hat’s enterprise security features including SELinux, and a steady release cadence between OpenStack versions. Also, Red Hat OpenStack Platform version 13 is a long-life release with up to three years of standard support and an additional, optional two years of extended life-cycle support (ELS).

This reference architecture also validates Red Hat OpenShift running on OpenStack. Red Hat OpenShift provides enterprise Kubernetes bundled CI/CD pipelines, automated builds, and deploy, which enable developers to focus on application logic while leveraging all best-of-class enterprise infrastructure. This approach provides a unified platform to run virtualized workloads and microservices in a reliable manner.

2 Program Summary

2.1 NetApp Verified Architecture

NetApp HCI for Private Cloud with Red Hat is a prevalidated, best-practice data center architecture for deploying OpenStack private cloud infrastructure at enterprise scale. This document describes the enterprise requirements for deploying production grade OpenStack, Red Hat OpenShift on OpenStack, the various design choices and technical requirements to achieve a flexible, and a reliable and predictable infrastructure that scales independently with your application demands. The architecture described in this document is codesigned and coengineered by subject matter experts (SMEs) from within NetApp and Red Hat to provide the advantages of open source innovation with enterprise robustness.

NetApp HCI provides the widely recognized benefits of hyperconverged solutions including lower TCO, ease of purchasing, deployment, and growth and management for virtualized workloads. However, NetApp HCI is different because it allows IT to scale storage and compute separately. NetApp HCI with Red Hat OpenStack Platform provides an out-of-box, cloud-like experience while simplifying day-to-day operations and management. Together, these technology solutions provide for ease of procurement, deployment, and ongoing management and growth of your business-critical hardware, virtual resources, and enterprise applications.

2.2 NetApp HCI Design Principles

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, 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 quality of service (QoS) limits that are available natively with NetApp Element® software. NetApp Element software allows the granular control of every application and volume, eliminates noisy neighbors, and satisfies performance SLAs. NetApp HCI multitenancy capabilities can help eliminate more than 90% of traditional performance-related problems[1].

Flexible

Previous generations of hyperconverged infrastructures required fixed resource ratios, limiting deployments to four- to eight-node configurations. NetApp HCI, however, scales 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. With NetApp HCI, your licensing costs are reduced. NetApp HCI is available in mix-and-match small, medium, and large storage and compute configurations. The architectural design choices that are offered enable you to confidently scale on your terms, making HCI viable for core 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, made up of storage nodes, compute nodes, or both. A minimum configuration is two chassis with eight nodes, consisting of four storage nodes and four compute nodes.

Red Hat OpenStack Platform supports adding additional nodes in a nondisruptive manner by using the Red Hat OpenStack Platform Director. It is easy to resize the OpenShift VMs and to create new worker nodes as your scale needs change. You can also upgrade operating systems nondisruptively with rollback through NetApp Snapshot™ copies and maintain HA with native HA and antiaffinity rules.

Simple

A driving imperative within the IT community is to automate all routine tasks, eliminating the risk of user error while freeing up resources to focus on more interesting, higher-value projects. NetApp HCI can help

[1] Source: https://www.netapp.com/us/resources/esg-lab-report-quantifying-the-economic-value-of-asolidfire-deployment.

your IT department become more agile and responsive by simplifying deployment and ongoing management. The deployment mechanism for this reference architecture, including Mellanox switches, is completely automated by using Ansible and Red Hat OpenStack Platform Director.

3 Solution Overview

NetApp HCI for Private Cloud with Red Hat enables a fully integrated and production-grade OpenStack deployment that offers the following capabilities:

- Multitenancy and quotas for resources
- Isolation of networks with overlapping IP address space
- Block devices for workloads with performance guarantees
- Object storage
- Networking constructs such as virtual routers
- Provider networks to communicate with devices in the data center
- Load balancing as a service
- Booting instances with persistent storage
- Automated and fully supported OpenShift deployment for running containerized workloads
- Security compliance

3.1 Target Audience

The target audience for the solution includes the following groups:

- Enterprise IT cloud administrators
- Service providers
- NetApp and Red Hat partners
- DevOps practitioners

3.2 Solution Technology

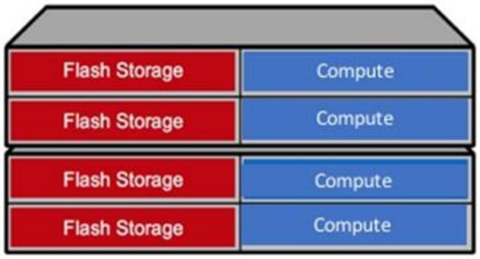
NetApp HCI

NetApp HCI is an enterprise-scale hybrid cloud infrastructure solution that delivers compute and storage resources in an agile, scalable, easy-to-manage two-rack unit (2RU or 1RU), four-node building block. This solution is based on the following minimum configuration (as shown in Figure 1):

- NetApp H-Series all-flash storage nodes running NetApp Element software
- NetApp H-Series compute nodes
- Red Hat OpenStack Platform 13 which provides enterprise grade hardened OpenStack cloud

For details about and technical specifications for compute and storage nodes in NetApp HCI, see the NetApp HCI datasheet.

Figure 1) NetApp HCI minimum configuration for private cloud with Red Hat.



Red Hat OpenStack Platform

Red Hat OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable private OpenStack cloud. The Red Hat OpenStack Platform infrastructure as a service (IaaS) cloud is implemented by a collection of control services that manage the 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 CLI and API that allow full automation capabilities to administrators and end users.

This reference architecture is based on Red Hat OpenStack Platform version13, which is a long-term support release of OpenStack from Red Hat.

Figure 2 provides a high-level overview of core OpenStack services and their relationship to each other.

Figure 2) OpenStack components.

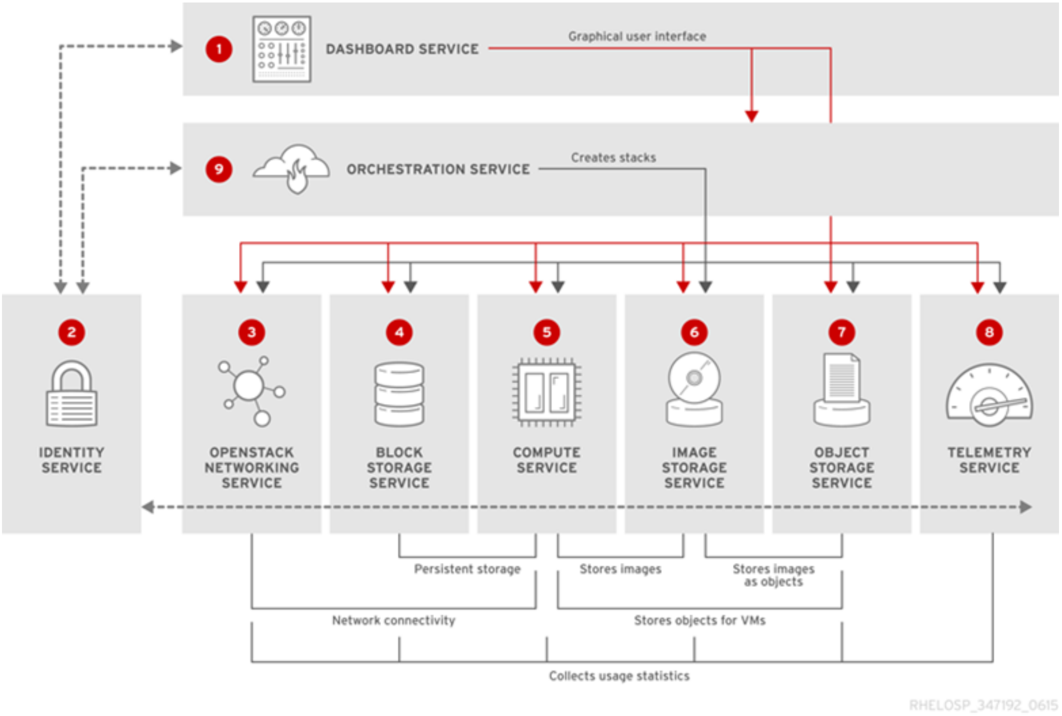


Table 1 describes each OpenStack component.

Table 1) OpenStack components.

Service	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	Service that provides connectivity between the interfaces of OpenStack services.
Block storage	Cinder	Service that manages persistent block storage volumes for virtual machines.
Compute	Nova	Service that manages and provisions VMs running on hypervisor nodes.
Image	Glance	Registry service that you use to store resources such as virtual machine images and volume snapshots.
Object storage	Swift	Service that allows users to store and retrieve files and arbitrary data.
Telemetry	Ceilometer	Service that provides measurements of cloud resources
Orchestration	Heat	Template-based orchestration engine that supports automatic creation of resource stacks.

Containerized Services

All the OpenStack Platform services are deployed as containers. This approach not only ensures isolation of services but also enables easy upgrades. Red Hat 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 registry.
- These service containers are managed using docker container runtime.
- These services are deployed, configured, and maintained with Red Hat OpenStack Director.

NetApp Element Software

NetApp Element software provides modular, scalable performance with each storage node delivering guaranteed capacity and throughput to the environment.

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 of the cluster. As a part of the iSCSI login process, the 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 then 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, the 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.

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 default to one minute. If a volume is running below the maximum IOPS, burst credits are accumulated. When performance levels become very high and are pushed their maximum levels, short bursts of IOPS beyond maximum IOPS are allowed on the volume.

Multitenancy

Secure multitenancy is achieved through 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.
- **Virtual routing and forwarding (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 leverages key features to increase 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 stores only 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 and stored in 4K blocks, and after it has been compressed, it 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.

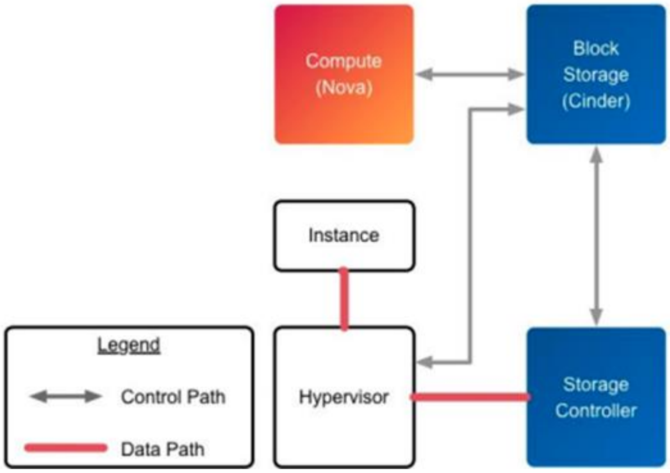
Note: Element software was designed for automation. All the storage features are available through APIs. These APIs are the only method that the UI uses to control the system.

For more information, see the Element Software product page.

OpenStack Cinder

The OpenStack Block Storage service provides management of persistent block storage resources. In addition to acting as secondarily attached persistent storage, you can write images into a Cinder volume for Nova to use as a bootable, persistent root volume for an instance. Figure 3 shows the Cinder data and management paths.

Figure 3) Cinder data path and management path.



As a management service, Cinder controls the provisioning and lifecycle management of block storage volumes. It does not reside in the I/O (data) path between the hypervisor and the storage controller.

Glance

The OpenStack image service (Glance) provides discovery, registration, and delivery services for VM, disk, and server images. Glance provides a RESTful API that allows the querying of VM image metadata as well as the retrieval of the actual image. A stored image can be used as a template to start up new servers quickly and consistently, rather than provisioning multiple servers, installing a server operating system, and individually configuring additional services. Such an image can also be used to store and catalog an unlimited number of backups.

To make sure that VM images are consistent and highly available for the Glance service, Swift storage is used as a backend for Glance.

Swift

Swift is a highly available, distributed, and consistent (ensured by the Swift replication mechanism) object store. Object storage does not present a traditional file system. Instead, object storage is a distributed storage system for static, unstructured data objects such as VM images, photo storage, e-mail storage, backups, and archives. Swift is also configured as a default backend for storing VM images for Glance.

Customers can use NetApp HCI storage to provide block storage (LUNs) for the Swift service and then scale horizontally by adding additional NetApp HCI storage nodes as the object store grows. The NetApp SolidFire hosts the Swift data by using the iSCSI protocol. The three OpenStack controller nodes are also used as Swift nodes and handle account, container, and object services for Swift. In addition, these three nodes also serve as proxy servers for the Swift service.

Swift uses zoning to isolate the cluster into separate partitions and isolate the cluster from failures. Swift data is replicated across the cluster in zones that are as unique as possible. Typically, zones are established using the physical attributes of the cluster, including geographical locations, separate networks, equipment racks, storage subsystems, or even single drives. Zoning allows the cluster to function and tolerate equipment failures without data loss or loss of connectivity to the remaining cluster.

By default, Swift replicates data three times across the cluster. Swift replicates data across zones in a unique way that promotes HA and high durability. Swift chooses a server in an unused zone before it chooses an unused server in a zone that already has a replica of the data.

Octavia: OpenStack Load Balancing as a Service

Octavia is a supported load balancer solution that is recommended to be used in conjunction with OpenShift Container Platform in order to load balance the external incoming traffic and provide a single view of the OpenShift Container Platform master services for the applications. Octavia is automatically deployed as a part of OpenStack deployment.

Heat Orchestration Templates for NetApp HCI

Heat Orchestration Templates (HOT) are an integral part of Red Hat OpenStack Platform’s deployment and configuration. After the overcloud nodes are PXE booted, these templates configure the network interface cards (NICs) on the overcloud nodes, install and configure the relevant packages required for OpenStack deployment. As a part of this verified architecture, HOT were developed to provide a turnkey private cloud boot strapping experience.

Red Hat OpenShift Container Platform on OpenStack

OpenStack provides a IaaS platform to dynamically provision resources at scale. Resources such as VMs, private networks, routers, block devices, and object storage can be provisioned on demand. Furthermore, resources such as Elastic Load Balancers (ELB) are also provisioned by using the Octavia service. Thereby, providing a cloud-native infrastructure platform to deploy OpenShift.

The infrastructure components required for OpenShift such as instances and volumes are deployed by using Ansible roles. These roles call HOT under the hood. After the infrastructure for OpenShift is provisioned, a highly available OpenShift deployment is performed by using standard OpenShift Ansible roles. Red Hat OpenShift 3.11 is fully supported on Red Hat OpenStack Platform 13.

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, the world's leading enterprise Linux distribution. OpenShift is part of the Cloud Native Computing Foundation (CNCF) Certified Kubernetes program, providing portability and interoperability of your container workloads. OpenShift Container Platform 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.
- **Continuous integration and continuous development (CI/CD).** This source-code platform manages build and deployment images at scale.
- **Persistent storage.** By providing support for persistent storage, OpenShift Container Platform allows users to run both stateful applications and cloud-native stateless applications.
- **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) that are scanned and graded with a specific emphasis on security to provide certified, trusted, and secure application containers to end users.

4 Technology Requirements

This section covers the technology requirements for the NetApp HCI for Private Cloud with Red Hat OpenStack Platform solution.

4.1 Hardware Requirements

Table 2 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.

Table 2) Hardware requirements.

Layer	Product Family	Number of Nodes	Details
Compute	NetApp 410C	4	Red Hat OpenStack Platform overcloud nodes
Network	Mellanox SN2010	2	Mellanox switches
Storage	NetApp 410S	4	6 x 960GB Encrypting/nonencrypting

4.2 Software Requirements

Table 3 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.

Table 3) Software requirements.

Layer	Software	Version (or Other Information)
Storage	NetApp Element Software	11.1
–	Trident	19.04
Network	Onyx	3.6.8008
Cloud engine (IaaS)	Red Hat OpenStack Platform	13
Underlying operating system	Red Hat Enterprise Linux	7.6
Container PaaS	Red Hat OpenShift Container Platform	3.11

5 Solution Design

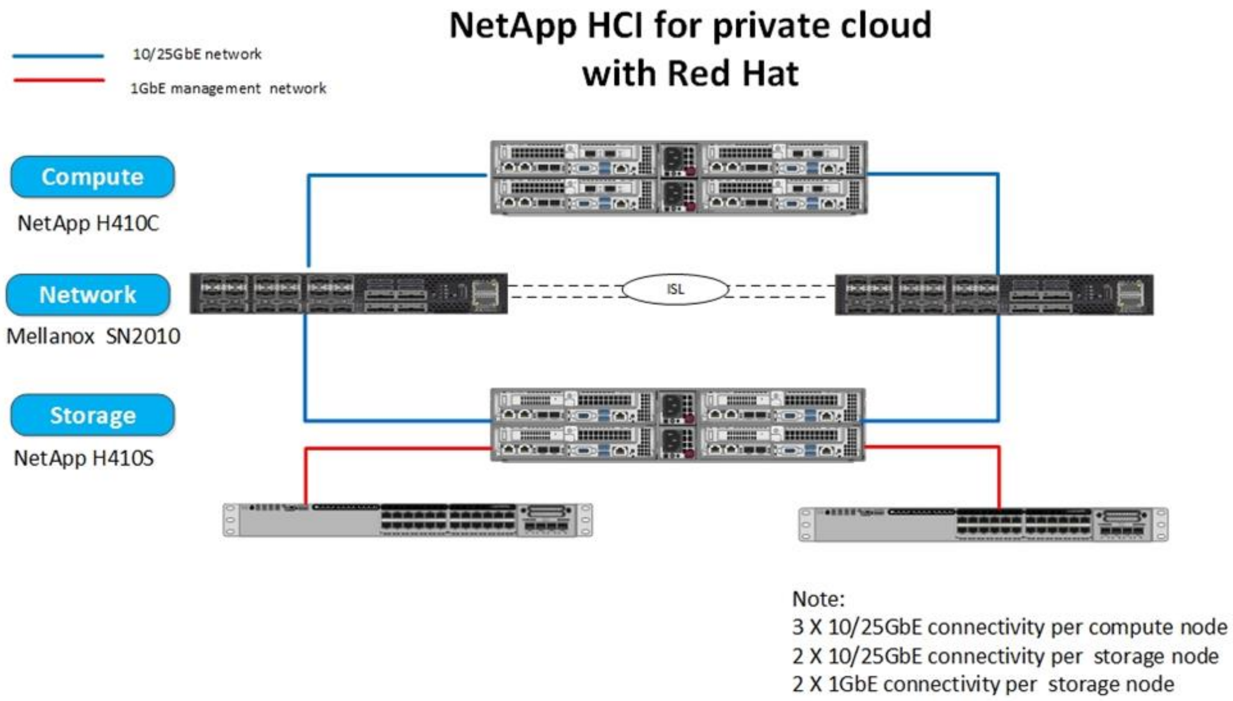
5.1 Architectural Overview

This reference architecture enables an end-to-end 25 Gigabit Ethernet (25GbE) iSCSI network:

- The two 25GbE storage ports on the NetApp H410S nodes are configured in a Link Aggregation Control Protocol (LACP) mode, thereby providing higher throughput and resiliency.
- Each compute node has 3 x 25GbE connected ports:
  - The first port is used for PXE booting.
  - The remaining two ports are connected in an Open vSwitch (OVS) bond with load balancing (slb mode) for higher resiliency and throughput.
- Jumbo frames are configured end-to-end.

NetApp HCI compute nodes are certified for Red Hat OpenStack Platform. This reference architecture requires a minimum of four NetApp HCI compute nodes. The undercloud node can deployed as a VM in an existing customer's environment as long as it meets the minimum requirements.

Figure 4) Solution architectural diagram.


NetApp HCI for private cloud with Red Hat
10/25GbE network
1GbE management network
Compute
NetApp H410C
Network
Mellanox SN2010
Storage
NetApp H410S
ISL
Note:
3 X 10/25GbE connectivity per compute node
2 X 10/25GbE connectivity per storage node
2 X 1GbE connectivity per storage node

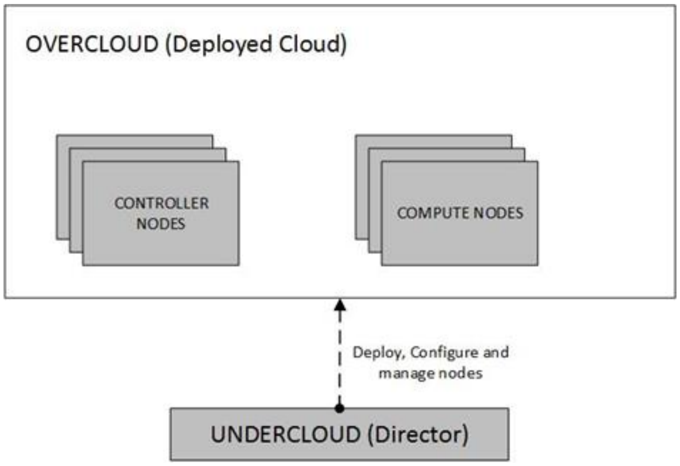
5.2 NetApp HCI Compute Nodes

The Red Hat OpenStack Platform director is a toolset for deploying and managing a complete OpenStack environment. It is based primarily on the OpenStack project TripleO, which is an abbreviation for OpenStack-On-OpenStack. This project takes advantage of OpenStack components to install a fully operational OpenStack environment.

Director is defined as a lifecycle management platform for OpenStack. It was designed from the ground up to bridge the gap between the planning and design (day 0), the installation tasks themselves (day 1), and the ongoing operation, administration, and management of the environment (day 2).

The Red Hat OpenStack Platform director uses two main concepts (as shown in Figure 5): an undercloud and an overcloud. The undercloud installs and configures the overcloud. The overcloud is the cloud that is consumed by the end-users. The undercloud is strictly reserved for operations such as scaling, updating, or upgrading the overcloud.

Figure 5) Undercloud and overcloud.


OVERCLOUD (Deployed Cloud)
CONTROLLER NODES
COMPUTE NODES
Deploy, Configure and manage nodes
UNDERCLOUD (Director)

Undercloud

The undercloud node provisions and manages the overcloud lifecycle. It is a single node, all-in-one OpenStack deployment that consists of OpenStack services such as Keystone, Ironic, Glance, Heat, and Neutron. These undercloud OpenStack services are used to deploy, configure, and manage the lifecycle of the overcloud nodes. For example, the Ironic service deploys the bare-metal nodes and the Neutron manages the network for the bare-metal nodes. Undercloud enables the following functions:

- Environment planning
- Bare-metal system control
- Orchestration
- CLI Tools and a Web UI

Environment Planning

The undercloud node provides planning functions for users to create and assign certain node roles. The undercloud includes a set of default node roles such as compute, controller, and various storage roles. It also provides the ability to define custom roles. This reference architecture uses two types of node roles: the controller node role, which configures the overcloud nodes as an OpenStack controller, and the compute node role, which configures the overcloud node as the OpenStack Nova compute node.

Bare-Metal System Control

The undercloud uses out-of-band management interface, Intelligent Platform Management Interface (IPMI), of each node for power management control and a PXE-based service to discover hardware attributes and install OpenStack to each node.

Note: NetApp HCI 410C nodes support IPMI and iPXE-based provisioning.

Orchestration

The undercloud provides a set of YAML templates that act as a plan for your environment. The undercloud imports this plan and follows their instructions to create the resulting OpenStack environment. This plan also include hooks that allow you to incorporate your own customizations as certain points in the environment creation process. This reference architecture provides HOT to provide a turnkey deployment experience.

CLI Tools and a Web UI

The Red Hat OpenStack Platform director performs these undercloud functions through a terminal-based CLI or a web-based UI.

Overcloud

The resulting Red Hat OpenStack Platform environment deployed by using undercloud is called the overcloud. The overcloud consists of two types of nodes: the controller nodes and compute nodes.

Controller Nodes

The controller nodes run the following OpenStack control-plane components of OpenStack:

- OpenStack Dashboard (Horizon)
- OpenStack Identity (Keystone)
- OpenStack Compute (Nova) API
- OpenStack Networking (Neutron)
- OpenStack Image Service (Glance)
- OpenStack Block Storage (Cinder)
- OpenStack Object Storage (Swift)
- OpenStack Orchestration (Heat)
- OpenStack Telemetry (Ceilometer)
- OpenStack Telemetry Metrics (Gnocchi)
- OpenStack Telemetry Alarming (Aodh)
- OpenStack Telemetry Event Storage (Panko)
- OpenStack Shared File Systems (Manila)
- OpenStack LaaS (Octavia)
- OpenStack Bare Metal (Ironic)
- HAProxy
- MemCached
- Redis
- MariaDB
- RabbitMQ
- OVS
- Pacemaker and Galera for HA services

Compute Nodes

The compute nodes provide the computing resources for the OpenStack environment. A default compute node consists of the following services:

- OpenStack Compute (Nova)
- KVM/QEMU
- OpenStack Telemetry (ceilometer) agent
- OVS

Note: The controller and compute services are deployed as containers on their respective nodes. The deployment and configuration of these services is automated by the Red Hat OpenStack Platform Director.

High Availability

The Red Hat OpenStack Platform director uses a controller node cluster to provide HA services to your OpenStack Platform environment. The Director installs a duplicate set of components on each controller node and manages them together as a single service. This type of cluster configuration provides a fallback in the event of operational failures on a single controller node, which provides OpenStack users with a certain degree of continuous operation.

The OpenStack Platform Director uses the following key pieces of software to manage components on the controller node:

- **Pacemaker (cluster resource manager).** Manages and monitors the availability of clustered services across all nodes in the cluster such as Galera and RabbitMQ.
- **HAProxy.** Provides load balancing and proxy services to the cluster.
- **Galera.** Replicates the Red Hat OpenStack Platform database across the cluster.
- **RabbitMQ.** Message bus used for internal communications between OpenStack components.
- **Memcached.** Provides session caching.

5.3 Network Design

Red Hat OpenStack Platform maps the different services onto separate network traffic types, which are isolated by using virtual LAN (VLANs).

Network isolation among different tenants can be achieved by using `vxlan` encapsulation protocol with the ML2 OVS plug-in. A tenant network (underlay network for VXLAN) is provisioned during the deployment. The overlay networks corresponding to different tenants are provisioned dynamically by the Neutron service; they don't require extra configuration on the switches. The tenant networks isolation is enforced by Linux Kernel namespaces leveraged by Red Hat Enterprise Linux. Figure 6 shows the network topology.

Figure 6) Network topology.

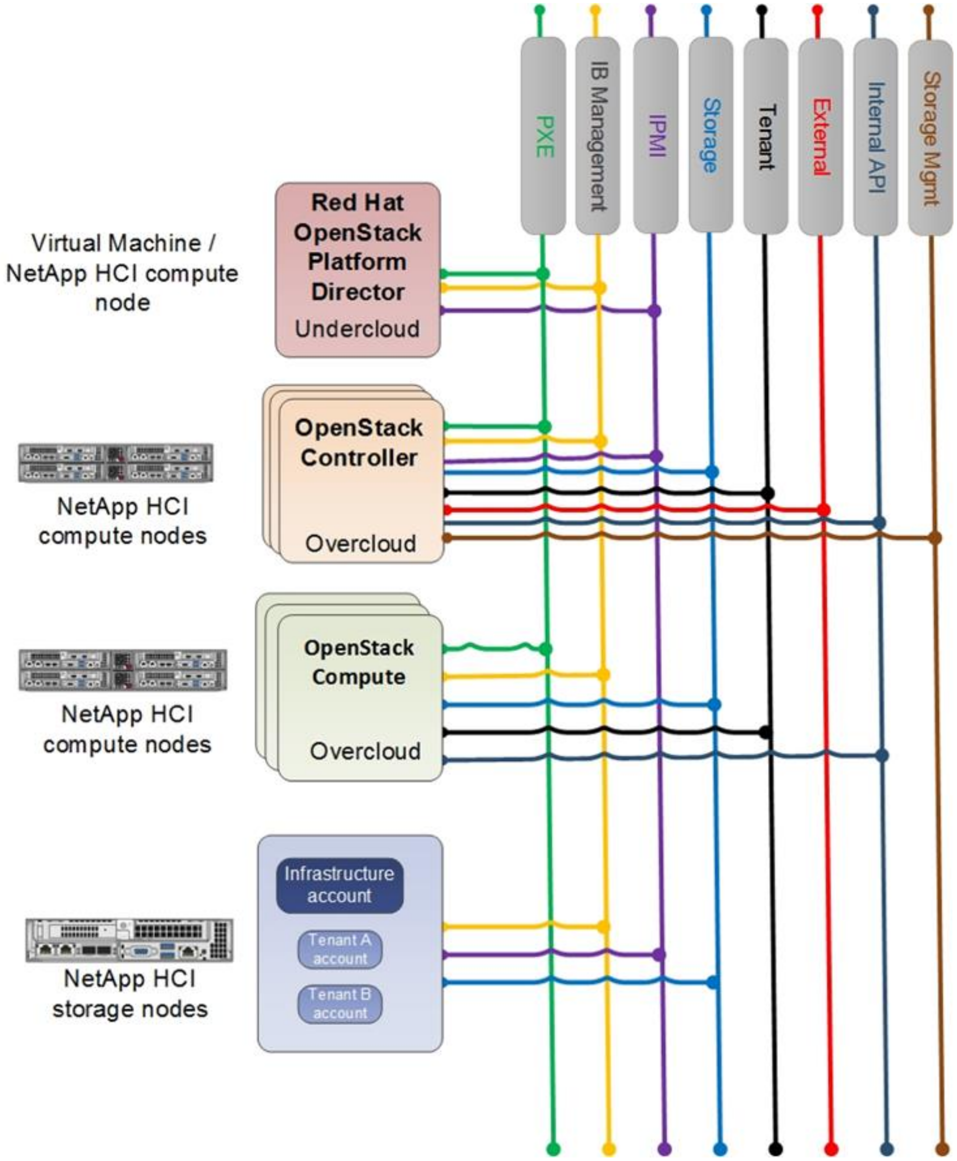


Table 4 lists the OpenStack networks.

Table 4) OpenStack networks.

Network Type	Description	Nodes
IPMI	This network is used for power management of nodes. This network uses the IPMI port on the NetApp HCI compute nodes	All nodes
PXE	The Director uses this network to deploy new nodes over PXE boot and orchestrate the installation of OpenStack Platform on the overcloud bare metal servers. **Note:** The Mellanox NICs (10/25GbE) ports on the compute nodes are enabled for PXE boot.	All nodes

Network Type	Description	Nodes
Internal API	This network is used for communication between the OpenStack services by using API communication, RPC messages, and database communication.	Overcloud nodes
Tenant	Neutron provides each tenant with their own networks using either tunneling (through VXLAN). Network traffic is isolated within each tenant network. Each tenant network has an IP subnet associated with it, and network namespaces means that multiple tenant networks can use the same address range without causing conflicts.	Overcloud nodes
Storage	This network corresponds to the storage traffic to the Element Nodes.	Overcloud nodes
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.	Controller nodes
External	This network hosts the OpenStack Dashboard (Horizon) for graphical system management, the public APIs for OpenStack services.	Controller nodes
IB management	Provides access for system administration functions such as SSH access, DNS traffic, and NTP traffic. This network also acts as a gateway for noncontroller nodes.	Controller nodes

Note: The node introspection and provisioning of nodes is performed over the PXE network. The undercloud node runs a Dynamic Host Configuration Protocol (DHCP) server. Make sure that no other DHCP server is running on the PXE network VLAN.

Note: An OpenStack administrator creates a provider network with the same VLAN segmentation ID as the data center network. A provider network enables connectivity between the entities connected to the provider network and the data center components connected on the same VLAN.

- All the nodes are connected through 1G IPMI interface. This is also the OOB management network for the servers,
- By default, 10/25G interfaces on the NetApp HCI compute nodes are enabled for PXE and iPXE boot. The first 10/25G interface is used to iPXE boot the overcloud node from the Director.
- The LUNs for swift storage are provisioned from Element software and are mounted during the deployment process. Therefore, controller nodes also have access to the storage network.

Figure 7 illustrates the undercloud network interfaces.

Figure 7) Undercloud network interfaces.

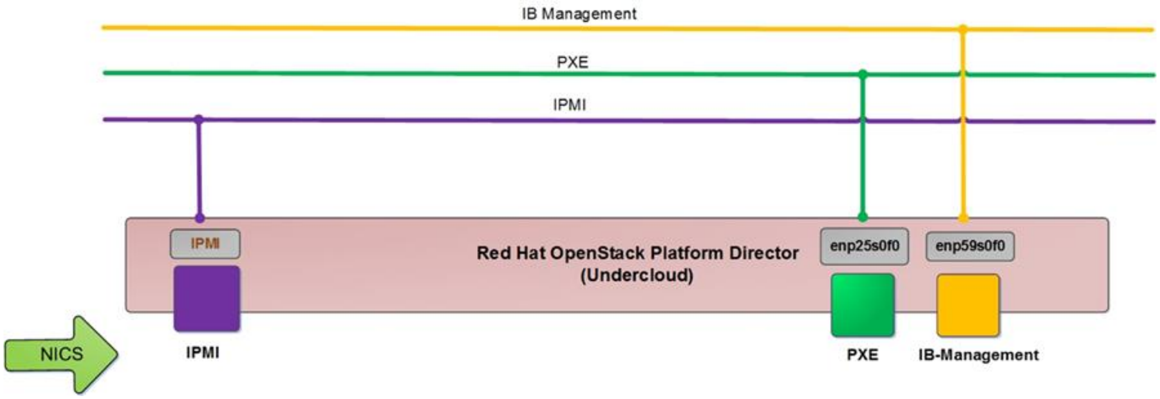


Figure 8 illustrates the overcloud controller network interfaces.

Figure 8) Overcloud controller network interfaces.

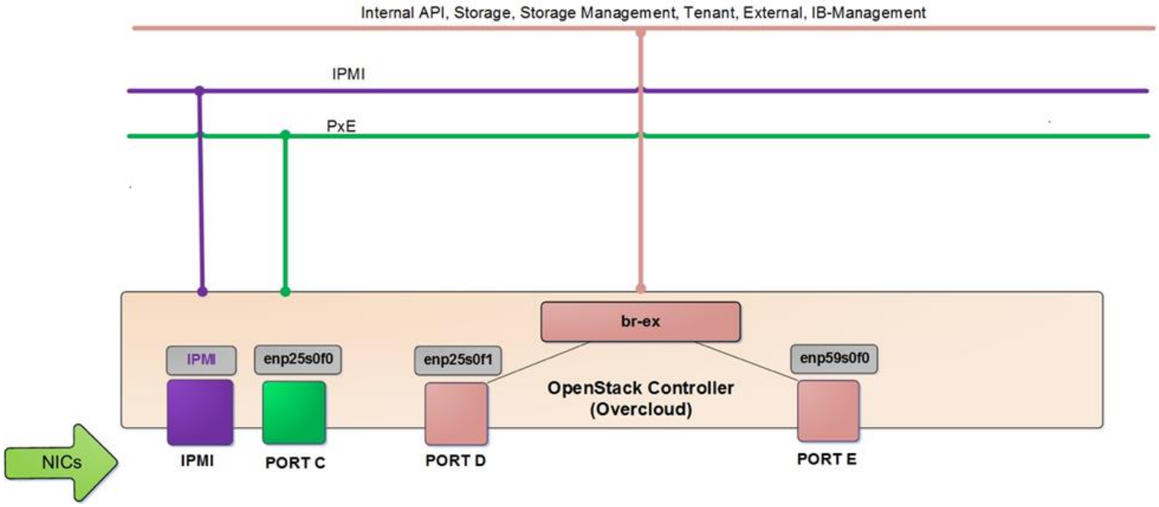
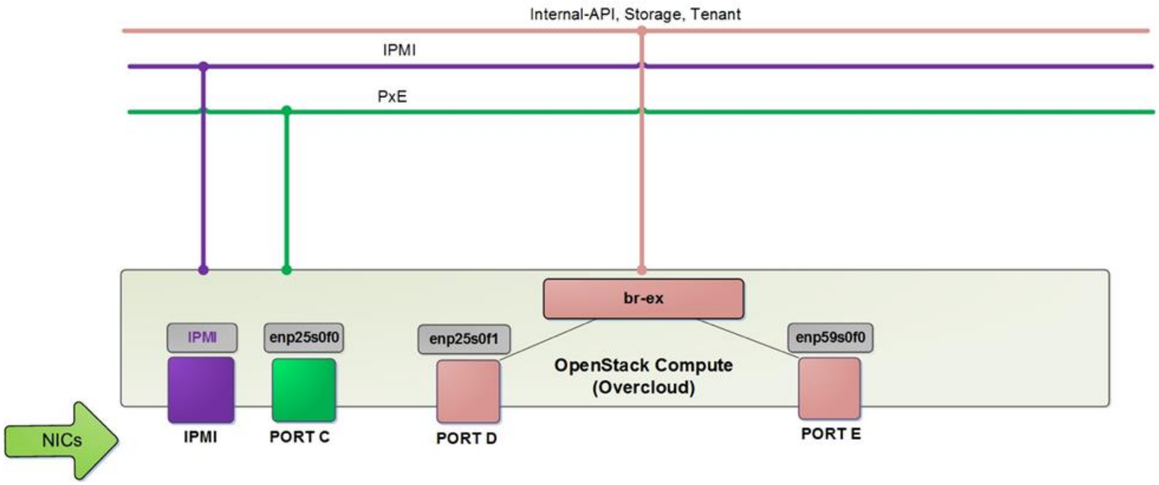


Figure 9 illustrates the overcloud computer network interfaces.

Figure 9) Overcloud computer network interfaces.



Note: The br-ex corresponds to the OVS bridge to indicate that the network interfaces are bonding.

5.4 Storage Design

NetApp Element Cinder Driver

Element software is integrated with OpenStack through the Cinder driver and the standard iSCSI protocol. Element Cinder driver is elegant and straightforward, which allows customers to quickly configure OpenStack for NetApp HCI without the need for additional configuration libraries or add-ons. The maturity of the driver reveals itself in its easy setup for users and the completeness of the following features after setup is complete:

- Clones full environment with no impact to storage capacity.
- Guarantees performance through full QoS.
- Provides multiattachment support.
- Extends in-use volumes (volumes attached to instances).
- Deploys speed with Glance image caching and boot-from-volume capabilities.
- Live-migrates instances between OpenStack compute servers (nondisruptively).
- Easily triages issues through:
  - 1:1 ID mapping between Cinder Vol-ID and SolidFire volume name
  - 1:1 mapping and automated creation of tenant or project IDs as NetApp SolidFire accounts
- Automatically configures the NetApp Element Cinder driver (available through the Puppet classes of TripleO)

Element Cinder drivers are included with all recent OpenStack distributions and are integrated and certified with Red Hat OpenStack Platform Director to enable seamless implementation during the deployment. Heat templates are used to complete the configuration, in which the `cinder.conf` file is generated with the storage service name, IP address, and login credentials. All Nova compute nodes are then automatically configured to use the containerized Cinder.

Element Cinder driver enables to associate the QoS attributes such as minIOPS, maxIOPS, and burstIOPS (as described in the previous section) by using the volume types and the QoS specs available in OpenStack. Element can carry tenant information into the storage subsystem and create unique iSCSI CHAP credentials for each tenant. The Element Cinder driver creates a new account corresponding to each tenant or project in OpenStack and associates all the Cinder volumes (SolidFire volumes) with this account.

Note: The NetApp Element Cinder driver is referenced as NetApp SolidFire Cinder Driver in all the upstream documents and the code.

Image Caching

Element image caching is used to eliminate the copy of Glance images to volumes every time a bootable volume is required. Element image caching is built into the OpenStack Cinder driver and is enabled by default.

The image cache, upon the first copy operation from Glance to a volume (if the property is set), copies the image to a volume stored under the OpenStack administrative user, and then, using the lightweight cloning operation on the Element cluster, clones a copy for the user who has requested the volume. On subsequent image to volume operations for the same image, the volume for the administrator user is checked to make sure it is still current, and if current, a clone operation is performed to quickly present the next volume. This process reduces the provisioning times for instances and enables efficiency for the development and test environments.

Note: An additional cluster SVIP with a different VLAN is created to segregate the infrastructure storage traffic and the Cinder storage traffic.

Storage for OpenStack Swift

NetApp Element storage serves as the storage medium for Swift. As part of this design, a 1TB SolidFire volume (associated with the infrastructure account) is attached to each of the Swift nodes to store the account, container, and object data. The default replica count of three makes sure that the storage is still available and accessible in the event of two Swift node failures. The inherent data-efficiency mechanisms of SolidFire nodes, such as deduplication and compression, make sure that the actual storage utilization is optimized despite the three replica count. Furthermore, SolidFire Helix data protection makes sure that all the data is reconstructed regardless of drive or other component failures in the storage subsystem.

Swift cluster can be easily scaled by adding additional storage to each of the Swift controller nodes. Additional volumes are provisioned from the SolidFire array and attached to each of the Swift controller nodes to layout the account, container, and the object data.

5.5 Red Hat OpenShift Container Platform on OpenStack

OpenStack provides an IaaS platform to dynamically provision resources at scale. In order to maintain isolation, all the OpenShift resources are deployed in a separate project or tenant. The following OpenStack resources are created to deploy OpenShift:

- Project or tenant and a user.

  Note: The role of the user is set to `_member_`.

- The following default tenant quotas for different OpenStack resources are modified:
  - Number of security groups, security group rules, ports, Cinder volumes, and Cinder volume Snapshot copies.
- Instance flavors m1.master and m1.node are created with min vCPU and RAM system requirements for OpenShift.

Table 5) OpenShift instance flavors

Node Type	CPU	RAM	Root Disk	Flavor
Masters	4	16GB	0	m1.master

Node Type	CPU	RAM	Root Disk	Flavor
Nodes	1	8GB	0	m1.node

Note: All the instances for OpenShift Container Platform are booted from Cinder volumes. The value for the root disk in the flavors is set to 0.

Note: Modifications to the HOT to boot the OpenShift instances from Cinder volumes are provided in the deployment guide instructions.

Some of the considerations for deploying OpenShift on OpenStack

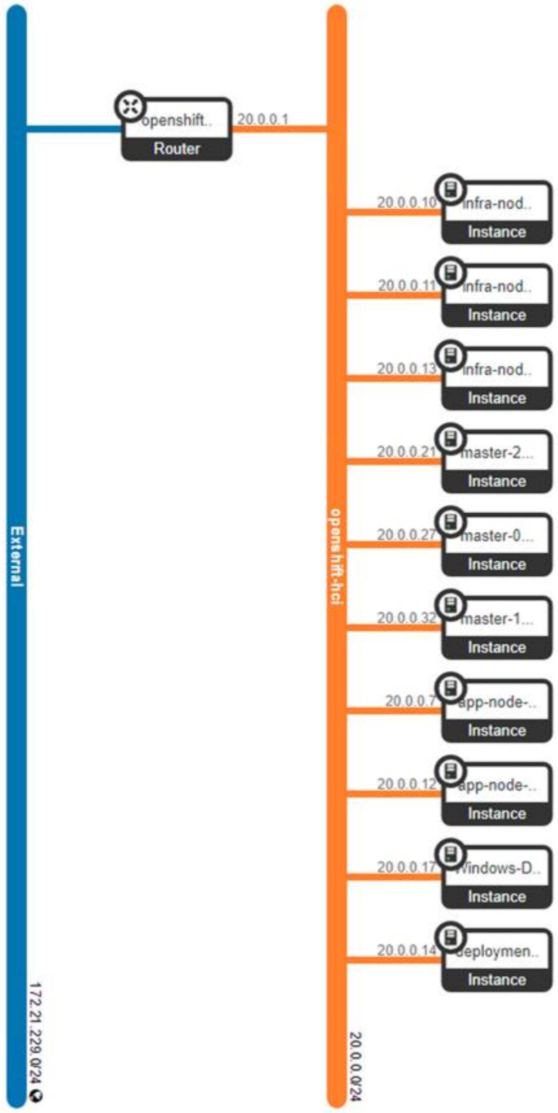
- OpenShift deployment requires a DNS server. An existing DNS server in the customer's environment could be used or the DNS server can be provisioned.
- Octavia is an open source, operator-scale, load-balancing solution designed to work with OpenStack.
- OpenShift on OpenStack creates two load balancers:
  - A load balancer instance to load balance OpenShift API servers (the listeners are configured to include all the masters at port 8443).
  - A load balancer instance to load balancer OpenShift router requests (the listeners are configured to include all the infra nodes running OpenShift router service).

  Note: The deployment and configuration of the load balancers is automated through HOT.
- The wildcard record for OpenShift applications in the DNS server points to the router load balancer instance.

Network Topology for OpenShift on OpenStack

Figure 10 demonstrates the OpenShift network topology. All the instances are connected to a tenant network `openshift-hci`. This network connects to an OpenStack router, through which the instances have egress access to download packages. The DNS instance and the deployment instance are also connected to this network.

Figure 10) Network topology for OpenShift.



Note: You can add NetApp HCI compute nodes and configure the OpenShift Nova instances with antiaffinity rules so that they are scheduled on different physical NetApp HCI compute nodes.

Storage for OpenShift on OpenStack

Mapping different types of storage in OpenShift to OpenStack storage primitives

- All the OpenShift instances are booted from Cinder volumes
- A separate Cinder volume for Docker graph storage
- Cinder volume is deployed for OpenShift registry
- Persistent volumes for logging and monitoring applications

NetApp Trident for Persistent Storage

Cinder is configured as the default dynamic storage provisioner in OpenShift. It allows you to provision PVs dynamically in an out-of-the box manner but doesn’t expose all the Element capabilities such as associating QoS capabilities to PVC, cloning, importing an existing volume using native OpenShift constructs. NetApp Trident is officially supported as the dynamic storage provisioner in OpenShift.

NetApp Trident runs as a pod within OpenShift and watches for PVC requests on the API server. NetApp Trident for Element software allows you to create different storage pools corresponding to different QoS levels, imports existing volumes and enables you to clone volumes.

Trident is deployed and managed by an OpenShift administrator. NetApp recommends that you implement the following best practices when deploying Trident:

- Deploy Trident on OpenShift infrastructure nodes alongside other OpenShift services such as OCR or OpenShift Platform router.
- Create a Trident user in Element software with access privileges to read, volumes, accounts, and ClusterAdmins resource types.
- Use CHAP authentication to provide access to PVs in the nodes. Set UseCHAP to `true` in the Trident storage back-end.
- Depending on the workload requirements, use storage pools to create different storage classes that correspond to different QoS bands.
- For generic use cases without specific QoS requirements, create a default storage class that specifies Trident as the storage provisioner and that uses the default QoS settings in Element software.

For details about storage design considerations for OpenShift on NetApp HCI, see NVA-1124-DESIGN: Red Hat OpenShift Container Platform with NetApp HCI NVA Design.

Note: From an operational perspective, the OpenStack administrator must create a provider network that connects to NetApp Element software and attach it to the OpenShift Infra nodes running Trident.

6 Solution Verification

The following use cases were validated:

- Highly available Red Hat OpenStack Platform deployment
- Functional testing of OpenStack project, quotas, and instances
- Functional testing of OpenStack tenant networks by using VXLAN ml2 plug-in and OpenStack routers
- OpenStack external networks, provider networks, and floating IPs
- Booting an instance from Cinder volume and image caching
- Cinder features such as volume operations, snapshots, clones, QoS types
- Cinder dynamic volume retyping
- OpenStack Swift storage for Glance
- Highly available and automated Red Hat OpenShift Container Platform deployment
- Octavia Load Balancer as a Service
- OpenShift logging and monitoring
- Persistent storage for OpenShift applications by using both Cinder and Trident
- OpenShift registry deployment

6.1 Security Recommendations

NetApp strongly recommends that you keep SELinux and the firewall enabled on all the nodes in your deployment.

7 Conclusion

NetApp HCI for Private Cloud with Red Hat solution enables your organization to build a production-grade OpenStack cloud in a reliable and risk-free manner. This solution also includes Red Hat OpenShift Container Platform to run microservices workloads on-premises to accelerate application development and deployment.

Acknowledgements

The author of this document would like to thank and acknowledge the following people; they all had a positive impact on the quality of this document:

- James Bradshaw, Technical Marketing Engineer, NetApp
- Thomas Hanvey, Technical Marketing Engineer, NetApp
- Aaron Patten, Principal Architect, NetApp
- Bala Rameshbabu, Technical Marketing Engineer, NetApp
- Martin Andre, Senior Software Engineer, Red Hat
- Tomas Sedovic, Senior Software Engineer, Red Hat

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
  https://www.netapp.com/us/products/converged-systems/hyper-converged-infrastructure.aspx
- NetApp HCI datasheet
  https://www.netapp.com/us/media/ds-3881.pdf
- NetApp HCI with Mellanox Switches
  https://www.netapp.com/us/media/tr-4735.pdf
- Red Hat OpenStack Platform 13
  https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/
- Red Hat OpenShift Container Platform 3.11
  https://access.redhat.com/documentation/en-us/openshift_container_platform/3.11/
- Deploying OpenShift on OpenStack
  https://docs.openshift.com/container-platform/3.11/install_config/configuring_openstack.html
- OpenStack Kolla
  https://github.com/openstack/kolla
- NIST Definition of Cloud Computing
  https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
- OpenStack: A Business Perspective
  https://www.openstack.org/assets/pdf-downloads/business-perspectives.pdf
- NetApp Trident
  https://netapp-trident.readthedocs.io/en/stable-v19.04/#
- NetApp Product Documentation
  https://docs.netapp.com

Version History

Version	Date	Document Version History
Version 1.0	May 2019	Initial release.

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

Copyright Information

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.

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

Trademark Information

NVA-1133-DESIGN-0519