Technical Report

Red Hat OpenStack Platform 8 on FlexPod Reference Architecture and Storage Deployment

Dave Cain, NetApp
April 2016 | TR-4506

Abstract

FlexPod® forms a flexible, open, integrated foundation for your enterprise-grade OpenStack cloud environment. FlexPod combines best-in-class components (Cisco Unified Computing System [Cisco UCS] servers, Cisco Nexus switches, and NetApp® FAS and E-Series storage) into a unified platform for physical, virtual, and cloud applications that speeds deployment and provisioning, reduces risk, and lowers IT costs for application workloads.
TABLE OF CONTENTS

1 Solution Overview ...................................................................................................................................... 6
   1.1 Solution Technology ........................................................................................................................... 7
   1.2 Use Case Summary ............................................................................................................................. 20

2 Solution Technology .................................................................................................................................. 20
   2.1 Solution Hardware ............................................................................................................................... 20
   2.2 Solution Software ................................................................................................................................ 21
   2.3 OpenStack Storage Elements ........................................................................................................... 22
   2.4 NetApp Storage for OpenStack .......................................................................................................... 23
   2.5 Red Hat OpenStack Platform Director ............................................................................................... 29
   2.6 Heat Orchestration Templates (HOT) and Overcloud Customization ............................................ 32

3 Solution Configuration ................................................................................................................................ 40
   3.1 Physical Topology .................................................................................................................................. 40
   3.2 Cabling .................................................................................................................................................. 41
   3.3 Laboratory and Network Diagrams ...................................................................................................... 48
   3.4 Cisco Nexus 9396PX Setup ............................................................................................................... 53
   3.5 NetApp FAS8040 Setup ....................................................................................................................... 59
   3.6 NetApp E-Series E5660 Setup ............................................................................................................ 79
   3.7 Cisco UCS Setup .................................................................................................................................... 91

4 Solution Deployment ................................................................................................................................... 93
   4.1 Deploy Red Hat OpenStack Platform Director .................................................................................. 93
   4.2 Customize Red Hat OpenStack Platform Director ........................................................................... 102
   4.3 Launch Introspection of Servers ....................................................................................................... 109
   4.4 Deploy Overcloud ................................................................................................................................ 114

5 Solution Operations: Functionality Verification ...................................................................................... 118
   5.1 Log in to Horizon ................................................................................................................................ 118
   5.2 Upload Image to Image Store ............................................................................................................. 119
   5.3 Create Flavors ...................................................................................................................................... 121
   5.4 Create Project and User ...................................................................................................................... 121
   5.5 Create Tenant Network and Router ................................................................................................... 122
   5.6 Set Gateway and Create Floating IP Network and Subnet ............................................................... 124
   5.7 Create and Boot Instances from Volume ............................................................................................. 126
   5.8 Associate Floating IP Address with Instance .................................................................................... 129
   5.9 Verify Inbound and Outbound Network Traffic to Instance ............................................................ 130
5.10 Provision and Access Share Using File Share Service (Manila) ..................................................131
5.11 Upload to and Access Objects in Object Store (Swift) .................................................................135

6  Solution Operations: Load Testing ........................................................................................................136
  6.1 OpenStack Rally ................................................................................................................................136
  6.2 Scenario Definitions and Tasks...........................................................................................................138

7  Solution Maintenance ................................................................................................................................144
  7.1 Update Undercloud ..............................................................................................................................144
  7.2 Update Overcloud ...............................................................................................................................144

8  Conclusion ................................................................................................................................................145

Acknowledgments .......................................................................................................................................146

References ..................................................................................................................................................146
  FlexPod ..................................................................................................................................................146
  NetApp FAS Storage ..............................................................................................................................146
  NetApp E-Series Storage ........................................................................................................................147
  Cisco UCS ...............................................................................................................................................147
  Cisco Nexus Networking ........................................................................................................................147
  Red Hat Enterprise Linux 7 .....................................................................................................................147
  Red Hat OpenStack Platform 8 ..............................................................................................................148
  OpenStack at NetApp ............................................................................................................................148
  OpenStack Upstream .............................................................................................................................148
  Hardware and Software Certification ....................................................................................................148

Version History ..........................................................................................................................................148

LIST OF TABLES
Table 1) Solution hardware. ......................................................................................................................20
Table 2) Software requirements. ................................................................................................................21
Table 3) Types of persistent storage in OpenStack. ..................................................................................22
Table 4) OpenStack storage summary. .................................................................................................23
Table 5) Undercloud components. ...........................................................................................................31
Table 6) network-environment.yaml variable definitions. ...................................................................33
Table 7) flexpod.yaml variable definitions. ............................................................................................35
Table 8) NetApp FAS8040 cabling information. ......................................................................................44
Table 9) NetApp E-5660 cabling information. ........................................................................................44
Table 10) Cisco UCS 6248UP Fabric Interconnect cabling .................................................................45
Table 11) Cisco Nexus 9396PX infraswitch cabling. ..............................................................................46
Table 12) Cisco UCS 5108 chassis cabling.................................................................48
Table 13) Necessary VLANs ..........................................................................................50
Table 14) VLAN purpose ...............................................................................................50
Table 15) Cluster details for the clustered Data ONTAP software configuration ................60
Table 16) Cluster details for the cluster-join operation .......................................................65
Table 17) iSCSI target name for Cisco UCS booting ..........................................................79
Table 18) SANtricity OS software configuration worksheet .............................................79
Table 19) Undercloud.conf variable definitions ...............................................................101
Table 20) Introspection JSON attributes .......................................................................109
Table 21) NFS shares used by Rally ..............................................................................137
Table 22) Configuration changes required on controller systems ......................................137
Table 23) Overcloud configuration addition required ........................................................137
Table 24) Scenario 1 volume results .............................................................................139
Table 25) Scenario 2 results ..........................................................................................143

LIST OF FIGURES
Figure 1) FlexPod Datacenter component family. .........................................................7
Figure 2) NetApp FAS8040 front and rear view. ..............................................................8
Figure 3) NetApp clustered Data ONTAP .....................................................................10
Figure 4) NetApp E-Series E5660 ..............................................................................15
Figure 5) Disk failure in a DDP ..................................................................................16
Figure 6) Cisco fabric interconnect front and rear views.................................................18
Figure 7) Cisco UCS 5108 Blade chassis ....................................................................18
Figure 8) Cisco UCS 2208XP Fabric Extender .............................................................18
Figure 9) Cisco Nexus 9000 Series switch .................................................................19
Figure 10) Instance caching ..........................................................................................28
Figure 11) Overcloud and undercloud relationship .......................................................30
Figure 12) High-level solution diagram .....................................................................41
Figure 13) FAS and E-Series cabling to Cisco Nexus switch pair ....................................43
Figure 14) Fabric interconnect and Cisco Nexus cabling ..............................................45
Figure 15) Cisco UCS chassis cabling to fabric interconnect pair ....................................47
Figure 16) Connectivity diagram ...............................................................................49
Figure 17) Logical network segmentation diagram for compute and storage ..................52
Figure 18) Undercloud vNIC and network segmentation .............................................92
Figure 19) Overcloud controller vNIC and network segmentation ..................................92
Figure 20) Overcloud compute vNIC and network segmentation ..................................93
Figure 21) Comparison of total time taken to complete Scenario 1.................................140
Figure 22) Comparison of total disk space consumed in Scenario 1...............................141
Figure 23) Comparison of time taken to create a single bootable volume in Scenario 2 ....143

©2016 NetApp, Inc. All rights reserved.
1 Solution Overview

FlexPod is a predesigned, best practice data center architecture built on the Cisco Unified Computing System (Cisco UCS), the Cisco Nexus family of switches, and NetApp fabric-attached storage (FAS) and/or E-Series systems.

FlexPod is a suitable platform for running various virtualization hypervisors and bare metal operating systems and enterprise workloads. FlexPod delivers not only a baseline configuration but also the flexibility to be sized and optimized to accommodate many different use cases and requirements.

The FlexPod architecture is highly modular, delivers a baseline configuration, and has the flexibility to be sized and optimized to accommodate many different use cases and requirements. The FlexPod architecture can both scale up (adding additional resources within a FlexPod unit) and scale out (adding additional FlexPod units). FlexPod paired with Red Hat OpenStack Platform 8 is an extension of the already wide range of FlexPod validated and supported design portfolio entries and includes best-in-class technologies from NetApp, Cisco, and Red Hat.

FlexPod Datacenter:

- Is suitable for large enterprises and cloud service providers that have mature IT processes and rapid growth expectations and want to deploy a highly scalable, shared infrastructure for multiple business-critical applications
- Simplifies and modernizes IT with continuous innovation and broad support for any cloud strategy
- Provides easy infrastructure scaling with a clearly defined upgrade path that leverages all existing components and management processes
- Reduces cost and complexity with maximum uptime and minimal risk

The FlexPod Datacenter family of components is illustrated in Figure 1.
1.1 Solution Technology

FlexPod Datacenter with Red Hat OpenStack Platform 8 uses technologies from NetApp, Cisco, and Red Hat that are configured according to each company’s best practices. This section discusses the products and technologies used in the solution.

NetApp FAS8000

The FlexPod Datacenter solution includes the NetApp FAS8000 series, unified scale-out storage system for OpenStack Cinder block storage service, OpenStack Glance image service, and OpenStack Manila file share service. Powered by NetApp clustered Data ONTAP®, the FAS8000 series unifies the SAN and
NAS storage infrastructure. Systems architects can choose from a range of models representing a spectrum of cost-versus-performance points. Every model, however, provides the following core benefits:

- **HA and fault tolerance.** Storage access and security are achieved through clustering, high availability (HA) pairing of controllers, hot-swappable components, NetApp RAID DP® disk protection (allowing two independent disk failures without data loss), network interface redundancy, support for data mirroring with NetApp SnapMirror® software, application backup integration with the NetApp SnapManager® storage management software, and customizable data protection with the NetApp Snap Creator® framework and NetApp SnapProtect® products.

- **Storage efficiency.** Users can store more data with less physical media. This efficiency is achieved with thin provisioning (unused space is shared among volumes), NetApp Snapshot® copies (zero-storage, read-only clones of data over time), NetApp FlexClone® volumes and logical unit numbers (LUNs) (read/write copies of data in which only changes are stored), deduplication (dynamic detection and removal of redundant data), and data compression.

- **Unified storage architecture.** Every model runs the same software (clustered Data ONTAP); supports all storage protocols (CIFS, NFS, iSCSI, FCP, and FCoE); and uses SATA, SAS, or solid-state drive (SSD) storage (or a mix) on the back end. This allows freedom of choice in upgrades and expansions, without the need for rearchitecting the solution or retraining operations personnel.

- **Advanced clustering.** Storage controllers are grouped into clusters for both availability and performance pooling. Workloads are movable between controllers, permitting dynamic load balancing and zero-downtime maintenance and upgrades. Physical media and storage controllers are added as needed to support growing demand without downtime.

### NetApp Storage Controllers

NetApp storage controllers receive and send data from the host. Controller nodes are deployed in HA pairs, with these HA pairs participating in a single storage domain or cluster. This unit detects and gathers information about its own hardware configuration, the storage system components, the operational status, hardware failures, and other error conditions. A storage controller is redundantly connected to storage through disk shelves, which are the containers or device carriers that hold disks and associated hardware such as power supplies, connectivity interfaces, and cabling.

The NetApp FAS8000 features a multicore Intel chipset and leverages high-performance memory modules, NVRAM to accelerate and optimize writes, and an I/O-tuned Peripheral Component Interconnect Express (PCIe) Gen 3 architecture that maximizes application throughput. The FAS8000 series come with integrated unified target adapter (UTA2) ports that support 16GB Fibre Channel (FC), 10GbE, or FCoE. Figure 2 shows a front and rear view of the FAS8040/8060 controllers.

*Figure 2) NetApp FAS8040 front and rear view.*
If storage requirements change over time, NetApp storage offers the flexibility to change quickly as needed without expensive and disruptive forklift upgrades. This applies to different types of changes:

- Physical changes, such as expanding a controller to accept more disk shelves and then more hard-disk drives (HDDs) without an outage
- Logical or configuration changes, such as expanding a RAID group to incorporate these new drives without requiring an outage
- Access protocol changes, such as modification of a virtual representation of a hard drive to a host by changing a LUN from FC access to iSCSI access, with no data movement required, but only a simple dismount of the FC LUN and a mount of the same LUN using iSCSI

In addition, a single copy of data can be shared between Linux and Windows systems while allowing each environment to access the data through native protocols and applications. In a system that was originally purchased with all SATA disks for backup applications, high-performance SSDs can be added to the same storage system to support tier 1 applications, such as Oracle, Microsoft Exchange, or Microsoft SQL Server.

For more information about NetApp FAS8000 series, see NetApp FAS8000 Series Unified Scale-Out Storage for the Enterprise.

NetApp Clustered Data ONTAP 8.3.2 Fundamentals

NetApp provides enterprise-ready, unified scale-out storage with clustered Data ONTAP 8.3.2, the operating system physically running on the storage controllers in the NetApp FAS appliance. Developed from a solid foundation of proven Data ONTAP technology and innovation, clustered Data ONTAP is the basis for large virtualized shared-storage infrastructures that are architected for nondisruptive operations over the system lifetime.

Note: Data ONTAP operating in 7-Mode is not available as a mode of operation in version 8.3.2.

Data ONTAP scale-out is one way to respond to growth in a storage environment. All storage controllers have physical limits to their expandability; the number of CPUs, number of memory slots, and amount of space for disk shelves dictate maximum capacity and controller performance. If more storage or performance capacity is needed, it might be possible to add CPUs and memory or install additional disk shelves, but ultimately the controller becomes populated, with no further expansion possible. At this stage, the only option is to acquire another controller. One way to acquire another controller is to scale up: that is, to add additional controllers in such a way that each is an independent management entity that does not provide any shared storage resources. If the original controller is replaced by a newer, larger controller, data migration is required to transfer the data from the old controller to the new one. This process is time consuming and potentially disruptive and most likely requires configuration changes on all of the attached host systems.

If the newer controller can coexist with the original controller, then the two storage controllers must be individually managed, and there are no native tools to balance or reassign workloads across them. The situation becomes worse as the number of controllers increases. If the scale-up approach is used, the operational burden increases consistently as the environment grows, and the result is an unbalanced and difficult-to-manage environment. Technology refresh cycles require substantial planning, lengthy outages, and configuration changes, which introduce risk into the system.

In contrast, a scale-out approach seamlessly adds additional controllers to the resource pool residing on a shared storage infrastructure, even as the storage environment grows. Host and client connections, as well as volumes, can move seamlessly and nondisruptively anywhere in the resource pool, so that existing workloads can be easily balanced over the available resources, and new workloads can be easily deployed. Refresh technology—replace disk shelves, add or completely replace storage controllers—while the environment remains online and continues to serve data.

Although scale-out products have been available for some time, these products were typically subject to one or more of the following shortcomings:
• Limited protocol support: block only
• Limited hardware support: supported only a particular type of storage controller or a limited set
• Little or no storage efficiency: thin provisioning, deduplication, compression
• Little or no data replication capability

Therefore, although these products are positioned for certain specialized workloads, they are less flexible, less capable, and not robust enough for broad deployment throughout the enterprise.

Data ONTAP is the first product to offer a complete scale-out solution, and it offers an adaptable, always-available storage infrastructure for a modern, highly virtualized environment.

Scale-Out

Data centers require agility. In a data center, each storage controller has CPU, memory, and disk shelf limits. Scale-out means that as the storage environment grows, additional controllers can be added seamlessly to the resource pool residing on a shared storage infrastructure. Seamlessly and nondisruptively move host and client connections, as well as volumes, anywhere within the resource pool.

The benefits of scale-out include:
• Nondisruptive operations
• The ability to add tenants, instances, volumes, networks, and so on, without downtime in OpenStack
• Operational simplicity and flexibility

As Figure 3 shows, clustered Data ONTAP offers a way to solve the scalability requirements in a storage environment. A clustered Data ONTAP system can scale up to 24 nodes, depending on platform and protocol, and can contain different disk types and controller models in the same storage cluster with up to 101PB of capacity.

Figure 3) NetApp clustered Data ONTAP.

Nondisruptive Operations

The move to shared infrastructure has made it nearly impossible to schedule downtime for routine maintenance. NetApp clustered Data ONTAP is designed to eliminate the need for planned downtime for maintenance operations and lifecycle operations as well as the unplanned downtime caused by hardware and software failures. NetApp storage solutions provide redundancy and fault tolerance through clustered storage controllers and redundant, hot-swappable components, such as cooling fans, power supplies,
disk drives, and shelves. This highly available and flexible architecture enables customers to manage all data under one common infrastructure and meet mission-critical uptime requirements.

Three standard tools that eliminate the possible downtime:

- **NetApp DataMotion™ data migration software for volumes (vol move).** Allows you to move data volumes from one aggregate to another on the same or a different cluster node.

- **Logical interface (LIF) migration.** Allows you to virtualize the physical Ethernet interfaces in clustered Data ONTAP. LIF migration allows the administrator to move these virtualized LIFs from one network port to another on the same or a different cluster node.

- **Aggregate relocate (ARL).** Allows you to transfer complete aggregates from one controller in an HA pair to the other without data movement.

Used individually and in combination, these tools allow you to nondisruptively perform a full range of operations, from moving a volume from a faster to a slower disk all the way up to a complete controller and storage technology refresh.

As storage nodes are added to the system, all physical resources (CPUs, cache memory, network input/output [I/O] bandwidth, and disk I/O bandwidth) can easily be kept in balance. Data ONTAP enables users to:

- Move data between storage controllers and tiers of storage without disrupting users and applications.

- Dynamically assign, promote, and retire storage, while providing continuous access to data as administrators upgrade or replace storage.

- Increase capacity while balancing workloads and reduce or eliminate storage I/O hot spots without the need to remount shares, modify client settings, or stop running applications.

These features allow a truly nondisruptive architecture in which any component of the storage system can be upgraded, resized, or rearchitected without disruption to the private cloud infrastructure.

**Availability**

Shared storage infrastructure provides services to many different tenants in an OpenStack deployment. In such environments, downtime produces disastrous effects. NetApp FAS eliminates sources of downtime and protects critical data against disaster through two key features:

- **HA.** A NetApp HA pair provides seamless failover to its partner in the event of any hardware failure. Each of the two identical storage controllers in the HA pair configuration serves data independently during normal operation. During an individual storage controller failure, the data service process is transferred from the failed storage controller to the surviving partner.

- **RAID DP.** During any OpenStack deployment, data protection is critical because any RAID failure might disconnect and/or shut off hundreds or potentially thousands of end users from their virtual machines (VMs), resulting in lost productivity. RAID DP provides performance comparable to that of RAID 10 but requires fewer disks to achieve equivalent protection. RAID DP provides protection against double-disk failure, in contrast to RAID 5, which only protects against one disk failure per RAID group, in effect providing RAID 10 performance and protection at a RAID 5 price point.

For more information, see [Clustered Data ONTAP 8.3 High-Availability Configuration Guide](#).

**NetApp Advanced Data Management Capabilities**

This section describes the storage efficiencies, advanced storage features, and multiprotocol support capabilities of the NetApp FAS8000 system.

**Storage Efficiencies**

NetApp FAS includes built-in thin provisioning, data deduplication, compression, and zero-cost cloning with NetApp FlexClone technology, achieving multilevel storage efficiency across OpenStack instances,
installed applications, and user data. This comprehensive storage efficiency enables a significant reduction in storage footprint, with a capacity reduction of up to 10:1, or 90% (based on existing customer deployments and NetApp Solutions Lab validation). Four features make this storage efficiency possible:

- **Thin provisioning.** Allows multiple applications to share a single pool of on-demand storage, eliminating the need to provision more storage for one application if another application still has plenty of allocated but unused storage.

- **Deduplication.** Saves space on primary storage by removing redundant copies of blocks in a volume that hosts hundreds of instances. This process is transparent to the application and the user, and it can be enabled and disabled dynamically or scheduled to run at off-peak hours.

- **Compression.** Compresses data blocks. Compression can be run whether or not deduplication is enabled and can provide additional space savings whether it is run alone or together with deduplication.

- **FlexClone technology.** Offers hardware-assisted rapid creation of space-efficient, writable, point-in-time images of individual VM files, LUNs, or flexible volumes. The use of FlexClone technology in OpenStack deployments provides high levels of scalability and significant cost, space, and time savings. The NetApp Cinder driver provides the flexibility to rapidly provision and redeploy thousands of instances with little space used on the storage system.

**Advanced Storage Features**

Data ONTAP advanced storage features include:

- **NetApp Snapshot copy backups.** A manual or automatically scheduled point-in-time copy that writes only changed blocks, with no performance penalty. A Snapshot copy consumes minimal storage space because only changes to the active file system are written. Individual files and directories can easily be recovered from any Snapshot copy, and the entire volume can be restored back to any Snapshot state in seconds. A NetApp Snapshot copy incurs no performance overhead. Users can comfortably store up to 255 NetApp Snapshot copies per NetApp FlexVol® volume, all of which are accessible as read-only and online versions of the data.

  **Note:** Snapshot copies are created at the FlexVol volume level, so they cannot be directly leveraged within an OpenStack user context. This is because a Cinder user requests that a Snapshot copy of a particular Cinder volume be created, not the containing FlexVol volume. Because a Cinder volume is represented either as a file in the NFS or as a LUN (in the case of iSCSI or FC), Cinder snapshots can be created by using FlexClone, which allows you to create many thousands of Cinder snapshots of a single Cinder volume. NetApp Snapshot copies are, however, available to OpenStack administrators to do administrative backups, create and/or modify data protection policies, and so on.

- **LIFs.** A LIF is a logical interface that is associated with a physical port, interface group, or virtual LAN (VLAN) interface. There are three types of LIFs: NFS LIFs, iSCSI LIFs, and FC LIFs. More than one LIF might be associated with a physical port at the same time. LIFs are logical network entities that have the same characteristics as physical network devices but are not tied to physical objects. LIFs used for Ethernet traffic are assigned specific Ethernet-based details such as IP addresses and iSCSI qualified names and are then associated with a specific physical port capable of supporting Ethernet. LIFs used for FC-based traffic are assigned specific FC-based details, such as worldwide port names (WWPNs), and are associated with a specific physical port capable of supporting FC or FCoE. NAS LIFs can nondisruptively migrated to any other physical network port throughout the entire cluster at any time, either manually or automatically (by using policies), whereas SAN LIFs rely on Microsoft Multipath I/O (MPIO) and asymmetric logical unit access (ALUA) to notify clients of any change in the network topology.

- **Storage virtual machines (SVMs).** An SVM is a secure virtual storage server that contains data volumes and one or more LIFs, through which it serves data to the clients. An SVM securely isolates the shared, virtualized data storage and network and appears as a single dedicated server to its clients. Each SVM has a separate administrator authentication domain and can be managed independently by an SVM administrator.
Unified Storage Architecture and Multiprotocol Support

NetApp also offers a NetApp unified storage architecture with a family of storage systems that simultaneously support SAN (through FCoE, FC, and iSCSI) and NAS (through CIFS and NFS) across many operating environments, including OpenStack, VMware, Windows, Linux, and UNIX. This single architecture provides access to data by using industry-standard protocols, including NFS, CIFS, iSCSI, FCP, SCSI, and NDMP.

Connectivity options include standard Ethernet (10/100/1000Mb or 10GbE) and FC (4, 8, or 16Gbps). In addition, all systems can be configured with high-performance SSDs or SAS (SAS) disks for primary storage applications, low-cost SATA disks for secondary applications (backup, archiving, and so on), or a mix of the different disk types. By supporting all common NAS and SAN protocols on a single platform, NetApp FAS enables:

- Direct access to storage for each client
- Network file sharing across different platforms without the need for protocol-emulation products such as SAMBA, NFS Maestro, or PC-NFS
- Simple and fast data storage and data access for all client systems
- Fewer storage systems
- Greater efficiency with each system deployed

Clustered Data ONTAP supports several protocols concurrently in the same storage system, and data replication and storage efficiency features are supported across all protocols. The following protocols are supported:

- NFS v3, v4, and v4.1, including pNFS
- iSCSI, FCI
- FCoE
- SMB 1, 2, 2.1, and 3

SVMs

An SVM is the secure logical storage partition through which data is accessed in clustered Data ONTAP. A cluster serves data through at least one and possibly multiple SVMs. An SVM is a logical abstraction that represents a set of physical resources of the cluster. Data volumes and logical network LIFs are created and assigned to an SVM and can reside on any node in the cluster to which the SVM has been given access. An SVM can own resources on multiple nodes concurrently, and those resources can be moved nondisruptively from one node to another. For example, a flexible volume can be nondisruptively moved to a new node, and an aggregate or a data LIF can be transparently reassigned to a different physical network port. The SVM abstracts the cluster hardware and is not tied to specific physical hardware.

An SVM is capable of supporting multiple data protocols concurrently. Volumes within the SVM can be combined together to form a single NAS namespace, which makes all of an SVM's data available to NFS and CIFS clients through a single share or mount point. For example, a 24-node cluster licensed for UNIX and Windows File Services that has a single SVM configured with thousands of volumes can be accessed from a single network interface on one of the nodes. SVMs also support block-based protocols, and LUNs can be created and exported by using iSCSI, FC, or FCoE. Any or all of these data protocols can be configured for use within a given SVM.

An SVM is a secure entity. Therefore, it is aware of only the resources that have been assigned to it and has no knowledge of other SVMs and their respective resources. Each SVM operates as a separate and distinct entity with its own security domain. Tenants can manage the resources allocated to them through a delegated SVM administration account. An SVM is effectively isolated from other SVMs that share physical hardware and therefore is uniquely positioned to align with OpenStack tenants for a truly
comprehensive, multitenant environment. Each SVM can connect to unique authentication zones, such as AD, LDAP, or NIS.

From a performance perspective, maximum IOPS and throughput levels can be set per SVM by using QoS policy groups, which allow the cluster administrator to quantify the performance capabilities allocated to each SVM.

Clustered Data ONTAP is highly scalable, and additional storage controllers and disks can easily be added to existing clusters to scale capacity and performance to meet rising demands. Because these are virtual storage servers within the cluster, SVMs are also highly scalable. As new nodes or aggregates are added to the cluster, the SVM can be nondisruptively configured to use them. New disk, cache, and network resources can be made available to the SVM to create new data volumes or to migrate existing workloads to these new resources to balance performance.

This scalability also enables the SVM to be highly resilient. SVMs are no longer tied to the lifecycle of a given storage controller. As new replacement hardware is introduced, SVM resources can be moved nondisruptively from the old controllers to the new controllers, and the old controllers can be retired from service while the SVM is still online and available to serve data.

SVMs have three main components:

- **LIFs.** All SVM networking is done through LIFs created within the SVM. As logical constructs, LIFs are abstracted from the physical networking ports on which they reside.

- **Flexible volumes.** A flexible volume is the basic unit of storage for an SVM. An SVM has a root volume and can have one or more data volumes. Data volumes can be created in any aggregate that has been delegated by the cluster administrator for use by the SVM. Depending on the data protocols used by the SVM, volumes can contain either LUNs for use with block protocols, files for use with NAS protocols, or both concurrently. For access using NAS protocols, the volume must be added to the SVM namespace through the creation of a client-visible directory called a junction.

- **Namespaces.** Each SVM has a distinct namespace through which all of the NAS data shared from that SVM can be accessed. This namespace can be thought of as a map to all of the junctioned volumes for the SVM, regardless of the node or the aggregate on which they physically reside. Volumes can be junctioned at the root of the namespace or beneath other volumes that are part of the namespace hierarchy.
  - For more information about namespaces, see TR-4129: Namespaces in Clustered Data ONTAP.
  - For more information about Data ONTAP, see the NetApp Data ONTAP Operating System product page.

**NetApp E5000 Series**

This FlexPod Datacenter solution also makes use of the NetApp E-Series E5660 storage system, primarily for the OpenStack Object Storage (Swift) service. An E5660 is composed of dual E5600 controllers mated with the 4U 60-drive DE6600 chassis. The NetApp E5600 storage system family is designed to meet the demands of the most data-intensive applications and provide continuous access to data. It is from the E-Series line, which offers zero scheduled downtime systems, redundant hot-swappable components, automated path failover, and online administration capabilities.

The E5660 is shown in Figure 4.
NetApp E-Series Storage Controllers

The E5000 series controllers deliver enterprise-level availability with:

- Dual active controllers, fully redundant I/O paths, and automated failover
- Battery-backed cache memory that is destaged to flash upon power loss
- Extensive monitoring of diagnostic data that provides comprehensive fault isolation, simplifying analysis of unanticipated events for timely problem resolution
- Proactive repair that helps get the system back to optimal performance in minimum time

This storage system additionally provides the following high-level benefits:

- **Flexible interface options.** The E-Series platform supports a complete set of host or network interfaces designed for either direct server attachment or network environments. With multiple ports per interface, the rich connectivity provides ample options and bandwidth for high throughput. The interfaces include quad-lane SAS, iSCSI, FC, and InfiniBand to connect with and protect investments in storage networking.

- **HA and reliability.** E-Series simplifies management and maintains organizational productivity by keeping data accessible through redundant protection, automated path failover, and online administration, including online NetApp SANtricity® OS and drive firmware updates. Advanced protection features and extensive diagnostic capabilities deliver high levels of data integrity, including T10-PI data assurance to protect against silent drive errors.

- **Maximum storage density and modular flexibility.** E-Series offers multiple form factors and drive technology options to meet your storage requirements. The ultradense 60-drive system shelf supports up to 360TB in just 4U of space. It is perfect for environments with large amounts of data and limited floor space. Its high-efficiency power supplies and intelligent design can lower power use up to 40% and cooling requirements by up to 39%.

- **Intuitive management.** SANtricity Storage Manager software offers extensive configuration flexibility, optimal performance tuning, and complete control over data placement. With its dynamic capabilities, SANtricity software supports on-the-fly expansion, reconfigurations, and maintenance without interrupting storage system I/O.

For more information about the NetApp E5660, see the NetApp E5600 Hybrid Storage System product page.

**NetApp SANtricity Operating System Fundamentals**

With over 20 years of storage development behind it, and approaching nearly one million systems shipped, the E-Series platform is based on a field-proven architecture that uses the SANtricity storage management software on the controllers. SANtricity OS is designed to provide high reliability and greater than 99.999% availability, data integrity, and security.

SANtricity OS also:
• Delivers best-in-class reliability with automated features, online configuration options, state-of-the-art RAID, proactive monitoring, and NetApp AutoSupport® capabilities.

• Extends data protection through FC- and IP-based remote mirroring, SANtricity Dynamic Disk Pools (DDP), enhanced Snapshot copies, data-at-rest encryption, data assurance to make sure of data integrity, and advanced diagnostics.

• Includes plug-ins for application-aware deployments of Oracle, VMware, Microsoft, and Splunk applications.

For more information, see the NetApp SANtricity Operating System product page.

DDP

DDP increases the level of data protection, provides more consistent transactional performance, and improves the versatility of E-Series systems. DDP dynamically distributes data, spare capacity, and parity information across a pool of drives. An intelligent algorithm (with seven patents pending) determines which drives are used for data placement, and data is dynamically recreated and redistributed as needed to maintain protection and uniform distribution.

Consistent Performance During Rebuilds

DDP minimizes the performance drop that can occur during a disk rebuild, allowing rebuilds to complete up to eight times more quickly than with traditional RAID. Therefore, your storage spends more time in an optimal performance mode that maximizes application productivity. Shorter rebuild times also reduce the possibility of a second disk failure occurring during a disk rebuild and protect against unrecoverable media errors. Stripes with several drive failures receive priority for reconstruction.

Overall, DDP provides a significant improvement in data protection: the larger the pool, the greater the protection. A minimum of 11 disks is required to create a disk pool.

How DDP Works

When a disk fails with traditional RAID, data is recreated from parity on a single hot spare drive, creating a bottleneck. All volumes using the RAID group suffer. DDP distributes data, parity information, and spare capacity across a pool of drives. Its intelligent algorithm, based on the Controlled Replication Under Scalable Hashing (CRUSH) algorithm, defines which drives are used for segment placement, making sure of full data protection. DDP dynamic rebuild technology uses every drive in the pool to rebuild a failed drive, enabling exceptional performance under failure. Flexible disk-pool sizing optimizes utilization of any configuration for maximum performance, protection, and efficiency.

When a disk fails in a DDP, reconstruction activity is spread across the pool, and the rebuild is completed eight times more quickly, as shown in Figure 5.

Figure 5) Disk failure in a DDP.
Cisco Unified Computing System

The Cisco Unified Computing System (Cisco UCS) is a next-generation solution for blade and rack server computing. The system integrates a low-latency, lossless, 10GbE unified network fabric with enterprise-class, x86-architecture servers. The system is an integrated, scalable, multichassis platform in which all resources participate in a unified management domain. Cisco UCS accelerates the delivery of new services simply, reliably, and securely through end-to-end provisioning and migration support for both virtualized and nonvirtualized systems. Cisco UCS consists of the following components:

- **Compute.** The Cisco UCS B-Series Blade Servers are designed to increase performance, energy efficiency, and flexibility for demanding virtualized and nonvirtualized applications. Cisco UCS B-Series Blade Servers adapt processor performance to application demands and intelligently scale energy use based on utilization. Each Cisco UCS B-Series Blade Server uses converged network adapters (CNAs) for access to the unified fabric. This design reduces the number of adapters, cables, and access-layer switches while still allowing traditional LAN and SAN connectivity. This Cisco innovation reduces capital expenditures and operating expenses, including administrative overhead and power and cooling costs.

- **Network.** The system integrates into a low-latency, lossless, 10Gbps unified network fabric. This network foundation consolidates LANs, SANs, and high-performance computing networks, which are separate networks today. The unified fabric lowers costs by reducing the number of network adapters, switches, and cables and by decreasing the power and cooling requirements.

- **Virtualization.** The system unleashes the full potential of virtualization by enhancing the scalability, performance, and operational control of virtual environments. Cisco security, policy enforcement, and diagnostic features extend into virtualized environments to better support changing business and IT requirements.

- **Storage access.** The system provides consolidated access to both SAN storage and NAS over the unified fabric. By unifying the storage access, Cisco UCS can access storage over Ethernet (SMB 3.0 or iSCSI), FC, and FCoE. This provides customers with storage choices and investment protection. In addition, server administrators can preassign storage-access policies to storage resources, for simplified storage connectivity and management, leading to increased productivity.

- **Management.** The system uniquely integrates all system components to enable the entire solution to be managed as a single entity by Cisco UCS Manager. Cisco UCS Manager has an intuitive graphical user interface (GUI), a command-line interface (CLI), and a powerful scripting library module for Microsoft PowerShell built on a robust application programming interface (API) to manage all system configuration and operations.

Cisco UCS fuses access-layer networking and servers. This high-performance, next-generation server system provides a data center with a high degree of workload agility and scalability.

**Cisco UCS 6248UP Fabric Interconnects**

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

- Cisco UCS 6200 Series Fabric Interconnects support the system's 80Gbps unified fabric with low-latency, lossless, cut-through switching that supports IP, storage, and management traffic using a single set of cables. The fabric interconnects feature virtual interfaces that terminate both physical and virtual connections equivalently, establishing a virtualization-aware environment in which blade, rack servers, and VMs are interconnected using the same mechanisms. The Cisco UCS 6248UP is a 1RU fabric interconnect that features up to 48 universal ports that can support 80GbE, FCoE, or native FC connectivity. The Cisco UCS interconnect front and rear views are shown in Figure 6.

For more information, see [Cisco UCS 6200 Series Fabric Interconnects](#).
Cisco UCS 5108 Blade Server Chassis

The Cisco UCS 5100 Series Blade Server chassis is a crucial building block of Cisco UCS, delivering a scalable and flexible blade server chassis. The Cisco UCS 5108 Blade Server chassis is six rack units (6RU) high and can mount in an industry-standard 19-inch rack. A single chassis can house up to eight half-width Cisco UCS B-Series Blade Servers and can accommodate both half-width and full-width blade form factors. Four single-phase, hot-swappable power supplies are accessible from the front of the chassis. These power supplies are 92% efficient and can be configured to support nonredundant, N+1 redundant, and grid-redundant configurations. The rear of the chassis contains eight hot-swappable fans, four power connectors (one per power supply), and two I/O bays for Cisco UCS 2204XP or 2208XP fabric extenders. A passive midplane provides up to 40Gbps of I/O bandwidth per server slot and up to 80Gbps of I/O bandwidth for two slots. The chassis is capable of supporting future 80GbE standards. The Cisco UCS 5108 Blade chassis is shown in Figure 7.

For more information, see Cisco UCS 5100 Series Blade Server Chassis.

Cisco UCS Fabric Extenders

The Cisco UCS 2208XP Fabric Extender (Figure 8) has 8 10GbE, FCoE-capable, Enhanced Small Form-Factor Pluggable (SFP+) ports that connect the blade chassis to the fabric interconnect. Each Cisco UCS 2208XP has 32 10GbE ports connected through the midplane to each half-width slot in the chassis. Typically configured in pairs for redundancy, two fabric extenders provide up to 160Gbps of I/O to the chassis. The Cisco UCS 2208XP Fabric Extender is shown in Figure 8.
Cisco Nexus 9000 Series Switch

Cisco Nexus 9000 Series switches deliver proven high performance and density, low latency, and exceptional power efficiency in a broad range of compact form factors. Operating in Cisco NX-OS Software mode or in Application Centric Infrastructure (ACI) mode, these switches are ideal for traditional or fully automated data center deployments. This NetApp technical report uses NX-OS mode on the 9396PX switch pair. The Cisco Nexus 9000 Series switch is shown in Figure 9.

The Cisco Nexus 9000 Series switches offer both modular and fixed 10/40/100GbE switch configurations with scalability up to 30Tbps of nonblocking performance with less than 5-microsecond latency, 1,152 10Gbps or 288 40Gbps, nonblocking layer 2 and layer 3 Ethernet ports and wire-speed VXLAN gateway, bridging, and routing support.

For more information, see Cisco Nexus 9000 Series Switches.

Cisco UCS for OpenStack

Cloud-enabled applications can run on organization premises, in public clouds, or on a combination of the two (hybrid cloud) for greater flexibility and business agility. Finding a platform that supports all these scenarios is essential. With Cisco UCS, IT departments can take advantage of the following technological advancements and lower the cost of their OpenStack deployments:

- **Open architecture.** A market-leading, open alternative to expensive, proprietary environments, the simplified architecture of Cisco UCS running OpenStack software delivers greater scalability, manageability, and performance at a significant cost savings compared to traditional systems, both in the data center and in the cloud. Using industry-standard x86-architecture servers and open-source software, IT departments can deploy cloud infrastructure today without concern for hardware or software vendor lock-in.

- **Accelerated cloud provisioning.** Cloud infrastructure must be able to flex on demand, providing infrastructure to applications and services on a moment's notice. Cisco UCS simplifies and accelerates cloud infrastructure deployment through automated configuration. The abstraction of Cisco UCS integrated infrastructure for RHEL server identity, personality, and I/O connectivity from the hardware allows these characteristics to be applied on demand. Every aspect of a server’s configuration, from firmware revisions and BIOS settings to network profiles, can be assigned through Cisco UCS service profiles. Cisco service profile templates establish policy-based configuration for server, network, and storage resources and can be used to logically preconfigure these resources even before they are deployed in the cloud infrastructure.

- **Simplicity at scale.** With IT departments challenged to deliver more applications and services in shorter periods, the architectural silos that result from an ad hoc approach to capacity scaling with traditional systems pose a barrier to successful cloud infrastructure deployment. Start with the computing and storage infrastructure needed today and then scale easily by adding components. Because servers and storage systems integrate into the unified system, they do not require additional supporting infrastructure or expert knowledge. The system simply, quickly, and cost-effectively presents more computing power and storage capacity to cloud infrastructure and applications.

- **Virtual infrastructure density.** Cisco UCS enables cloud infrastructure to meet ever-increasing guest OS memory demands on fewer physical servers. The system’s high-density design increases consolidation ratios for servers, saving the capital, operating, physical space, and licensing costs that would be needed to run virtualization software on larger servers.
- **Simplified networking.** In OpenStack environments, underlying infrastructure can become a sprawling complex of networked systems. Unlike traditional server architecture, Cisco UCS provides greater network density with less cabling and complexity. Cisco's unified fabric integrates Cisco UCS servers with a single, high-bandwidth, low-latency network that supports all system I/O. This approach simplifies the architecture and reduces the number of I/O interfaces, cables, and access-layer switch ports compared to the requirements for traditional cloud infrastructure deployments. This unification can reduce network complexity by up to a factor of three, and the system’s “wire once” network infrastructure increases agility and accelerates deployment with zero-touch configuration.

- **Installation confidence.** Organizations that choose OpenStack for their cloud can take advantage of the Red Hat OpenStack Platform director. This software performs the work needed to install a validated OpenStack deployment. Unlike other solutions, this approach provides a highly available, highly scalable architecture for OpenStack services.

- **Easy management.** Cloud infrastructure can be extensive, so it must be easy and cost effective to manage. Cisco UCS Manager provides embedded management of all software and hardware components in Cisco UCS. Cisco UCS Manager resides as embedded software on the Cisco UCS fabric interconnects, fabric extenders, servers, and adapters. No external management server is required, simplifying administration and reducing capital expenses for the management environment.

1.2 **Use Case Summary**

This document describes the deployment procedures and best practices to set up a FlexPod Datacenter deployment with Red Hat OpenStack Platform 8. The server operating system/hypervisor is RHEL 7.2, and an OpenStack deployment composed of three controller systems and four compute systems is built to provide an infrastructure as a service (IaaS) that is quick, easy to deploy, and scalable. As part of this solution, the following use cases were validated:

- Deliver an architecture and a prescriptive reference deployment that provide a high level of resiliency against component failure.
- Demonstrate simplified deployment instructions and automation templates to assist the customer with the deployment of Red Hat OpenStack Platform on a FlexPod system.
- Validate the solution by demonstrating common operations in an OpenStack deployment, functioning as they should.
- Demonstrate the scalability, speed, and space efficiency of VM creation in the resulting deployment.

2 **Solution Technology**

2.1 **Solution Hardware**

Red Hat OpenStack Platform 8 was validated using the hardware components listed in Table 1.

Table 1) Solution hardware.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td></td>
</tr>
<tr>
<td>NetApp FAS8040 storage controllers</td>
<td>2 nodes configured as an active-active pair</td>
</tr>
<tr>
<td><strong>Note:</strong> NetApp AFF8040 can be substituted for the FAS8040 if customer requirements dictate an all-flash configuration. <strong>Note:</strong> The solution has the capability to scale to 6 nodes in the case of SAN designs and to 24 nodes for non-SAN deployments. In either scenario, more nodes would be required.</td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>Quantity</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>NetApp DS2246 2U disk shelf</td>
<td>1 shelf</td>
</tr>
<tr>
<td>• 24 900GB 10K SAS drives</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> If using a NetApp AFF8040, substitute the above drives with SSDs.</td>
<td></td>
</tr>
<tr>
<td>NetApp E5600 storage controllers</td>
<td>2 nodes configured as an active-active pair, installed in a 4U DE6600 disk shelf.</td>
</tr>
<tr>
<td>• 60 3TB 7.2K SAS drives</td>
<td></td>
</tr>
<tr>
<td><strong>Compute</strong></td>
<td></td>
</tr>
<tr>
<td>Cisco UCS 6248UP fabric interconnects</td>
<td>2 configured as an active-active pair</td>
</tr>
<tr>
<td>Cisco UCS 5108 B-Series blade chassis</td>
<td>2; Requires the use of Cisco UCS 2208XP B-Series blade FEX modules</td>
</tr>
<tr>
<td>Cisco UCS 2208XP B-Series blade FEX modules</td>
<td>2 per Cisco UCS 5108 B-Series blade chassis</td>
</tr>
<tr>
<td>Cisco UCS B200 M3 B-Series blade servers</td>
<td>8</td>
</tr>
<tr>
<td>• 1 Cisco UCS 1240 VIC</td>
<td></td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td></td>
</tr>
<tr>
<td>Cisco Nexus 9396PX</td>
<td>2</td>
</tr>
</tbody>
</table>

### 2.2 Solution Software

Red Hat OpenStack Platform 8 was validated using the software components listed in Table 2.

Table 2) Software requirements.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version or Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Storage</strong></td>
<td></td>
</tr>
<tr>
<td>NetApp Data ONTAP</td>
<td>8.3.2</td>
</tr>
<tr>
<td>NetApp Data ONTAP licenses required</td>
<td>Cluster base, NFS, iSCSI, FlexClone</td>
</tr>
<tr>
<td>NetApp Data ONTAP licenses suggested</td>
<td>CIFS (for Manila shares utilizing CIFS/SMB)</td>
</tr>
<tr>
<td>NetApp SANtricity OS</td>
<td>8.25.04.00</td>
</tr>
<tr>
<td>OpenStack Cinder</td>
<td>NetApp unified driver distributed with Red Hat OpenStack Platform 8</td>
</tr>
<tr>
<td>OpenStack Swift</td>
<td>Packages distributed with Red Hat OpenStack Platform 8</td>
</tr>
<tr>
<td>OpenStack Manila</td>
<td>NetApp Manila driver distributed with Red Hat OpenStack Platform 8</td>
</tr>
<tr>
<td><strong>Compute</strong></td>
<td></td>
</tr>
<tr>
<td>Cisco UCS fabric interconnects</td>
<td>3.1(1e)</td>
</tr>
</tbody>
</table>
### 2.3 OpenStack Storage Elements

To make the right decisions to store and protect your data, it is important to understand the various types of storage that you may come across in the context of an OpenStack cloud.

#### Ephemeral Storage

Cloud users do not have access to any form of persistent storage by default if you only deploy Nova (compute service). When a user terminates a VM, the associated ephemeral disks are lost along with their data.

#### Persistent Storage

As the name suggests, persistent storage allows your saved data and storage resources to exist even if an associated instance is removed. OpenStack supports the types of persistent storage listed in Table 3.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block storage</td>
<td>Also called volume storage, users can use block storage volumes for their VM instance boot volumes and attached secondary storage volumes. Unlike ephemeral volumes, block volumes retain their data when they are remounted on another VM. Cinder provides block storage services in an OpenStack cloud. It enables access to the underlying storage hardware’s block device through block storage drivers. This results in improved performance and allows users to consume any feature or technology supported by the underlying storage hardware, such as deduplication, compression, and thin provisioning. To learn more about Cinder and block storage, visit <a href="https://wiki.openstack.org/wiki/Cinder">https://wiki.openstack.org/wiki/Cinder</a>.</td>
</tr>
<tr>
<td>File share systems</td>
<td>A share is a remote, mountable file system that can be shared among multiple hosts at the same time. The OpenStack file share service (Manila) is responsible for providing the required set of services for the management of shared file systems in a multitenant cloud.</td>
</tr>
</tbody>
</table>
To learn more about Manila and file share systems, visit https://wiki.openstack.org/wiki/Manila.

Object storage
The OpenStack object storage service (Swift) allows users to access binary objects through a REST API, which is useful for the management of large datasets in a highly scalable, highly available manner.
To learn more about Swift and object storage, visit http://docs.openstack.org/developer/swift/.

Table 4 summarizes the different storage types in an OpenStack cloud.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ephemeral Storage</td>
<td>Nova (compute)</td>
</tr>
<tr>
<td>Block Storage</td>
<td>Cinder</td>
</tr>
<tr>
<td>Shared File System</td>
<td>Manila</td>
</tr>
<tr>
<td>Object Storage</td>
<td>Swift</td>
</tr>
<tr>
<td>Persistence</td>
<td>Deleted when instance is deleted</td>
</tr>
<tr>
<td></td>
<td>Not deleted when instance is deleted¹; persists until deleted by user</td>
</tr>
<tr>
<td></td>
<td>Not deleted when instance is deleted; persists until deleted by user</td>
</tr>
<tr>
<td></td>
<td>Persists until deleted by user</td>
</tr>
<tr>
<td>Storage accessibility</td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td>Block device that can be formatted and used with a file system</td>
</tr>
<tr>
<td></td>
<td>Shared file system (NFS, CIFS, and so on)</td>
</tr>
<tr>
<td></td>
<td>REST API</td>
</tr>
<tr>
<td>Usage</td>
<td>Used for running the operating system of a VM</td>
</tr>
<tr>
<td></td>
<td>Used for providing additional block storage for VM instances²</td>
</tr>
<tr>
<td></td>
<td>Used for adding additional persistent storage or for sharing file systems among multiple instances</td>
</tr>
<tr>
<td></td>
<td>Used for storing and managing large datasets that may include VM images³</td>
</tr>
</tbody>
</table>

### 2.4 NetApp Storage for OpenStack

Most options for OpenStack integrated storage solutions aspire to offer scalability but lack the features and performance needed for efficient and cost-effective cloud deployment at scale.

NetApp has developed OpenStack integration that offers the value proposition of FAS, E-Series, and SolidFire® to enterprise customers, providing them with open-source options that provide lower cost, faster innovation, unmatched scalability, and the promotion of standards. Valuable NetApp features are accessible through the interfaces of standard OpenStack management tools (CLI, Horizon, and so on), allowing customers to benefit from simplicity and automation engineered by NetApp.

With access to NetApp technology features such as data deduplication, thin provisioning, cloning, Snapshot copies, DDP, and mirroring, among others, customers can be confident that the storage

¹ Except when delete on terminate option is used when launching an instance with the boot from image deployment choice.
² Can also be used for running the operating system of a VM through the usage of the boot from image deployment choice when launching an instance.
³ Cold storage only. An instance may not be actively running in Swift.
infrastructure underlying their OpenStack IaaS environment is highly available, is flexible, and performs well.

NetApp technology integrates with the following OpenStack services:

- Cinder
- Swift
- Glance
- Nova
- Manila

Users can build on this proven and highly scalable storage platform not only with greenfield deployments, as illustrated in this technical report, but also with brownfield deployments for customers who want to optimize their existing NetApp storage infrastructure.

Cinder

The OpenStack block storage service (Cinder) provides management of persistent block storage resources. In addition to acting as secondarily attached persistent storage, you can write images to a Cinder volume for Nova to utilize as a bootable, persistent root volume for an instance.

Cinder volumes are stored on the NetApp FAS8040 storage array and accessed using the pNFS (NFS version 4.1) protocol. The pNFS protocol has a number of advantages at scale in a large, heterogeneous hybrid cloud, including dynamic failover of network paths, high performance through parallelization, and an improved NFS client.

Red Hat OpenStack Platform Director Integration

As a part of an OpenStack deployment, the Red Hat OpenStack Platform director tool handles the Cinder service configuration (and pertinent configuration files on the resulting controller hosts) automatically. An overridden environment template is passed to the overcloud deployment script to match the specific environment configuring Cinder to use the NetApp unified driver during the core Puppet configuration, after nodes are provisioned.

Environment template options include support for NetApp clustered Data ONTAP, Data ONTAP operating in 7-Mode, and E-Series platforms. For this reference architecture, NetApp clustered Data ONTAP was selected, and pertinent details that are reflective of a NetApp FAS8040 storage subsystem were filled in.

NetApp Unified Driver for Clustered Data ONTAP with NFS

A Cinder driver is a particular implementation of a Cinder back end that maps the abstract APIs and primitives of Cinder to appropriate constructs within the particular storage solution underpinning the Cinder back end.

The NetApp unified driver for clustered Data ONTAP with NFS is a driver interface from OpenStack block storage to a Data ONTAP cluster system. This software provisions and manages OpenStack volumes on NFS exports provided by the Data ONTAP cluster system. The NetApp unified driver for the Data ONTAP cluster does not require any additional management software to achieve the desired functionality because it uses NetApp APIs to interact with the Data ONTAP cluster. It also does not require additional configuration beyond overriding default templates during OpenStack deployment using the Red Hat OpenStack Platform director.

All of the resulting Cisco UCS blades that are provisioned with RHEL 7.2 mount the (appropriately designated) NetApp FlexVol volumes on the FAS8040 at the highest protocol level possible for the instance volumes (NFS version 4.1 or Parallelized NFS [pNFS]).

Note: A FlexVol volume is a logical container for one or more Cinder volumes.
The NetApp OpenStack contribution strategy adds all new capabilities directly into upstream OpenStack repositories, so that all of the features are available in Red Hat OpenStack Platform 8.0 out of the box. For more information regarding the NetApp unified driver (including other protocols available), see NetApp Data ONTAP Drivers for OpenStack Block Storage (Cinder).

For more information about why NFS was chosen over iSCSI, see Deployment Choice: NFS Versus iSCSI.

Storage Service Catalogs

A storage service catalog (SSC) enables efficient, repeatable, and consistent use and management of storage resources by defining policy-based services and mapping those services to the back-end storage technology. It abstracts the technical implementations of features at a storage back end into a set of simplified configuration options.

These storage features are organized or combined into groups based on a customer’s particular scenario or use case. Based on this catalog of storage features, intelligent provisioning decisions are made by infrastructure or software enabling the SSC. In OpenStack, this is achieved by the Cinder and Manila filter schedulers, through their respective NetApp drivers, by making use of volume-type, extra-specs support, together with the respective filter scheduler. There are some prominent features that are exposed by using the NetApp drivers, including mirroring, deduplication, compression, and thin provisioning. Workloads can be tied to Cinder volume types or Manila share types in an OpenStack context, and these workloads then have NetApp technologies enabled on the storage system. Examples of needed functionalities include:

- Transactional databases that require high IOPS with SSD disks and data protection
- Test and development workloads that benefit from thin provisioning and compression
- Disaster recovery processes that require a SnapMirror relationship to another NetApp storage system

When you use the NetApp unified driver with a clustered Data ONTAP storage system, you can leverage extra specs to make sure that Cinder volumes or Manila shares are created on storage back ends that have certain features configured (for example, deduplication, thin provisioning, disk types [flash, SAS, SATA, and so on], and compression). These extra specs are associated with either Cinder volume types or Manila share types, so that when users request volumes or shares, respectively, they are created on storage back ends that meet the list of requirements (for example, available space, extra specs, and so on).

For a list of extra specs that can be associated with Cinder volume types and thus leveraged in an environment that uses NetApp clustered Data ONTAP, see Using Cinder Volume Types to Create a Storage Service Catalog.

For a listing of extra specs that can be associated with Manila share types and thus leveraged in an environment that uses NetApp clustered Data ONTAP, see Using Manila Share Types to Create a Storage Service Catalog.

Swift

The OpenStack object store project (Swift) provides a fully distributed, scale-out-ready, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving, and data retention and retrieval. Object storage does not present a traditional file system, but rather a distributed storage system for static, unstructured data objects such as VM images, photo storage, e-mail storage, backups, and archives.

Customers can start with the ultradense 60-drive enclosure, as demonstrated in this technical report, and then can scale horizontally with multiple controller pairs as the size of the object storage needs grow. The NetApp E5660 storage array hosts Swift data using the iSCSI protocol. Three of the Cisco UCS servers
are used as Swift nodes and handle account, container, and object services. In addition, these three nodes also serve as proxy servers for the Swift service.

**E-Series Resiliency**

E-Series storage hardware serves effectively as the storage medium for Swift. The data reconstruction capabilities associated with DDP eliminates the need for data replication within zones in Swift. DDP reconstruction provides RAID 6–like data protection against multiple simultaneous drive failures within the storage subsystem. Data that resides on multiple failed drives has top priority during reconstruction. This data has the highest potential for being lost if a third drive failure occurs and is, therefore, reconstructed first on the remaining optimal drives in the storage subsystem. After this critical data is reconstructed, all other data on the failed drives is reconstructed. This prioritized data reconstruction dramatically reduces the possibility of data loss due to drive failure.

As disk sizes increase, the rebuild time after failure also increases. The time taken by the traditional RAID system to rebuild after a failure to an idle spare is longer. This is because the idle spare in the traditional RAID system receives all of the write traffic during a rebuild, slowing down the system and data access during this process. One of the main goals of DDP is to spread the workload around if a disk fails and its data must be rebuilt. This provides consistent performance, keeps you in the green zone, and maintains nondisruptive performance. DDP has shown the ability to provide up to eight times faster reconstruction of a failed disk data throughout the pool when compared to an equivalent, standard RAID configuration disk rebuild.

The dynamic process of redistributing the data occurs in the background in a nondisruptive, minimal-impact manner so that the I/O continues to flow.

**Scalability on NetApp E-Series**

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 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. E-Series data reconstruction makes sure that clients always have access to data regardless of drive or other component failures within the storage subsystem. When E-Series storage is used, Swift data replication counts that are specified when rings are built can be reduced from three to two. This reduces both the replication traffic normally sent on the standard IPv4 data center networks and the amount of storage required to save copies of the objects in the Swift cluster.

**Reduction in Physical Resources Using Swift on NetApp E-Series**

In addition to the previously discussed issues, using Swift on NetApp E-Series enables:

- **Reduced Swift-node hardware requirements.** Internal drive requirements for storage nodes are reduced, and only operating system storage is required. Disk space for Swift object data, and optionally the operating system itself, is supplied by the E-Series storage array.
- **Reduced rack space, power, cooling, and footprint requirements.** Because a single storage subsystem provides storage space for multiple Swift nodes, no dedicated physical servers with direct-attached storage (DAS) are required for data storage and retention of Swift data.
**Red Hat OpenStack Platform Director Integration**

Swift is installed automatically through the Red Hat OpenStack Platform director. Heat orchestration templates (HOT) are provided in this technical report to override the default behavior of installing Swift on the local root disk of the controller nodes to the NetApp E-Series E5660 instead.

For more information regarding Swift on NetApp, see [OpenStack Object Storage Service (Swift)](https://www.netapp.com/enterprise/solutions/openstack/).

**Glance**

The OpenStack image service (Glance) provides discovery, registration, and delivery services for virtual machine, 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, as opposed to 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.

In this technical report, Glance uses NFS version 4.0 to communicate with the NetApp FAS8040 storage array using Pacemaker in a highly available configuration.

**Red Hat OpenStack Platform Director Integration**

Glance configuration (much like Cinder) is handled by passing an overridden environment template to the overcloud deployment script in Red Hat OpenStack Platform director. An example template is provided along with this technical report to aid customers in configuring Glance to take advantage of an already configured NetApp FlexVol volume through NFS with deduplication enabled.

**Note:** Because there is a high probability of duplicate blocks in a repository of VM images, NetApp highly recommends enabling deduplication on the FlexVol volumes where the images are stored.

**Image Formats: QCOW and Raw**

Glance supports a variety of image formats, but RAW and QCOW2 are the most common. QCOW2 does provide some advantages over the RAW format (for example, the support of copy-on-write, snapshots, and dynamic expansion). However, when images are copied into Cinder volumes, they are automatically converted into the RAW format after being stored on a NetApp back end. Therefore:

- NetApp recommends the QCOW2 image format for ephemeral disks due to its inherent benefits when taking instance Snapshot copies.
- The RAW image format can be advantageous when Cinder volumes are used as persistent boot disks because Cinder does not have to convert from an alternate format to RAW.

Both the RAW and QCOW2 formats respond well to NetApp deduplication technology, which is often used with Glance deployments.

QCOW2 is not live migration safe on NFS when the `cache=writeback` setting is enabled, which is commonly used for performance improvement of QCOW2. If space savings are the desired outcome for the image store, RAW format files are actually created as sparse files on the NetApp storage system. Deduplication within NetApp FlexVol volumes happens globally rather than only within a particular file, resulting in much better aggregate space efficiency than QCOW2 can provide. Deduplication processing can be finely controlled to run at specific times of day (off peak).

**Copy Offload Tool**

The NetApp copy offload feature was added in the Icehouse release of OpenStack, which enables images to be efficiently copied to a destination Cinder volume that is backed by a clustered Data ONTAP FlexVol volume. When Cinder and Glance are configured to use the NetApp NFS copy offload client, a controller-side copy is attempted before reverting to downloading the image from the image service. This improves image-provisioning times while reducing the consumption of bandwidth and CPU cycles on the...
hosts running the Glance and Cinder services. This is due to the copy operation being performed completely within the storage cluster.

Although the copy offload tool can be configured automatically as a part of an OpenStack deployment using Heat orchestration templates (documented in this technical report), it must still be downloaded from the NetApp Utility ToolChest site by the customer.

Note: If Cinder and Glance share the same NetApp FlexVol volume, the copy offload tool is not necessary. Rather, a direct API call to the NetApp storage system is utilized through the NetApp unified driver that facilitates a controller-side copy relative to a network copy.

For more information about this functionality, including a helpful process flowchart, see OpenStack Deployment and Operations Guide - Version 6.0, Theory of Operation and Deployment Choices, Glance and Clustered Data ONTAP.

Rapid Cloning

NetApp provides two capabilities that enhance instance booting by using persistent disk images in the shortest possible time and in the most storage capacity–efficient manner possible: the NetApp copy offload tool and instance caching.

The enhanced persistent instance creation feature (sometimes referred to as rapid cloning) uses NetApp FlexClone technology and the NetApp copy offload tool. Rapid cloning can significantly decrease the time needed for the Nova service to fulfill image provisioning and boot requests. It also supports much larger images with no noticeable degradation of boot time.

One feature that facilitates rapid cloning in an NFS/pNFS setup within the NetApp unified driver is instance caching. Whenever a Cinder volume is created out of a Glance template, it is cached locally on the NetApp FlexVol volume that hosts the Cinder volume instance. Later, when you want to create the same OS instance again, Cinder creates a space-efficient file clone. This clone does not take up any more space because it shares the same blocks as the cached image. Only deltas take up new blocks on disk. Figure 10 illustrates this concept.

This not only makes the instance and Cinder volume creation operations faster, but also reduces the CPU load on the Cinder and Glance hosts and reduces the network traffic almost completely. The cache also provides a time-to-live option, which invalidates old cache entries automatically after a specified period of time.
For more information regarding Glance on NetApp, see OpenStack Image Service (Glance).

**Nova**

The OpenStack compute service (Nova) is a cloud computing fabric controller that is the primary part of an IaaS system. You can use Nova to host and manage cloud instances (VMs).

**Root and Ephemeral Disks**

Each instance requires at least one root disk containing the bootloader and core operating system files, and each instance might also have optional ephemeral disks that use the definition of the type selected at instance creation time. The content for the root disk comes either from an image stored within the Glance repository, which is copied to storage attached to the destination hypervisor, or from a persistent block storage volume through Cinder.

By selecting the Boot from Image (Creates a New Volume) option in Nova, you can leverage the rapid cloning capabilities described previously. Normally volumes created as a result of this option are persistent beyond the life of the instance. However, you can select the Delete on Terminate option in combination with the Boot from Image (Creates a New Volume) option to create an ephemeral volume while still leveraging the rapid cloning capabilities described in the Rapid Cloning section. This can provide a significantly faster provisioning and boot sequence relative to the normal way that ephemeral disks are provisioned. In the normal way, a copy of the disk image is made from Glance to local storage on the hypervisor node where the instance resides.

**Note:** A Glance instance image of 20GB can, for example, be cloned in 300ms using NetApp FlexClone technology.

For more information about using the Nova service in conjunction with NetApp, see OpenStack Compute Service (Nova).

**Manila**

The OpenStack shared file system service (Manila) provides management of persistent shared file system resources. Much of the total storage shipped worldwide is based on shared file systems, and, with help from the OpenStack community, NetApp is delivering these capabilities to the OpenStack environment. Before Manila, OpenStack had only the Cinder module for block files.

NetApp designed, prototyped, and built the Manila module, which is the Cinder equivalent for shared or distributed file systems. Manila emerged as an official, independent project in the Grizzly release of OpenStack.

Manila is typically deployed in conjunction with other OpenStack services (compute, object storage, image, and so on) as part of a larger, more comprehensive cloud infrastructure. This is not an explicit requirement, because Manila has been successfully deployed as a standalone solution for shared file system provisioning and lifecycle management.

**Note:** Although still a technology preview in Red Hat OpenStack Platform 8, this technical report demonstrates how to configure and enable OpenStack Manila as a part of an OpenStack deployment using the Red Hat OpenStack Platform director in a highly available manner, through the usage of postdeployment scripts specifically written for this technical report.

For more information about using the Manila service in conjunction with NetApp, see OpenStack File Share Service (Manila).

### 2.5 Red Hat OpenStack Platform Director

The Red Hat OpenStack Platform director is a tool set for installing and managing a complete OpenStack environment. It is based primarily on the OpenStack TripleO project; TripleO is an abbreviation for "OpenStack on OpenStack." This project takes advantage of OpenStack components to install a fully
operational OpenStack environment. This includes new OpenStack components that provision and control bare metal systems to use as OpenStack nodes. This provides a simple method for installing a complete Red Hat OpenStack environment that is both lean and robust.

Red Hat OpenStack Platform director provides the following benefits:

- Simplified deployment through ready-state provisioning of bare metal resources.
- Flexible network definitions.
- Deployment with confidence (Red Hat OpenStack Platform provides a hardened and stable branch release of OpenStack and Linux, which is supported by Red Hat for a three-year production lifecycle, well beyond the typical six-month release cadence of unsupported community OpenStack.)
- HA through integration with the Red Hat Enterprise Linux server high-availability add-on.
- Content management using the Red Hat content delivery network or Red Hat satellite server.
- SELinux-enforced data confidentiality and integrity, as well as process protection from untrusted inputs using a preconfigured and hardened security layer.

**Note:** SELinux is deployed throughout the resulting OpenStack deployment.

The Red Hat OpenStack Platform director uses two main concepts: an undercloud and an overcloud, as shown in Figure 11. The undercloud installs and configures the overcloud. The next few sections outline the concepts of each.

**Figure 11) Overcloud and undercloud relationship.**

**Undercloud**

The undercloud is the main director node. It is a single-system OpenStack installation that includes components for provisioning and managing the OpenStack nodes that compose your OpenStack environment (the overcloud). The components that form the undercloud provide the following functions:

- **Environment planning.** The undercloud provides planning functions for users to assign Red Hat OpenStack Platform roles, including compute, controller, and various storage roles.
- **Bare metal system control.** The undercloud uses the intelligent platform management interface (IPMI) of each node for power management control and a preboot execution environment (PXE)—based service to discover hardware attributes and install OpenStack to each node. This provides a method to provision bare metal systems as OpenStack nodes.
Orchestration. The undercloud provides and reads a set of YAML templates to create an OpenStack environment.

The Red Hat OpenStack Platform director uses undercloud functions through both a web-based GUI and a terminal-based CLI.

The undercloud uses the components listed in Table 5.

Table 5) Undercloud components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Code Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenStack dashboard</td>
<td>Horizon</td>
<td>The web-based dashboard for the Red Hat OpenStack Platform director</td>
</tr>
<tr>
<td>OpenStack bare metal</td>
<td>Ironic</td>
<td>Manages bare metal nodes</td>
</tr>
<tr>
<td>OpenStack compute</td>
<td>Nova</td>
<td>Manages bare metal nodes</td>
</tr>
<tr>
<td>OpenStack networking</td>
<td>Neutron</td>
<td>Controls networking for bare metal nodes</td>
</tr>
<tr>
<td>OpenStack image server</td>
<td>Glance</td>
<td>Stores images that are written to bare metal machines</td>
</tr>
<tr>
<td>OpenStack orchestration</td>
<td>Heat</td>
<td>Provides orchestration of nodes and configuration of nodes after the director writes the overcloud image to disk</td>
</tr>
<tr>
<td>OpenStack telemetry</td>
<td>Ceilometer</td>
<td>For monitoring and data collection</td>
</tr>
<tr>
<td>OpenStack identity</td>
<td>Keystone</td>
<td>Authentication for the director’s components</td>
</tr>
</tbody>
</table>

The following components are also used by the undercloud:

- **Puppet.** Declarative-based configuration management and automation framework
- **MariaDB.** Database for the Red Hat OpenStack Platform director
- **RabbitMQ.** Messaging queue for the Red Hat OpenStack Platform director components

**Overcloud**

The overcloud is the resulting Red Hat OpenStack Platform environment created using the undercloud. The following nodes are illustrated in this technical report:

- **Controller nodes.** Provide administration, networking, and HA for the OpenStack environment. An ideal OpenStack environment recommends three of these nodes together in an HA cluster:
  - A default controller node contains the following components: Horizon, Keystone, Nova API, Neutron Server, Open vSwitch, Glance, Cinder Volume, Cinder API, Swift Storage, Swift Proxy, Heat Engine, Heat API, Ceilometer, MariaDB, RabbitMQ.
  - The controller node also uses Pacemaker and Galera for high-availability functions.
- **Compute nodes.** Provide computing resources for the OpenStack environment. Add more compute nodes to scale your environment over time to handle more workloads:
  - A default compute node contains the following components: Nova Compute, Nova KVM, Ceilometer Agent, Open vSwitch.

**High Availability**

HA provides continuous operation to a system or components set through an extended length of time. The Red Hat OpenStack Platform director provides HA to an OpenStack Platform environment through the use of a controller node cluster. The director installs a set of the same components on each controller.
node and manages them as one whole service. Having a cluster provides a fallback in case of operational failures on a single controller node. This provides OpenStack users with a certain degree of continuous operation.

The Red Hat OpenStack Platform director uses some key pieces of software to manage components on the controller node:

- **Pacemaker.** A cluster resource manager that manages and monitors the availability of OpenStack components across all machines in the cluster
- **HA Proxy.** Provides load balancing and proxy services to the cluster
- **Galera.** Provides replication of the Red Hat OpenStack Platform database across the cluster
- **Memcached.** Provides database caching

## 2.6 Heat Orchestration Templates (HOT) and Overcloud Customization

The Red Hat OpenStack Platform director uses Heat orchestration templates (HOT) as a template format for its overcloud deployment plan; they are expressed in YAML format. The purpose of a template is to define and create a stack, which is a collection of resources that Heat creates on a per-resource basis. Resources are objects in OpenStack and can include compute resources, network configuration, security groups, scaling rules, and so on.

Custom templates can also be developed for unique requirements or to automate portions of the overcloud creation to suit individual needs.

NetApp has contributed several custom Heat templates, postdeployment scripts, and exhibits from our laboratory to aid the reader in configuring the Red Hat OpenStack Platform director with additional parameters and configuration necessary for an eventual OpenStack deployment on FlexPod.

These templates are hosted in a repository specific to NetApp on GitHub and can be accessed here: [https://github.com/NetApp/snippets/tree/master/RedHat/osp8-liberty/tr](https://github.com/NetApp/snippets/tree/master/RedHat/osp8-liberty/tr).

Proper networking segmentation is critical for the systems composing the eventual OpenStack deployment. 802.1Q VLANs and specific subnets for the OpenStack services in the overcloud accomplish this. There are two types of templates required to achieve network segmentation in the overcloud:

- Network interface card (NIC) templates
- Network environment templates

### NIC Templates

NIC templates are YAML files that detail which vNICs assigned to the Cisco UCS service profile are assigned to which VLANs composed in the overcloud, their respective MTU settings, and which interfaces are bonded together from a link aggregation perspective. Other items worth noting include:

- For the compute role, the storage, internal API, external, tenant, and IPMI networks are tagged down to the enp9s0 and enp10s0 physical interfaces in the Cisco UCS service profiles.
- For the controller role, the storage, storage management, internal API, external, floating IP, tenant, and IPMI networks are tagged down to the enp9s0 and enp10s0 physical interfaces in the Cisco UCS service profiles.

The following files are included on GitHub:

**compute.yaml**

The `compute.yaml` file should be located in the `/home/stack/flexpod-templates/nic-configs` directory on the Red Hat OpenStack Platform director server.
This template configures the resulting overcloud deployment’s compute servers to use the enp9s0 and enp10s0 interfaces bonded together in a link aggregation in Linux. The following interfaces are also configured for jumbo frames in this template:

- Enp9s0 and enp10s0 physical device interfaces in Linux
- Internal API VLAN (OSP-Backend VLAN 421 in our environment)
- Storage VLAN (NFS VLAN 67 in our environment)

**controller.yaml**

The controller.yaml file should be located in the /home/stack/flexpod-templates/nic-configs directory on the Red Hat OpenStack Platform director server.

This template configures the resulting overcloud deployment’s controller servers to use the enp9s0 and enp10s0 interfaces bonded together in a link aggregation in Linux. The following interfaces are also configured for jumbo frames in this template:

- Enp9s0 and enp10s0 physical device interfaces in Linux
- Internal API VLAN (OSP-Backend VLAN 421 in our environment)
- Storage VLAN (NFS VLAN 67 in our environment)
- Storage management VLAN (OSP-StorMgmt VLAN 99 in our environment)

**Network Environment Templates**

Network environment templates are YAML files that detail the subnetting and VLAN configuration in the overcloud. Because OpenStack requires a number of segmented networks and respective IP addresses for the various services and daemons involved in the overall architecture, making sure this file is correct—with no overlap or duplicate address assignment with the rest of the overall network infrastructure—is critical for success.

The following files are included on GitHub:

**network-environment.yaml**

The network-environment.yaml file should be located in the /home/stack/flexpod-templates directory on the Red Hat OpenStack Platform director server.

The network-environment.yaml file describes the overcloud network environment and points to the network interface configuration files (controller.yaml and compute.yaml). We define the subnets for our network along with IP address ranges. These values can be customized for the local environment.

Table 6 lists the variables in this file.

**Table 6) network-environment.yaml variable definitions.**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Variable Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlPlaneDefaultRoute</td>
<td>Input the IP address of the Red Hat OpenStack Platform director server’s PXE/provisioning NIC.</td>
</tr>
<tr>
<td>EC2MetadataIP</td>
<td>Input the IP address of the Red Hat OpenStack Platform director server’s PXE/provisioning NIC.</td>
</tr>
<tr>
<td>DnsServers</td>
<td>Input the DNS servers to be utilized by the overcloud servers.</td>
</tr>
<tr>
<td>Variable</td>
<td>Variable Definition</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ExternalNetCidr</td>
<td>Classless interdomain routing (CIDR) notation entry for the external network. In our lab, this rides in higher order addresses on the OOB-Management subnet defined on the Cisco Nexus 9000 switch pair.</td>
</tr>
<tr>
<td>ExternalNetAllocationPools</td>
<td>Range of addresses used for the external network used in the overcloud.</td>
</tr>
<tr>
<td>ExternalNetworkVlanID</td>
<td>The 802.1Q VLAN tag for the external network used in the overcloud.</td>
</tr>
<tr>
<td>ExternalInterfaceDefaultRoute</td>
<td>The default gateway for the servers used in the overcloud. This was chosen to specifically be different from the Red Hat OpenStack Platform director server’s PXE/provisioning NIC to provide redundancy in case of the Red Hat OpenStack Platform director server’s failure.</td>
</tr>
<tr>
<td>InternalApiNetCidr</td>
<td>CIDR notation entry for the internal API network. In our lab, this rides in lower order addresses on the OSP-Backend subnet defined on the Cisco Nexus 9000 switch pair.</td>
</tr>
<tr>
<td>InternalApiAllocationPools</td>
<td>Range of addresses used for the internal API network used in the overcloud.</td>
</tr>
<tr>
<td>InternalApiNetworkVlanID</td>
<td>The 802.1Q VLAN tag for the internal API network used in the overcloud. This VLAN is dedicated to the overcloud and has no other infrastructure using it.</td>
</tr>
<tr>
<td>StorageNetCidr</td>
<td>CIDR notation entry for the storage network. In our lab, this rides in higher order addresses on the NFS subnet.</td>
</tr>
<tr>
<td>StorageAllocationPools</td>
<td>Range of addresses used for the storage network used in the overcloud.</td>
</tr>
<tr>
<td>StorageNetworkVlanID</td>
<td>The 802.1Q VLAN tag for the storage network used in the overcloud. This VLAN is shared with the NetApp FAS8040’s NFS interfaces on both nodes.</td>
</tr>
<tr>
<td>StorageMgmtNetCidr</td>
<td>CIDR notation entry for the storage management network. In our lab, this rides on higher order addresses in the OSP-StorMgmt subnet defined on the Cisco Nexus 9000 switch pair.</td>
</tr>
<tr>
<td>StorageMgmtAllocationPools</td>
<td>Range of addresses used for the storage management network used in the overcloud.</td>
</tr>
<tr>
<td>StorageMgmtAllocationVlanID</td>
<td>The 802.1Q VLAN tag for the storage management network used in the overcloud. This VLAN is dedicated to the overcloud and has no other infrastructure using it.</td>
</tr>
<tr>
<td>TenantNetCidr</td>
<td>CIDR notation entry for the tenant network. In our lab, this rides on higher order addresses in the tunnel subnet defined on the Cisco Nexus 9000 switch pair.</td>
</tr>
<tr>
<td>TenantAllocationPools</td>
<td>Range of addresses used for the tenant network used in the overcloud.</td>
</tr>
<tr>
<td>TenantNetworkVlanID</td>
<td>The 802.1Q VLAN tag for the tenant network used in the overcloud. This VLAN is dedicated to the overcloud and has no other infrastructure using it.</td>
</tr>
</tbody>
</table>
Because we’re using channel bonding in Linux, active backup or bonding mode 1 is used on the servers in the overcloud.

### FlexPod Templates

These HOT templates were developed specifically for this technical report. The following are included on GitHub, and all pertinent variable definitions inside the respective files are defined in this section.

These templates can be found in a subdirectory of the NetApp GitHub repository and are available directly at https://github.com/NetApp/snippets/tree/master/RedHat/osp8-liberty/tr/flexpod-templates.

The following files are included on GitHub:

#### flexpod.yaml

The `flexpod.yaml` file should be located in the `/home/stack/flexpod-templates` directory on the director server.

This is the main template demonstrated in this technical report, because it is passed as an environment argument (-e) to the overcloud deployment directly using the Red Hat OpenStack Platform director.

The resource_register portion of the template is defined as follows:

- **OS::TripleO::NodeExtraConfig**. The `flexpod-allnodes-pre.yaml` template file is given as an argument here. These tasks are done before the core Puppet configuration and customization by the Red Hat OpenStack Platform director.

- **OS::TripleO::ControllerExtraConfigPre**. The `flexpod-allcontrollers-pre.yaml` template file is given as an argument here. These tasks are done on the controller systems before the core Puppet configuration done by the Red Hat OpenStack Platform director.

- **OS::TripleO::NodeExtraConfigPost**. The `flexpod-allnodes-post.yaml` template file is given as an argument here. These tasks are done on all servers in the overcloud after the core Puppet configuration is done by the Red Hat OpenStack Platform director.

The parameter_defaults portion of the template consists of user variables that are customized before the Red Hat OpenStack Platform director initiates an OpenStack deployment. These variables are defined in Table 7.

**Table 7) `flexpod.yaml` variable definitions.**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Variable Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CinderNetappLogin</td>
<td>Administrative account name used to access the backend or its proxy server. For this parameter, you can use an account with cluster-level administrative permissions (namely, admin) or a cluster-scoped account with the appropriate privileges.</td>
</tr>
<tr>
<td>CinderNetappPassword</td>
<td>The corresponding password of CinderNetappLogin.</td>
</tr>
<tr>
<td>CinderNetappServerHostname</td>
<td>The value of this option should be the IP address or host name of either the cluster management LIF or SVM LIF.</td>
</tr>
<tr>
<td>CinderNetappServerPort</td>
<td>The TCP port that the block storage service should use to communicate with the NetApp back end. If not specified, Data ONTAP drivers use 80 for HTTP and 443</td>
</tr>
<tr>
<td>Variable</td>
<td>Variable Definition</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------</td>
</tr>
<tr>
<td><strong>CinderNetappStorageFamily</strong></td>
<td>The storage family type used on the back-end device. Use ontap_cluster for clustered Data ONTAP, ontap_7mode for Data ONTAP operating in 7-Mode, or eseries for E-Series.</td>
</tr>
<tr>
<td><strong>CinderNetappStorageProtocol</strong></td>
<td>The storage protocol to be used. NFS is utilized in this technical report.</td>
</tr>
<tr>
<td><strong>CinderNetappTransportType</strong></td>
<td>Transport protocol to be used for communicating with the back end. Valid options include http and https.</td>
</tr>
<tr>
<td><strong>CinderNetappVserver</strong></td>
<td>Specifies the name of the SVM where Cinder volume provisioning should occur. This refers to a single SVM on the storage cluster.</td>
</tr>
<tr>
<td><strong>CinderNetappNfsShares</strong></td>
<td>Comma-separated list of data LIFs exported from the NetApp Data ONTAP device to be mounted by the controller nodes. This list gets written to the location defined by CinderNetappNfsSharesConfig.</td>
</tr>
<tr>
<td><strong>CinderNetappNfsSharesConfig</strong></td>
<td>Absolute path to the NFS exports file. This file contains a list of available NFS shares to be used as a back end, separated by commas.</td>
</tr>
<tr>
<td><strong>CinderNetappCopyOffloadToolPath</strong></td>
<td>Specifies the path of the NetApp copy offload tool binary. This binary (available from the NetApp Support portal) must have the execute permissions set, because the openstack-cinder-volume process needs to execute this file.</td>
</tr>
<tr>
<td><strong>GlanceNetappCopyOffloadMount</strong></td>
<td>Specifies the NFS export used by Glance to facilitate rapid instance creation should Glance and Cinder use different FlexVol volumes under the same storage SVM on a NetApp clustered Data ONTAP system. See this link for more information.</td>
</tr>
<tr>
<td><strong>SwiftReplicas</strong></td>
<td>Number of replica copies performed by OpenStack Swift.</td>
</tr>
<tr>
<td><strong>SwiftNetappEseriesHic1Pl</strong></td>
<td>Specifies the IP address of the iSCSI interface configured on the NetApp E-Series host interface card (HIC) 1, port 1.</td>
</tr>
<tr>
<td><strong>SwiftNetappEseriesLuns</strong></td>
<td>Specifies the LUNs exposed to the controller systems in the overcloud from the NetApp E-Series storage system. This should be the same across all three controllers and is passed as a space-delimited array.</td>
</tr>
<tr>
<td><strong>CloudDomain</strong></td>
<td>The DNS domain name to be used by the overcloud.</td>
</tr>
<tr>
<td><strong>glance::api::show_image_direct_url</strong></td>
<td>Set this value to True to override Glance so that it places the direct URL of image uploads into the metadata stored in the Galera database. The NetApp copy offload tool requires this information in order to function.</td>
</tr>
<tr>
<td>Variable</td>
<td>Variable Definition</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CinderEnableIscsiBackend</td>
<td>Set this value to False to disable the default iSCSI back end being presented to the servers in the overcloud from the director server. This back end is not needed.</td>
</tr>
<tr>
<td>CinderEnableRbdBackend</td>
<td>Set this value to False to disable the RBD back end for Cinder. This is not needed in this technical report.</td>
</tr>
<tr>
<td>GlanceBackend</td>
<td>Set this value to False to disable the default iSCSI back end being presented to the servers in the overcloud from the director server. This back end is not needed.</td>
</tr>
<tr>
<td>GlanceFilePcmkManage</td>
<td>Whether to make Glance file back end a mount managed by Pacemaker. Set this to True. Effective when GlanceBackend is File.</td>
</tr>
<tr>
<td>GlanceFilePcmkFstype</td>
<td>Set this value to False to disable the default iSCSI back end being presented to the servers in the overcloud from the director server. This back end is not needed.</td>
</tr>
<tr>
<td>GlanceFilePcmkDevice</td>
<td>Specifies the NFS export used by Glance and backed by the NetApp FAS device. This is mounted and tracked by Pacemaker during the overcloud deployment.</td>
</tr>
<tr>
<td>GlanceFilePcmkOptions</td>
<td>Mount options for Pacemaker mount used as Glance storage. Special allowances for SELinux need to be accounted for here. Pass context=system_u:object_r:glance_var_lib_t:s0.</td>
</tr>
</tbody>
</table>

flexpod-allnodes-pre.yaml

The `flexpod-allnodes-pre.yaml` file should be located in the `/home/stack/flexpod-templates` directory on the director server.

This template runs on all of the servers provisioned in the overcloud before the core Puppet configuration is done by the Red Hat OpenStack Platform director. It performs the following tasks:

1. Updates the iSCSI initiator name on the server to the value stored in the iSCSI Boot Firmware table (IBFT).
2. Restarts the iscsid service to pick up the changes made in step 1. This is necessary so that the correct initiator name is used to log in to the NetApp E-Series system in future steps and provision it to be a target for Swift.
3. Adds two additional paths that are exposed from the NetApp FAS8040 system for the optimum number of available paths for ALUA-based failover in the DM-multipath subsystem.

This template is not meant to be modified.

flexpod-allcontrollers-pre.yaml

The `flexpod-allcontrollers-pre.yaml` file should be located in the `/home/stack/flexpod-templates` directory on the Red Hat OpenStack Platform director server.

This template runs on all of the controller servers provisioned in the overcloud before the core Puppet configuration is done by the Red Hat OpenStack Platform director. This template can be loosely defined as a wrapper template, meaning it is used to chain several different templates together to all be run on the controller systems before the core Puppet configuration is done by the director in the overcloud. For more context on the wrapper template functionality, refer to [https://bugzilla.redhat.com/show_bug.cgi?id=1313049](https://bugzilla.redhat.com/show_bug.cgi?id=1313049).
This template performs the following tasks:

1. Launches the cinder-netapp.yaml template file. Contained in the /home/stack/flexpod-templates/netapp-extra directory, this template automates the installation and configuration of the NetApp unified driver in /etc/cinder/cinder.conf and mounts NFS-backed FlexVol volumes set up in this technical report in section 3.5, “NetApp FAS8040 Setup.”

2. Launches the netapp-copyoffload.yaml template. Contained in the /home/stack/flexpod-templates/netapp-extra directory, this template automates the configuration of the NetApp copy offload tool and writes the necessary JSON information required for the tool to operate in /etc/glance/netapp.json. The tool binary itself is copied to the controller servers as a function of the post-deployment scripts demonstrated later.

3. Launches the netapp-eseries.yaml template. Contained in the /home/stack/flexpod-templates/netapp-extra directory, this template automates the Linux-related installation and setup of the NetApp E-Series storage system on the controller systems. This includes:
   a. Logs in to the E-Series system using iscsiadm.
   b. Partitions all of the E-Series system LUNs exposed to the controller systems with a single partition with an XFS label.
   c. Formats the partition with XFS.
   d. Writes custom udev rules to make sure that the partitions are mounted properly by their UUID.
   e. Mounts the E-Series LUNs to the /srv/node/eserieslun* directory structure to be consumed by the core Puppet configuration as done by the director.

None of the preceding templates are meant to be modified.

flexpod-allcontrollers-post.yaml

The flexpod-allcontrollers-post.yaml file should be located in the /home/stack/flexpod-templates directory on the director server.

This template runs on all of the servers provisioned in the overcloud after the core Puppet configuration is done by the director. It performs the following task:

- Creates Pacemaker file system resource records for NetApp E-Series on the first controller provisioned in the overcloud (overcloud-controller-0) only. Because we have a consistent udev naming structure that is the same across all controller servers, we can use Pacemaker’s resource creation functionality to mount the raw devices presented under /dev/eserieslunX to the proper /srv/node mount location on each system.

This template is not meant to be modified.

Postdeployment Scripts (Non-Heat)

These scripts perform tasks that were not ideally suited for Heat orchestration templates and are provided as scripts that are launched after an overcloud deployment is successful.

These templates can be found in a subdirectory of the NetApp GitHub repository and are available at https://github.com/NetApp/snippets/tree/master/RedHat/osp8-liberty/tr/postdeploy-flexpod-scripts.

For the Manila-related tasks automated in these script files, these postdeployment scripts represent an effort by NetApp to enable the Manila value proposition available in Red Hat OpenStack 8 for our customers until the following pieces of upstream OpenStack code successfully merge and can land in Red Hat OpenStack:

- Enable Manila Integration
- Enable NetApp Integration with Manila
At that point, this technical report will be updated to reflect how Manila can be installed in a manner similar to that of Cinder in the Red Hat OpenStack Platform director.

The following files are included on GitHub:

**flexpodupdate-start.sh**

The flexpodupdate-start.sh file should be located in the /home/stack/postdeploy-flexpod-scripts directory on the Red Hat OpenStack Platform director server.

This script should be launched by the user after the overcloud is successfully provisioned. The resulting overcloud has symmetric key cryptography (and passwordless SSH) enabled from the Red Hat OpenStack Platform director to all of the servers. Therefore, we use the heat-admin user to copy (using SCP) all of the individual, modular script files to be launched at specific intervals to configure the overcloud for integration specific to FlexPod. The operations run in the following order:

1. flexpodupdate-allsystems.sh is launched.
2. flexpodupdate-allcontrollers.sh is launched.
3. flexpodupdate-cont0.sh is launched.

This script is not meant to be modified as reflected in this technical report.

**flexpodupdate-allsystems.sh**

The flexpodupdate-allsystems.sh file should be located in the /home/stack/postdeploy-flexpod-scripts directory on the Red Hat OpenStack Platform director server.

This script runs on all of the resulting servers provisioned in the overcloud and is subsequently copied and launched as a function of the flexpodupdate-start.sh script. It performs the following tasks:

1. Updates the openstack-manila, openstack-manila-share, and python-manila packages on each host in the overcloud.
2. Installs the python-manilaclient package, which provides CLI administration and orchestration of Manila.

This script is not meant to be modified as reflected in this technical report.

**flexpodupdate-controllers.sh**

The flexpodupdate-controllers.sh file should be located in the /home/stack/postdeploy-flexpod-scripts directory on the Red Hat OpenStack Platform director server.

**Note:** Make sure that the MANILA_DB_PASSWORD and the MANILA_USER_PASSWORD are set to the same values in the flexpodupdate-controllers.sh script.

**Note:** Make sure that the following variables are set to match the customer environment:

- NETAPP_CLUSTERADMIN_LIF. Set to the IP address or host name of the cluster admin LIF to be used by the Manila service.
- NETAPP_CLUSTERADMIN_USER. Set to the cluster admin user name.
- NETAPP_CLUSTERADMIN_PASS. Set to the cluster admin password.
- NETAPP_MANILA_SVM. Set to the SVM to be used by the Manila service.

flexpodupdate-controllers.sh runs on all of the controller servers provisioned in the overcloud and is subsequently copied and launched as a function of the flexpodupdate-start.sh script. It performs the following tasks:
1. Copies the overcloud environment variables file from the /tmp directory on the controller to the /root directory. This environment file is used to write the necessary configuration variables for Manila into the /etc/manila/manila.conf configuration file in future steps.

2. Copies the NetApp copy offload tool to the proper /usr/local/bin directory.

3. Installs the openstack-manila-ui package for the Shares tab to be accessible in Horizon.

4. Configures HAProxy to listen on port 8786 in a highly available fashion for the Manila service.

5. Writes necessary configuration options for the Manila service in the /etc/manila/manila.conf configuration file.

6. Restarts the HAProxy service to pick up the Manila-specific configuration done in step 4.

flexpodupdate-cont0.sh

The flexpodupdate-cont0.sh file should be located in the /home/stack/postdeploy-flexpod-scripts directory on the Red Hat OpenStack Platform director server.

Note: Make sure that the MANILA_DB_PASSWORD and the MANILA_USER_PASSWORD are set to the same values in the flexpodupdate-controllers.sh script.

flexpodupdate-cont0.sh runs on only the first controller server provisioned in the overcloud (overcloud-controller-0) and is subsequently copied and launched as a function of the flexpodupdate-start.sh script. It performs the following tasks:

1. Creates user, role, and service records for Manila in the OpenStack Keystone identity service for the resulting overcloud.

2. Creates a database called "manila" in Galera and sets the appropriate privileges in the database to allow the Manila user (created in step 1) the ability to read and write to this database.

3. Performs the initial synchronization of the database using the manila-manage command.


5. Creates resource records and ordering prioritization in Pacemaker to monitor and start the Manila service daemons in the following high-availability configuration. This mimics the Cinder service, to which Manila is closely related:
   a. manila-api. Active-active service
   b. manila-scheduler. Active-active service
   c. manila-share. Active-passive service

3 Solution Configuration

3.1 Physical Topology

This section describes the physical layout of the integrated reference architecture. It includes pictorial layouts and cabling diagrams for all pieces of equipment in the solution design.

Figure 12 shows a high-level diagram showing the equipment presented in this technical report.
3.2 Cabling

This section provides cabling diagrams for the following:

- NetApp FAS and E-Series storage cabling to the Cisco Nexus 9396PX switch pair
- Cisco fabric interconnect cabling from the Cisco UCS 6248UP pair to the Cisco Nexus 9396PX switch pair
- Cisco UCS server cabling from both Cisco UCS 5108 chassis to the Cisco UCS 6248UP Fabric Interconnect pair
NetApp FAS and E-Series

NetApp FAS and E-Series systems are connected to the Cisco Nexus 9396PX switch pair in a redundant fashion per platform. Figure 13 shows these components along with a numbered cabling map.
Figure 13) FAS and E-Series cabling to Cisco Nexus switch pair.
Table 8 and Table 9 list the connections from the NetApp FAS controllers to the Cisco Nexus 9396PX switch pair. This information corresponds to each connection shown in Figure 13.

### Table 8) NetApp FAS8040 cabling information.

<table>
<thead>
<tr>
<th>Local Device</th>
<th>Local Port</th>
<th>Connection</th>
<th>Remote Device</th>
<th>Remote Port</th>
<th>Cabling Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp FAS8040 Controller A</td>
<td>e0M</td>
<td>GbE</td>
<td>GbE management switch</td>
<td>Any</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>e0P</td>
<td>GbE</td>
<td>SAS shelves</td>
<td>ACP port</td>
<td></td>
</tr>
<tr>
<td></td>
<td>e0b</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX A</td>
<td>Eth1/1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>e0d</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX B</td>
<td>Eth1/1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>e0a</td>
<td>10GbE</td>
<td>NetApp FAS8040 Controller B (cluster port)</td>
<td>e0a</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>e0b</td>
<td>10GbE</td>
<td>NetApp FAS8040 Controller B (cluster port)</td>
<td>e0b</td>
<td>30</td>
</tr>
<tr>
<td>NetApp FAS8040 Controller B</td>
<td>e0M</td>
<td>GbE</td>
<td>GbE management switch</td>
<td>Any</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>e0P</td>
<td>GbE</td>
<td>SAS shelves</td>
<td>ACP port</td>
<td></td>
</tr>
<tr>
<td></td>
<td>e0b</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX A</td>
<td>Eth1/2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>e0d</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX B</td>
<td>Eth1/2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>e0a</td>
<td>10GbE</td>
<td>NetApp FAS8040 Controller A (cluster port)</td>
<td>e0a</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>e0b</td>
<td>10GbE</td>
<td>NetApp FAS8040 Controller A (cluster port)</td>
<td>e0b</td>
<td>30</td>
</tr>
</tbody>
</table>

**Note:** When the term e0M is used, the physical Ethernet port to which the table is referring is the port indicated by a wrench icon on the rear of the chassis.

### Table 9) NetApp E-5660 cabling information.

<table>
<thead>
<tr>
<th>Local Device</th>
<th>Local Port</th>
<th>Connection</th>
<th>Remote Device</th>
<th>Remote Port</th>
<th>Cabling Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp E-5660 Controller A</td>
<td>Port 1 management</td>
<td>GbE</td>
<td>GbE management switch</td>
<td>Any</td>
<td>36</td>
</tr>
<tr>
<td>HIC 1, Port 1</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX A</td>
<td>Eth1/3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>HIC 1, Port 3</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX B</td>
<td>Eth1/4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>NetApp E-5660 Controller B</td>
<td>Port 1 management</td>
<td>GbE</td>
<td>GbE management switch</td>
<td>Any</td>
<td>37</td>
</tr>
<tr>
<td>HIC 1, Port 1</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX B</td>
<td>Eth1/3</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>HIC 1, Port 3</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX A</td>
<td>Eth1/4</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
Cisco Fabric Interconnect and Cisco Nexus 9396PX

Cisco UCS 6248UP Fabric Interconnect devices are connected to the Cisco Nexus 9396PX switch pair in a redundant fashion. Figure 14 shows these components along with a numbered cabling map.

Figure 14) Fabric interconnect and Cisco Nexus cabling.

The information provided in Table 10 and Table 11 corresponds to each connection shown in Figure 14. Table 10 lists the connections from the Cisco UCS 6248UP Fabric Interconnect devices to the Cisco Nexus 9396PX switch pair. Also shown in Table 11 are the intraconnections between the Cisco Nexus 9396PX switch pair for HA through the vPC peer link feature. Uplink cabling to the rest of the data center network infrastructure is omitted because this varies depending on customer requirements.

Table 10) Cisco UCS 6248UP Fabric Interconnect cabling.

<table>
<thead>
<tr>
<th>Local Device</th>
<th>Local Port</th>
<th>Connection</th>
<th>Remote Device</th>
<th>Remote Port</th>
<th>Cabling Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco UCS 6248UP</td>
<td>Eth1/19</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX A</td>
<td>Eth1/11</td>
<td>9</td>
</tr>
<tr>
<td>Fabric A</td>
<td>Eth1/20</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX A</td>
<td>Eth1/11</td>
<td>11</td>
</tr>
<tr>
<td>Cisco UCS 6248UP</td>
<td>Eth1/19</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX A</td>
<td>Eth1/12</td>
<td>10</td>
</tr>
<tr>
<td>Fabric B</td>
<td>Eth1/20</td>
<td>10GbE</td>
<td>Cisco Nexus 9396PX B</td>
<td>Eth1/12</td>
<td>12</td>
</tr>
</tbody>
</table>
Table 11) Cisco Nexus 9396PX infraswitch cabling.

<table>
<thead>
<tr>
<th>Local Device</th>
<th>Local Port</th>
<th>Connection</th>
<th>Remote Device</th>
<th>Remote Port</th>
<th>Cabling Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Nexus 9396PX</td>
<td>mgmt0</td>
<td>1GbE</td>
<td>Cisco Nexus 9396PX</td>
<td>mgmt0</td>
<td>33</td>
</tr>
<tr>
<td>Switch A</td>
<td>Eth2/1</td>
<td>40GbE</td>
<td>Cisco Nexus 9396PX</td>
<td>Eth2/1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eth2/2</td>
<td>40GbE</td>
<td>Cisco Nexus 9396PX</td>
<td>Eth2/2</td>
<td>32</td>
</tr>
<tr>
<td>Cisco Nexus 9396PX</td>
<td>mgmt0</td>
<td>1GbE</td>
<td>Cisco Nexus 9396PX</td>
<td>mgmt0</td>
<td>33</td>
</tr>
<tr>
<td>Switch B</td>
<td>Eth2/1</td>
<td>40GbE</td>
<td>Cisco Nexus 9396PX</td>
<td>Eth2/1</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Eth2/2</td>
<td>40GbE</td>
<td>Cisco Nexus 9396PX</td>
<td>Eth2/2</td>
<td>32</td>
</tr>
</tbody>
</table>

Cisco UCS Chassis and Fabric Interconnect

Both Cisco UCS 5108 chassis in this technical report are connected to the Cisco UCS 6248UP Fabric Interconnect pair in a redundant fashion. Figure 15 shows a diagram of these components along with a numbered cabling map.
The information provided in Table 12 corresponds to each connection shown in Figure 15.

Table 12 lists the connections from both Cisco UCS 5108 Chassis 1 and Chassis 2 to the 6248UP Fabric Interconnect devices.
Table 12) Cisco UCS 5108 chassis cabling.

<table>
<thead>
<tr>
<th>Local Device</th>
<th>Local Port</th>
<th>Connection</th>
<th>Remote Device</th>
<th>Remote Port</th>
<th>Cabling Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco UCS 5108 Chassis 1</td>
<td>Fex 1 Eth1/1</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/1</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Fex 1 Eth1/2</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/2</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Fex 1 Eth1/3</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/3</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Fex 1 Eth1/4</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/4</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/1</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/1</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/2</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/2</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/3</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/3</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/4</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/4</td>
<td>20</td>
</tr>
<tr>
<td>Cisco UCS 5108 Chassis 2</td>
<td>Fex 1 Eth1/1</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/5</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Fex 1 Eth1/2</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/6</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Fex 1 Eth1/3</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/7</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Fex 1 Eth1/4</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP A</td>
<td>Eth1/8</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/1</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/5</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/2</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/6</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/3</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/7</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Fex 2 Eth1/4</td>
<td>10GbE</td>
<td>Cisco UCS 6248UP B</td>
<td>Eth1/8</td>
<td>28</td>
</tr>
</tbody>
</table>

3.3 Laboratory and Network Diagrams

This technical report validates Red Hat OpenStack Platform 8 running on a FlexPod configuration in NetApp engineering laboratories located in Research Triangle Park, North Carolina, USA. Figure 16 shows a connectivity diagram of the FlexPod configuration in our lab.

Note:  Correct cabling is vital for correct and efficient operation of the infrastructure, in both the initial deployment and the ongoing lifecycle.
Note: Unless otherwise noted, all cabling is Cisco SFP+ copper twinax cables for all flows in the data path. Standard Cat6/6e copper cabling is used for all 1GbE management traffic.

Figure 16) Connectivity diagram.
Necessary VLANs

Table 13 lists which VLANs were required to be configured throughout the infrastructure in our lab to support Red Hat OpenStack 8. A definition of what the VLAN is used for follows this in Table 14.

Table 13) Necessary VLANs.

<table>
<thead>
<tr>
<th>VLAN Name</th>
<th>Value</th>
<th>OpenStack Network Type</th>
<th>Jumbo Frames?</th>
<th>Subnet CIDR</th>
<th>Infrastructure Used By</th>
<th>OpenStack Server Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS</td>
<td>67</td>
<td>Storage</td>
<td>Yes</td>
<td>192.168.67.0/24</td>
<td>NetApp FAS8040</td>
<td>All servers</td>
</tr>
<tr>
<td>OSP-StorMgmt</td>
<td>99</td>
<td>Storage management</td>
<td>Yes</td>
<td>192.168.99.0/24</td>
<td>N/A</td>
<td>Controller</td>
</tr>
<tr>
<td>iSCSI-A</td>
<td>188</td>
<td></td>
<td>Yes</td>
<td>192.168.188.0/24</td>
<td>NetApp FAS8040 NetApp E5660</td>
<td>All servers</td>
</tr>
<tr>
<td>iSCSI-B</td>
<td>189</td>
<td></td>
<td>Yes</td>
<td>192.168.189.0/24</td>
<td>NetApp FAS8040 NetApp E5660</td>
<td>All servers</td>
</tr>
<tr>
<td>OSP-Backend</td>
<td>421</td>
<td>Internal API</td>
<td>Yes</td>
<td>192.168.42.0/24</td>
<td>n/a</td>
<td>All servers</td>
</tr>
<tr>
<td>OOB-Management</td>
<td>3267</td>
<td>External</td>
<td>No</td>
<td>172.21.11.0/24</td>
<td>NetApp FAS8040 NetApp E5660 Cisco UCS Manager Cisco Nexus 9k inbound mgmt Cisco UCS server KVM mgmt</td>
<td>All servers</td>
</tr>
<tr>
<td>Public</td>
<td>3270</td>
<td>Floating IP</td>
<td>No</td>
<td>172.21.14.0/24</td>
<td>N/A</td>
<td>Controller</td>
</tr>
<tr>
<td>Tunnel</td>
<td>3272</td>
<td>Tenant</td>
<td>No</td>
<td>172.21.16.0/24</td>
<td>N/A</td>
<td>All servers</td>
</tr>
<tr>
<td>PXE</td>
<td>3275</td>
<td>IPMI</td>
<td>No</td>
<td>172.21.19.0/24</td>
<td>N/A</td>
<td>All servers</td>
</tr>
</tbody>
</table>

VLAN Purpose

Table 14) VLAN purpose.

<table>
<thead>
<tr>
<th>VLAN Name</th>
<th>Value</th>
<th>VLAN Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS</td>
<td>67</td>
<td>VLAN for NFS traffic used by Cinder and Manila carried to the NetApp FAS8040.</td>
</tr>
<tr>
<td>OSP-StorMgmt</td>
<td>99</td>
<td>OpenStack-specific network that the Swift service uses to synchronize data objects between participating replica nodes. The proxy service acts as the intermediary interface between user requests and the underlying storage layer.</td>
</tr>
<tr>
<td>iSCSI-A</td>
<td>188</td>
<td>VLAN designated for the iSCSI-A fabric path used by servers to access their root disks hosted on the NetApp FAS8040. This network also services the controller hosts when they log in to the NetApp E5660 to read and write to the Swift ACO LUNs.</td>
</tr>
<tr>
<td>iSCSI-B</td>
<td>189</td>
<td>VLAN designated for the iSCSI-B fabric path used by servers to access their root disks hosted on the NetApp FAS8040. This network also services the controller hosts as they log in to the NetApp E5660 to read and write to the Swift ACO LUNs.</td>
</tr>
<tr>
<td>VLAN Name</td>
<td>Value</td>
<td>VLAN Purpose</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OSP-Backend</td>
<td>421</td>
<td>OpenStack-specific network used for internal API communication between OpenStack services using API communication, RPC messages, and database communication.</td>
</tr>
<tr>
<td>OOB-Management</td>
<td>3267</td>
<td>Network for out-of-band management interfaces carried down to server data NICs and is an infrastructure-related VLAN. Also houses public APIs for OpenStack services and the Horizon dashboard.</td>
</tr>
<tr>
<td>Public</td>
<td>3270</td>
<td>Also known as the “floating IP network,” this is an OpenStack-specific network and allows incoming traffic to reach instances using 1-to-1 IP address mapping between the floating IP address and the IP address actually assigned to the instance in the tenant network. Because this is a separate VLAN from the OOB-Management network, we tag the public VLAN to the controller nodes and add it through OpenStack Neutron after overcloud creation. This is demonstrated later in this technical report.</td>
</tr>
<tr>
<td>Tunnel</td>
<td>3272</td>
<td>Neutron provides each tenant with its own network using either VLAN segregation, where each tenant network is a network VLAN, or tunneling through VXLAN or GRE. Network traffic is isolated within each tenant network. Each tenant network has an IP subnet associated with it, and multiple tenant networks may use the same addresses.</td>
</tr>
<tr>
<td>PXE</td>
<td>3275</td>
<td>The Red Hat OpenStack Platform director uses this network traffic type to deploy new nodes over PXE boot and orchestrate the installation of OpenStack Platform on the overcloud bare metal servers. This network is predefined before the installation of the undercloud.</td>
</tr>
</tbody>
</table>

Figure 17 diagrams which VLANs are carried to the respective equipment composing this solution, from a device perspective.
Figure 17) Logical network segmentation diagram for compute and storage.
3.4 Cisco Nexus 9396PX Setup

Setting up the Cisco Nexus 9396 in a step-by-step manner is outside the scope of this document. However, subsequent sections list the startup configuration files for both switches that were used in this technical report.

Port-channel 48 (and subsequently vPC 48, because both nx9396-a and nx9396-b use it) is the uplink out of the NetApp lab environment and is used for outbound connectivity.

**Note:** Virtual Router Redundancy Protocol (VRRP) was enabled inside of the NetApp lab environment to provide a workaround for a known bug in iSCSI-initiator-utils in RHEL7. During bootup, the iscsistart process incorrectly uses the first NIC enumerated in the machine (vNIC-A to log in to the fabric hosted on Node-2 of the NetApp FAS8040. This causes extremely long timeouts and makes only one path available to Cisco UCS nodes postinstallation from a dm-multipath perspective.

For more details, see [https://bugzilla.redhat.com/show_bug.cgi?id=1206191](https://bugzilla.redhat.com/show_bug.cgi?id=1206191).

---

### Cisco Nexus 9396PX Switch-A

```

version 7.0(3)i2(2a)
hostname nx9396-a
vdc nx9396-a id 1
  limit-resource vlan minimum 16 maximum 4094
  limit-resource vrf minimum 2 maximum 4096
  limit-resource port-channel minimum 0 maximum 511
  limit-resource u4route-mem minimum 248 maximum 248
  limit-resource u6route-mem minimum 96 maximum 96
  limit-resource m4route-mem minimum 58 maximum 58
  limit-resource m6route-mem minimum 8 maximum 8

feature vrrp
feature vpc
feature lacp
feature vpc
feature vpc
feature lldp

mac address-table aging-time 300

no password strength-check
username admin password $1$HYdSBmnQ$ikEN6.Ncu6iWbX19/xl/a0 role network-admin
ip domain-lookup
ip name-server 10.102.76.214 10.122.76.132

copp profile strict
snmp-server user admin network-admin auth md5 0x558b2f4a2c0d13666fc15ad119a97170 priv
0x558b2f4a2c0d13666fc15ad119a97170 localizedkey
rmon event 1 log trap public description FATAL(1) owner PMON@FATAL
rmon event 2 log trap public description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap public description ERROR(3) owner PMON@ERROR
rmon event 4 log trap public description WARNING(4) owner PMON@WARNING
rmon event 5 log trap public description INFORMATION(5) owner PMON@INFO
ntp server 10.56.32.32 use-vrf default

vlan 1,67,99,188-189,421,3267-3276
vlan 67
  name NFS
vlan 99
  name OSP-StorMgmt
vlan 188
  name iSCSI-A
vlan 189
  name iSCSI-B
vlan 421
  name OSP-Backend
vlan 3267
```
name OOB-Mgmt
vlan 3270
  name Public
vlan 3272
  name Tunnel
vlan 3275
  name PXE

vrf context management
port-channel load-balance src-dst 14port
vpc domain 1
  peer-switch
  role priority 490
  peer-keepalive destination 1.1.1.2
  delay restore 150
  peer-gateway
  auto-recovery

interface Vlan1
  no ip redirects
  no ipv6 redirects

interface Vlan188
  description iSCSI-A
  no shutdown
  no ip redirects
  ip address 192.168.188.2/24
  no ipv6 redirects
  vrrp 188
    priority 200
    address 192.168.188.1
    no shutdown

interface Vlan189
  description iSCSI-B
  no shutdown
  no ip redirects
  ip address 192.168.189.2/24
  no ipv6 redirects
  vrrp 189
    priority 200
    address 192.168.189.1
    no shutdown

interface Vlan3267
  description OOB-Management
  no shutdown
  no ip redirects
  ip address 172.21.11.10/24
  no ipv6 redirects

interface port-channel1
  description fas8040-openstack-01
  switchport mode trunk
  switchport trunk allowed vlan 67,188-189,3268-3269
  spanning-tree port type edge trunk
  mtu 9216
  vpc 1

interface port-channel12
  description fas8040-openstack-02
  switchport mode trunk
  switchport trunk allowed vlan 67,188-189,3268-3269
  spanning-tree port type edge trunk
  mtu 9216
  vpc 2

interface port-channel11
  description PO11-6248-A
  switchport mode trunk
<table>
<thead>
<tr>
<th>Description</th>
<th>VLANs</th>
<th>Mode</th>
<th>MTU</th>
<th>PC Group</th>
<th>Active Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>switchport trunk allowed</td>
<td>67,99,188-189,421,3267-3276</td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface port-channel12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description PO12-6248-B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>67,99,188-189,421,3267-3276</td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface port-channel148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>description PO48-UF LINK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>3267-3276</td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface port-channel199</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description PO99-vPC Peer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>67,99,188-189,421,3267-3276</td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Ethernet1/1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>description fas8040-openstack-01</td>
<td></td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>67,188-189,3268-3269</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Ethernet1/2</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>description fas8040-openstack-02</td>
<td></td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>67,188-189,3268-3269</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Ethernet1/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description e5660-openstack-a1</td>
<td></td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport access vlan</td>
<td>188</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Ethernet1/4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description e5660-openstack-b3</td>
<td></td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport access vlan</td>
<td>189</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Ethernet1/11</td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>description 6248-A:1/19</td>
<td></td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>67,99,188-189,421,3267-3276</td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Ethernet1/12</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>description 6248-B:1/19</td>
<td></td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>67,99,188-189,421,3267-3276</td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spanning-tree port type edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Ethernet1/48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>description E1/48-UF LINK</td>
<td></td>
<td></td>
<td>9216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>switchport trunk allowed</td>
<td>3267-3276</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
spanning-tree port type normal
channel-group 48 mode active

interface Ethernet2/1
  description nx9396-b
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  channel-group 99 mode active

interface Ethernet2/2
  description nx9396-b
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  channel-group 99 mode active

interface mgmt0
  description VPC Peer Keep Alive
  vrf member management
  ip address 1.1.1.1/24
  clock timezone EST -5 0
  clock summer-time EDT 2 Sun Mar 02:00 1 Sun Nov 02:00 60
  line console
  line vty
  session-limit 16
  boot nxos bootflash:/nxos.7.0.3.I2.2a.bin
  ip route 0.0.0.0/0 172.21.11.254
  no xml server exec-mode

Cisco Nexus 9396PX Switch-B

version 7.0(3)I2(2a)
hostname nx9396-b
vdc nx9396-b id 1
  limit-resource vlan minimum 16 maximum 4094
  limit-resource vrf minimum 2 maximum 4096
  limit-resource port-channel minimum 0 maximum 511
  limit-resource u4route-mem minimum 248 maximum 248
  limit-resource u6route-mem minimum 96 maximum 96
  limit-resource m4route-mem minimum 58 maximum 58
  limit-resource m6route-mem minimum 8 maximum 8

feature vrrp
cfs eth distribute
feature interface-vlan
feature lacp
feature vpc
feature lldp
mac address-table aging-time 300
no password strength-check
username admin password 5 $1$V7rzlFrS$xOyz.fm/LMrE3uKX4.InL1 role network-admin
ip domain-lookup
ip name-server 10.102.76.214 10.122.76.132
copp profile strict
snmp-server user admin network-admin auth md5 0x71459e800f250977f012058914f36e35 localizedkey
 0x71459e800f250977f012058914f36e35 priv
rmon event 1 log trap public description FATAL(1) owner PMON@FATAL
rmon event 2 log trap public description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap public description ERROR(3) owner PMON@ERROR
rmon event 4 log trap public description WARNING(4) owner PMON@WARNING
rmon event 5 log trap public description INFORMATION(5) owner PMON@INFO
ntp server 10.56.32.32 use-vrf default

vlan 1,67,99,188-189,421,3267-3276
vlan 67
  name NFS
vlan 99
  name OSP-StorMgmt
vlan 188
name iSCSI-A
  vlan 189
name iSCSI-B
  vlan 421
name OSP-Backend
  vlan 3267
name OOB-Mgmt
  vlan 3270
name Public
  vlan 3272
name Tunnel
  vlan 3275
name PxE

vrf context management
port-channel load-balance src-dst l4port
vpc domain 1
  role priority 500
  peer-keepalive destination 1.1.1.1
  delay restore 150
  peer-gateway
  auto-recovery

interface Vlan1
  no ip redirects
  no ipv6 redirects

interface Vlan188
  description iSCSI-A
  no shutdown
  no ip redirects
  ip address 192.168.188.3/24
  no ipv6 redirects
vrrp 188
  priority 150
  address 192.168.188.1
  no shutdown

interface Vlan189
  description iSCSI-B
  no shutdown
  no ip redirects
  ip address 192.168.189.3/24
  no ipv6 redirects
vrrp 189
  priority 150
  address 192.168.189.1
  no shutdown

interface Vlan3267
  description OOB-Management
  no shutdown
  no ip redirects
  ip address 172.21.11.11/24
  no ipv6 redirects

interface port-channel1
  description fas8040-openstack-01
  switchport mode trunk
  switchport trunk allowed vlan 67,188-189,3268-3269
  spanning-tree port type edge trunk
  mtu 9216
  vpc 1

interface port-channel2
  description fas8040-openstack-02
  switchport mode trunk
  switchport trunk allowed vlan 67,188-189,3268-3269
  spanning-tree port type edge trunk
  mtu 9216
vpc 2

interface port-channel11
  description PO11-6248-A
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  spanning-tree port type edge trunk
  mtu 9216
vpc 11

interface port-channel12
  description PO12-6248-B
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  spanning-tree port type edge trunk
  mtu 9216
vpc 12

interface port-channel48
  description PO48-UPLINK
  switchport mode trunk
  switchport trunk allowed vlan 3267-3276
  spanning-tree port type normal
vpc 48

interface port-channel199
  description PO99-vPC Peer
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  spanning-tree port type network
  vpc peer-link

interface Ethernet1/1
  description fas8040-openstack-01
  switchport mode trunk
  switchport trunk allowed vlan 67,188-189,3268-3269
  mtu 9216
  channel-group 1 mode active

interface Ethernet1/2
  description fas8040-openstack-02
  switchport mode trunk
  switchport trunk allowed vlan 67,188-189,3268-3269
  mtu 9216
  channel-group 2 mode active

interface Ethernet1/3
  description e5660-openstack-b1
  switchport access vlan 188
  spanning-tree port type edge
  mtu 9216

interface Ethernet1/4
  description e5660-openstack-a3
  switchport access vlan 189
  spanning-tree port type edge
  mtu 9216

interface Ethernet1/11
  description 6248-A:1/20
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  mtu 9216
  channel-group 11 mode active

interface Ethernet1/12
  description 6248-B:1/20
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  mtu 9216
  channel-group 12 mode active
interface Ethernet1/48
  description E1/48-UPLINK
  switchport mode trunk
  switchport trunk allowed vlan 3267-3276
  spanning-tree port type normal
  channel-group 48 mode active

interface Ethernet2/1
  description nx9396-a
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  channel-group 99 mode active

interface Ethernet2/2
  description nx9396-a
  switchport mode trunk
  switchport trunk allowed vlan 67,99,188-189,421,3267-3276
  channel-group 99 mode active

interface mgmt0
  description vPC Peer Keep Alive
  vrf member management
  ip address 1.1.1.2/24
  clock timezone EST -5 0
  clock summer-time EDT 2 Sun Mar 02:00 1 Sun Nov 02:00 60
  line console
  line vty
    session-limit 16
  boot nxos bootflash:/nxos.7.0.3.I2.2a.bin
  ip route 0.0.0.0/0 172.21.11.254
  no xml server exec-mode

3.5 NetApp FAS8040 Setup

For instructions on the physical installation of FAS8000 controllers, follow the procedures in the FAS8000 series documentation on the NetApp Support site. When planning the physical location of a storage system, refer to the following sections in the Site Requirements Guide:

- Site Preparation
- System Connectivity Requirements
- Circuit Breaker, Power Outlet Balancing, System Cabinet Power Cord Plugs, and Console Pinout Requirements
- 8000 Series Systems

Disk Shelves

NetApp storage systems support a wide variety of disk shelves and disk drives. Visit the NetApp Support site to view a complete list of supported disk shelves. This solution is built on a single DS2246 disk shelf with 24 900GB SAS disks. These disks provide an ample amount of storage at a moderate price point and are recommended for an OpenStack deployment. When using SAS disk shelves with NetApp storage controllers, refer to the SAS Disk Shelves Universal SAS and ACP Cabling Guide for information about cabling guidelines.

Clustered Data ONTAP 8.3.2

The procedure described in this section and the following subsections assumes that the storage system has been installed and cabled and is ready for setup. For detailed information about storage system installation, see the resources listed in the previous section.
Complete the Configuration Worksheet

Before performing the following procedures, review the configuration worksheets in the Clustered Data ONTAP 8.3 Software Setup Guide to learn about the information required to configure clustered Data ONTAP. Table 15 lists the information you need to configure two clustered Data ONTAP nodes. You should customize the cluster detail values with the information that is applicable to your deployment.

Table 15) Cluster details for the clustered Data ONTAP software configuration.

<table>
<thead>
<tr>
<th>Cluster Detail</th>
<th>Cluster Detail Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster name</td>
<td>&lt;&lt;var_clustername&gt;&gt;</td>
</tr>
<tr>
<td>Clustered Data ONTAP base license</td>
<td>&lt;&lt;var_cluster_base_license_key&gt;&gt;</td>
</tr>
<tr>
<td>Clustered Data ONTAP NFS license</td>
<td>&lt;&lt;var_nfs_license&gt;&gt;</td>
</tr>
<tr>
<td>Clustered Data ONTAP iSCSI license</td>
<td>&lt;&lt;var_iscsi_license&gt;&gt;</td>
</tr>
<tr>
<td>Clustered Data ONTAP FlexClone license</td>
<td>&lt;&lt;var_flexclone_license&gt;&gt;</td>
</tr>
<tr>
<td>Cluster management IP address</td>
<td>&lt;&lt;var_clustermgmt_ip&gt;&gt;</td>
</tr>
<tr>
<td>Cluster management netmask</td>
<td>&lt;&lt;var_clustermgmt_mask&gt;&gt;</td>
</tr>
<tr>
<td>Cluster management port</td>
<td>&lt;&lt;var_clustermgmt_port&gt;&gt;</td>
</tr>
<tr>
<td>Cluster management gateway</td>
<td>&lt;&lt;var_clustermgmt_gateway&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 01 IP address</td>
<td>&lt;&lt;var_node01_mgmt_ip&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 01 netmask</td>
<td>&lt;&lt;var_node01_mgmt_mask&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 01 gateway</td>
<td>&lt;&lt;var_node01_mgmt_gateway&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 01 service processor IP address</td>
<td>&lt;&lt;var_node01_sp_ip&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 01 service processor netmask</td>
<td>&lt;&lt;var_node01_sp_netmask&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 01 service processor gateway</td>
<td>&lt;&lt;var_node01_sp_gateway&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 IP address</td>
<td>&lt;&lt;var_node02_mgmt_ip&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 netmask</td>
<td>&lt;&lt;var_node02_mgmt_mask&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 gateway</td>
<td>&lt;&lt;var_node02_mgmt_gateway&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 service processor IP address</td>
<td>&lt;&lt;var_node02_sp_ip&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 service processor netmask</td>
<td>&lt;&lt;var_node02_sp_netmask&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 service processor gateway</td>
<td>&lt;&lt;var_node02_sp_gateway&gt;&gt;</td>
</tr>
</tbody>
</table>

Install Clustered Data ONTAP 8.3.2

Perform the following procedure on both of the storage nodes if the running version of Data ONTAP is lower than 8.3.2. If you already have Data ONTAP version 8.3.2 installed on your storage system, skip to the section “Create Cluster on Node 01.”

1. Connect to the storage system console port. You should see a Loader prompt. However, if the storage system is in a reboot loop, press Ctrl-C to exit the autoboot loop when you see this message:
Starting AUTOBOOT press Ctrl-C to abort

2. Set boot monitor defaults.

LOADER>set-defaults

3. Set the variable to boot clustered Data ONTAP.

LOADER>setenv bootarg.init.boot_clustered true

4. Allow the system to boot up.

LOADER>autoboot

5. Press Ctrl-C when the Press Ctrl-C for Boot Menu message appears.

   **Note:** If Data ONTAP 8.3.2 is not the version of software being booted, proceed with the following steps to install new software. If Data ONTAP 8.3.2 is the version being booted, then select option 8 and yes to reboot the node and continue to the section “Create Cluster on Node 01.”

6. To install new software, select option 7.

   Please choose one of the following:

   (1) Normal Boot.
   (2) Boot without /etc/rc.
   (3) Change password.
   (4) Clean configuration and initialize all disks.
   (5) Maintenance mode boot.
   (6) Update flash from backup config.
   (7) Install new software first.
   (8) Reboot node.
   Selection (1-8)? 7

7. Enter y to perform a nondisruptive upgrade.

   This procedure is not supported for Non-Disruptive Upgrade on an HA pair. The software will be installed to the alternate image, from which the node is not currently running. Do you want to continue? {y|n} y

8. Select e0M for the network port you want to use for the download.

   Select the network port you want to use for the download (for example, ‘e0a’) [e0M] e0M

9. Enter y to reboot now.

   The node needs to reboot for this setting to take effect. Reboot now? {y|n} (selecting yes will return you automatically to this install wizard) y

10. Enter the IP address, netmask, and default gateway for e0M in their respective places. The IP for node 01 is shown in the following commands. Substitute the node 02 IP address as needed.

    Enter the IP address for port e0M: <<storage_node1_mgmt_ip>>
    Enter the netmask for port e0M: <<node_mgmt_mask>>
    Enter IP address of default gateway: <<node_mgmt_gateway>>

11. Enter the URL for the location of the software.

    **Note:** This web server must be reachable from the storage controller.

    What is the URL for the package? <<url_boot Software>>

12. Press Enter for the user name, indicating no user name.

    What is the user name on “xxx.xxx.xxx.xxx”, if any? Enter

13. Enter y to set the newly installed software as the default to be used for subsequent reboots.

    Do you want to set the newly installed software as the default to be used for subsequent reboots? {y|n} y
14. Enter `y` to reboot the node.

The node must be rebooted to start using the newly installed software. Do you want to reboot now? [y|n] y

**Note:** When installing new software, the system might perform firmware upgrades to the BIOS and adapter cards, causing reboots and possible stops at the LOADER prompt. If these actions occur, the system might deviate from this procedure.

15. Press Ctrl-C when you see Press Ctrl-C for Boot Menu.

16. Select option 4 for a clean configuration and to initialize all disks.

Please choose one of the following:

(1) Normal Boot.
(2) Boot without /etc/rc.
(3) Change password.
(4) Clean configuration and initialize all disks.
(5) Maintenance mode boot.
(6) Update flash from backup config.
(7) Install new software first.
(8) Reboot node.

Selection (1-8)? 4

17. Enter `yes` to zero disks, reset config, and install a new file system.

Zero disks, reset config and install a new file system?: yes

18. Enter `yes` to erase all of the data on the disks.

This will erase all the data on the disks, are you sure?: yes

**Note:** The initialization and creation of the root volume can take up to eight hours to complete, depending on the number and type of disks attached. After initialization is complete, the storage system reboots. You can continue to configure node 01 while the disks for node 02 are zeroing and vice versa.

### Create Cluster on Node 01

In clustered Data ONTAP, the first node in a cluster performs the `cluster create` operation. All other nodes perform a `cluster join` operation. The first node in the cluster is considered node 01. After all of the disks have been zeroed out for the first node, you can see the prompt as follows. Use the values from Table 15 to complete the configuration of the cluster and each node.

To create a cluster on node 01, complete the following steps:

1. Connect to the storage system console port. The console settings are:
   - Baud rate: 9600
   - Data bits: 8
   - Parity: none
   - Stop bit: 1
   - Flow control: none

2. The Cluster Setup wizard starts on the console.

Welcome to the cluster setup wizard.

You can enter the following commands at any time:
- "help" or "?" - if you want to have a question clarified,
- "back" - if you want to change previously answered questions, and
- "exit" or "quit" - if you want to quit the cluster setup wizard.

Any changes you made before quitting will be saved.

You can return to cluster setup at any time by typing "cluster setup". To accept a default or omit a question, do not enter a value.
Do you want to create a new cluster or join an existing cluster? {create, join}:

Note: If a login prompt appears instead of the Cluster Setup wizard, you must start the wizard by logging in with the factory default settings and then run the cluster setup command.

3. Run the following command to create a new cluster:

create

4. Enter no for the single-node cluster option.

Do you intend for this node to be used as a single node cluster? {yes, no} [no]: no

5. Enter no for the option to use network switches for the cluster network.

Will the cluster network be configured to use network switches? [yes]: no

6. Activate HA and set storage failover.

Non-HA mode, Reboot node to activate HA

Do you want to reboot now to set storage failover (SFO) to HA mode? [yes, no] [yes]: Enter

7. After the reboot, enter admin in the login prompt.

admin

8. If the Cluster Setup wizard prompt is displayed again, repeat steps 3 and 4.

9. The system defaults are displayed. Enter no for the option to use the system defaults. Follow these prompts to configure the cluster ports:

Existing cluster interface configuration found:

<table>
<thead>
<tr>
<th>Port</th>
<th>MTU</th>
<th>IP</th>
<th>Netmask</th>
</tr>
</thead>
<tbody>
<tr>
<td>e0a</td>
<td>9000</td>
<td>169.254.204.185</td>
<td>255.255.0.0</td>
</tr>
<tr>
<td>e0b</td>
<td>9000</td>
<td>169.254.240.144</td>
<td>255.255.0.0</td>
</tr>
<tr>
<td>e0c</td>
<td>9000</td>
<td>169.254.49.216</td>
<td>255.255.0.0</td>
</tr>
<tr>
<td>e0d</td>
<td>9000</td>
<td>169.254.241.21</td>
<td>255.255.0.0</td>
</tr>
</tbody>
</table>

Do you want to use this configuration? {yes, no} [yes]: no

System Defaults:

Private cluster network ports {e0a,e0c}. Cluster port MTU values will be set to 9000. Cluster interface IP addresses will be automatically generated.

Do you want to use these defaults? {yes, no} [yes]: no

Step 1 of 5: Create a Cluster

You can type "back", "exit", or "help" at any question.

List the private cluster network ports {e0a,e0b,e0c,e0d}: e0a,e0c
Enter the cluster ports' MTU size [9000]: Enter
Enter the cluster network netmask [255.255.0.0]: Enter

Generating a default IP address. This can take several minutes...
Enter the cluster interface IP address for port e0a [169.254.73.54]: Enter

Generating a default IP address. This can take several minutes...
Enter the cluster interface IP address for port e0c [169.254.64.204]: Enter

10. Use the information in Table 15 to create a cluster.

Enter the cluster name: <<var_clustername>>
Enter the cluster base license key: <<var_cluster_base_license_key>>

Creating cluster <<var_clustername>>
Enter an additional license key []: <<var_nfs_license>>
Enter an additional license key []: <<var_iscsi_license>>
Enter an additional license key []: <<var_flexclone_license>>

**Note:** The cluster-create process can take a minute or two.

**Note:** Although not strictly required for this validated architecture, NetApp recommends that you also install license keys for NetApp SnapRestore® and the SnapManager suite. These license keys can be added now or at a later time using the CLI or GUI.

Enter the cluster administrators (username "admin") password: <<var_password>>
Retype the password: <<var_password>>
Enter the cluster management interface port [e0M]: e0M
Enter the cluster management interface IP address: <<var_clustermgmt_ip>>
Enter the cluster management interface netmask: <<var_clustermgmt_mask>>
Enter the cluster management interface default gateway: <<var_clustermgmt_gateway>>

11. Enter the DNS domain name.
Enter the DNS domain names: <<var_dns_domain_name>>
Enter the name server IP addresses: <<var_nameserver_ip>>

**Note:** If you have more than one DNS server on your network, separate each one with a comma.

12. Set up the node.

Where is the controller located []: <<var_node_location>>
Enter the node management interface port [e0M]: e0M
Enter the node management interface IP address: <<var_node01_mgmt_ip>>
Enter the node management interface netmask: <<var_node01_mgmt_mask>>
Enter the node management interface default gateway: <<var_node01_mgmt_gateway>>

**Note:** The node management interfaces and the cluster management interface should be in different subnets. The node management interfaces can reside on the out-of-band management network, and the cluster management interface can be on the in-band management network.

13. Enter no for the option to enable IPV4 DHCP on the service processor.

Enable IPV4 DHCP on the service processor interface [yes]: no

14. Set up the service processor.

Enter the service processor interface IP address: <<var_node01_sp_ip>>
Enter the service processor interface netmask: <<var_node01_sp_netmask>>
Enter the service processor interface default gateway: <<var_node01_sp_gateway>>

15. Press Enter to accept the NetApp AutoSupport message.

16. Log in to the cluster interface with the administrator user ID and <<var_password>> as the password.

**Join Node 02 to Cluster**

The first node in the cluster performs the cluster-create operation. All other nodes perform a cluster-join operation. The first node in the cluster is considered node 01, and the node joining the cluster in this example is node 02. Table 16 lists the cluster network information required for joining node 02 to the existing cluster. You should customize the cluster detail values with the information that is applicable to your deployment.
Table 16) Cluster details for the cluster-join operation.

<table>
<thead>
<tr>
<th>Cluster Detail</th>
<th>Cluster Detail Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster node 02 IP address</td>
<td>&lt;&lt;var_node02_mgmt_ip&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 netmask</td>
<td>&lt;&lt;var_node02_mgmt_mask&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 gateway</td>
<td>&lt;&lt;var_node02_mgmt_gateway&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 service processor IP address</td>
<td>&lt;&lt;var_node02_sp_ip&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 service processor netmask</td>
<td>&lt;&lt;var_node02_sp_netmask&gt;&gt;</td>
</tr>
<tr>
<td>Cluster node 02 service processor gateway</td>
<td>&lt;&lt;var_node02_sp_gateway&gt;&gt;</td>
</tr>
</tbody>
</table>

To join node 02 to the existing cluster, complete the following steps:

1. At the login prompt, enter `admin`.

```
admin
```

2. The Cluster Setup wizard starts on the console.

```
Welcome to the cluster setup wizard.
You can enter the following commands at any time:
"help" or "?" - if you want to have a question clarified,
"back" - if you want to change previously answered questions, and
"exit" or "quit" - if you want to quit the cluster setup wizard.
Any changes you made before quitting will be saved.
You can return to cluster setup at any time by typing "cluster setup".
To accept a default or omit a question, do not enter a value.

Do you want to create a new cluster or join an existing cluster? {join}:

**Note:** If a login prompt is displayed instead of the Cluster Setup wizard, you must start the wizard by logging in with the factory default settings and then running the cluster-setup command.

3. Run the following command to join a cluster:

```
join
```

4. Activate HA and set storage failover.

```
Non-HA mode, Reboot node to activate HA
Warning: Ensure that the HA partner has started disk initialization before rebootsing this node to enable HA.
Do you want to reboot now to set storage failover (SFO) to HA mode? {yes, no} [yes]: Enter
```

5. After the reboot, continue the cluster-join process.

6. Data ONTAP detects the existing cluster and agrees to join the same cluster. Follow these prompts to join the cluster:

```
Existing cluster interface configuration found:
Port  MTU  IP              Netmask
e0a  9000  169.254.50.100  255.255.0.0
e0b  9000  169.254.74.132  255.255.0.0
e0c  9000  169.254.147.156 255.255.0.0
e0d  9000  169.254.78.241  255.255.0.0
Do you want to use this configuration? {yes, no} [yes]: no
```
System Defaults:
Private cluster network ports [e0a, e0b, e0c, e0d].
Cluster port MTU values will be set to 9000.
Cluster interface IP addresses will be automatically generated.

Do you want to use these defaults? {yes, no} [yes]: no

Step 1 of 3: Join an Existing Cluster
You can type "back", "exit", or "help" at any question.

List the private cluster network ports [e0a, e0b, e0c, e0d]: e0a, e0c
Enter the cluster ports' MTU size [9000]: Enter
Enter the cluster network netmask [255.255.0.0]: Enter
Generating a default IP address. This can take several minutes...
Generating a default IP address. This can take several minutes...

Enter the name of the cluster you would like to join [<<var_clustername>>]: Enter

Note: The node should find the cluster name.

Note: The cluster-join process can take a minute or two.

8. Set up the node.

Enter the node management interface port [e0M]: e0M
Enter the node management interface IP address: <<var_node02_mgmt_ip>>
Enter the node management interface netmask: <<var_node02_mgmt_mask>>
Enter the node management interface default gateway: <<var_node02_mgmt_gateway>>

Note: The node management interfaces and the cluster management interface should be in different subnets. The node management interfaces can reside on the out-of-band management network, and the cluster management interface can be on the in-band management network.

9. Enter no for the option to enable IPV4 DHCP on the service processor.

Enable IPV4 DHCP on the service processor interface [yes]: no

10. Set up the service processor.

Enter the service processor interface IP address: <<var_node01_sp_ip>>
Enter the service processor interface netmask: <<var_node01_sp_netmask>>
Enter the service processor interface default gateway: <<var_node01_sp_gateway>>

11. Press Enter to accept the AutoSupport message.

12. Log in to the cluster interface with the admin user ID and <<var_password>> as the password.

Configure Initial Cluster Settings
To log in to the cluster, complete the following steps:
1. Open an SSH connection to the cluster IP address or to the host name.
2. Log in as the admin user with the password that you entered earlier.

Assign Disks for Optimal Performance
To achieve optimal performance with SAS drives, the disks in each chassis should be split between the controllers, as opposed to the default allocation method of assigning all disks in a shelf to a single controller. In this solution, assign 12 disks to each controller.
To assign the disks as required, complete the following steps:

1. Verify the current disk allocation.

   ```
   disk show
   ```

2. Assign disks to the appropriate controller. This reference architecture allocates half of the disks to each controller. However, workload design could dictate different percentages.

   ```
   disk assign -n <<#_of_disks>> -owner <<var_node01>> [-force]
   disk assign -n <<#_of_disks>> -owner <<var_node02>> [-force]
   ```

   **Note:** The `-force` option might be required if the disks are already assigned to another node. Verify that the disk is not a member of an existing aggregate before changing ownership.

**Zero All Spare Disks**

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

```
 disk zerospares
```

**Create Aggregates**

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

This solution uses one aggregate on each controller, with eight drives per aggregate. To create the aggregates required for this solution, complete the following steps:

1. Run the following commands:

   ```
   agg create -aggregate aggr01_node01 -node <<var_node01>> -diskcount 8
   agg create -aggregate aggr01_node02 -node <<var_node02>> -diskcount 8
   ```

   **Note:** Retain at least one disk (select the largest disk) in the configuration as a spare. A best practice is to have at least one spare for each disk type and size per controller.

   **Note:** The aggregate cannot be created until disk zeroing completes. Run the `aggr show` command to display the aggregate creation status. Do not proceed until both `aggr01_node01` and `aggr01_node02` are online.

2. Disable Snapshot copies for the two data aggregates that you created in step 1.

   ```
   system node run -node <<var_node01>> agg options aggr01_node01 nosnap on
   system node run -node <<var_node02>> agg options aggr01_node02 nosnap on
   ```

3. Delete any existing Snapshot copies for the two data aggregates.

   ```
   system node run -node <<var_node01>> snap delete -A -a -f aggr01_node01
   system node run -node <<var_node02>> snap delete -A -a -f aggr01_node02
   ```

4. Rename the root aggregate on node 01 to match the naming convention for this aggregate on node 02.

   ```
   aggr rename -aggregate aggr0 -newname <<var_node01_rootaggrname>>
   ```

**Verify Storage Failover**

To confirm that storage failover is enabled, complete the following steps for a failover pair:

1. Verify the status of storage failover.

   ```
   storage failover show
   ```
2. Both nodes, <<var_node01>> and <<var_node02>>, must be capable of performing a takeover. If the nodes are capable of performing a takeover, go to step 4.

3. Enable failover on one of the two nodes.

```
storage failover modify -node <<var_node01>> -enabled true
```

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

4. Verify the HA status for the two-node cluster.

```
cluster ha show
```

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

5. If HA is configured, go to step 7.

6. Enable the HA mode only for the two-node cluster.

```
cluster ha modify -configured true
```

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

7. Verify that the hardware-assisted failover feature is correctly configured and, if needed, modify the partner IP address.

```
storage failover hwassist show
storage failover modify -hwassist-partner-ip <<var_node02_mgmt_ip>> -node <<var_node01>>
storage failover modify -hwassist-partner-ip <<var_node01_mgmt_ip>> -node <<var_node02>>
```

## Set Onboard UTA2 Ports Personality

To set the onboard UTA2 ports personality, complete the following steps:

1. Run the ucadmin show command to verify the current mode and current type of the ports.

```
FLEXPOD-OPS-CLUSTER::>
ucadmin show
```

<table>
<thead>
<tr>
<th>Node</th>
<th>Adapter</th>
<th>Current Mode</th>
<th>Current Type</th>
<th>Pending Mode</th>
<th>Pending Type</th>
<th>Admin Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLEXPOD-OPS-CLUSTER-01</td>
<td>0e</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
<tr>
<td>FLEXPOD-OPS-CLUSTER-01</td>
<td>0f</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
<tr>
<td>FLEXPOD-OPS-CLUSTER-01</td>
<td>0g</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
<tr>
<td>FLEXPOD-OPS-CLUSTER-01</td>
<td>0h</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
<tr>
<td>FLEXPOD-OPS-CLUSTER-02</td>
<td>0e</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
<tr>
<td>FLEXPOD-OPS-CLUSTER-02</td>
<td>0f</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
<tr>
<td>FLEXPOD-OPS-CLUSTER-02</td>
<td>0g</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
<tr>
<td>FLEXPOD-OPS-CLUSTER-02</td>
<td>0h</td>
<td>fc</td>
<td>target</td>
<td>-</td>
<td>-</td>
<td>online</td>
</tr>
</tbody>
</table>

8 entries were displayed.

2. Verify that the current mode of the ports in use is cna and the current type is target. If they are, skip to the next section, “Disable Flow Control on 10GbE and UTA2 Ports.” If not, down the ports:

```
fcp adapter modify -node <<var_node01>> -adapter * -state down
fcp adapter modify -node <<var_node02>> -adapter * -state down
```

3. Change the port personality to CNA by running the following commands:
ucadmin modify -node <<var_node01>> -adapter * -mode cna -type target
ucadmin modify -node <<var_node02>> -adapter * -mode cna -type target

**Note:** The ports must be offline to run this command.

4. Run the `ucadmin show` command to verify that the adapters are pending a change in mode.

```plaintext
FLEXPOD-OPS-CLUSTER::> ucadmin show

Node          Adapter  Current Mode  Current Type  Pending Mode  Pending Type  Admin Status
----------  -----------  ------------  ----------  ------------  ----------  ------------
FLEXPOD-OPS-CLUSTER-01 0e  fc  target  cna  -  offline
FLEXPOD-OPS-CLUSTER-01 0f  fc  target  cna  -  offline
FLEXPOD-OPS-CLUSTER-01 0g  fc  target  cna  -  offline
FLEXPOD-OPS-CLUSTER-01 0h  fc  target  cna  -  offline
FLEXPOD-OPS-CLUSTER-02 0e  fc  target  cna  -  offline
FLEXPOD-OPS-CLUSTER-02 0f  fc  target  cna  -  offline
FLEXPOD-OPS-CLUSTER-02 0g  fc  target  cna  -  offline
FLEXPOD-OPS-CLUSTER-02 0h  fc  target  cna  -  offline
8 entries were displayed.
```

5. Set the port status to up for all of the system adapters.

```plaintext
fcp adapter modify -node <<var_node01>> -adapter * -state up
fcp adapter modify -node <<var_node02>> -adapter * -state up
```

6. Run `system node reboot` to pick up the changes.

```plaintext
system node reboot -node <<var_node01>>
system node reboot -node <<var_node02>>
```

**Disable Flow Control on 10GbE and UTA2 Ports**

A NetApp best practice is to disable flow control on all the 10GbE and UTA2 ports that are connected to external devices. To disable flow control, run the following command:

```plaintext
network port modify -node * -port e0a..e0h -flowcontrol-admin none
```

Warning: Changing the network port settings will cause a several second interruption in carrier.

Do you want to continue? [y|n]: y

**Note:** The `-node` and `-port` parameters in this example take advantage of the range operator available in the clustered Data ONTAP shell. For more information, refer to the section "Methods of Using Query Operators" in the Clustered Data ONTAP 8.3 System Administration Guide for Cluster Administrators.

**Create LACP Interface Groups**

Clustered Data ONTAP 8.3.2 includes support for setting up broadcast domains on a group of network ports that belong to the same layer 2 network. A common application for broadcast domains is when a cloud administrator wants to reserve specific ports for use by a certain client or a group of clients.

**Note:** More information about broadcast domains can be found in the Clustered Data ONTAP 8.3 Network Management Guide.

This type of interface group (ifgrp) requires two or more Ethernet interfaces and a network switch pair that supports the Link Aggregation Control Protocol (LACP). Therefore, confirm that the switches are configured properly.
To create interface groups, complete the following steps:

1. Create a new broadcast domain, which is used to conveniently group the data serving ports to use jumbo frames in the next step.

```bash
network port broadcast-domain create -mtu 9000 -broadcast-domain Jumbo
```

2. Remove the chosen ports e0b and e0d from the default broadcast domain.

```bash
network port broadcast-domain remove-ports -ports <<var_node01>>:e0b,<<var_node01>>:e0d,<<var_node02>>:e0b,<<var_node02>>:e0d -broadcast-domain Default
```

3. Run the following commands to add ports to the previously created interface group (ifgrp) and add the interface groups to the Jumbo broadcast domain:

```bash
network port ifgrp create -node <<var_node01>> -ifgrp a0a -distr-func port -mode multimode_lacp
network port ifgrp add-port -node <<var_node01>> -ifgrp a0a -port e0b
network port ifgrp add-port -node <<var_node01>> -ifgrp a0a -port e0d
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node01>>:a0a

network port ifgrp create -node <<var_node02>> -ifgrp a0a -distr-func port -mode multimode_lacp
network port ifgrp add-port -node <<var_node02>> -ifgrp a0a -port e0b
network port ifgrp add-port -node <<var_node02>> -ifgrp a0a -port e0d
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node02>>:a0a
```

**Note:** The interface group name must follow the standard naming convention of `<number><letter>`, where `<number>` is an integer in the range of 0 to 999 without leading zeros, and `<letter>` is a lowercase letter.

**Note:** Modifications to an interface group cause the underlying physical ports to inherit the same configuration. If the ports are later removed from the interface group, they retain these same settings. However, the inverse is not true; modifying the individual ports does not modify the interface group of which the ports are a member.

**Note:** After the interface group is added to the broadcast domain, the MTU is set to 9,000 for the group and the individual interfaces. All new VLAN interfaces created on that interface group also have an MTU of 9,000 bytes after they are added to the broadcast domain.

### Create VLANs

To create a VLAN for the NFS traffic on both nodes, as well as the VLANs necessary for facilitating Cisco UCS compute-node stateless booting through the iSCSI protocol (fabric-a and fabric-b), complete the following steps:

1. Run the following commands:

```bash
network port vlan create -node <<var_node01>> -vlan-name a0a-<<var_NFS_vlan_id>>
network port vlan create -node <<var_node02>> -vlan-name a0a-<<var_NFS_vlan_id>>
network port vlan create -node <<var_node01>> -vlan-name a0a-<<var_iSCSIA_vlan_id>>
network port vlan create -node <<var_node02>> -vlan-name a0a-<<var_iSCSIA_vlan_id>>
```

2. Add the newly created VLANs to the jumbo broadcast domain.

```bash
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node01>>:a0a-<<var_iSCSIA_vlan_id>>,<<var_node02>>:a0a-<<var_iSCSIA_vlan_id>>
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node01>>:a0a-<<var_NFS_vlan_id>>,<<var_node02>>:a0a-<<var_NFS_vlan_id>>
```

### Enable Cisco Discovery Protocol

To enable the Cisco Discovery Protocol (CDP) on the NetApp storage controllers, run the following command:

```bash
```

©2016 NetApp, Inc. All rights reserved.
**Note:** To be effective, CDP must also be enabled on directly connected networking equipment such as switches and routers.

```
system node run -node * options cdpd.enable on
```

**Note:** The message displayed after running this command can be safely ignored.

```
You are changing option cdpd.enable, which applies to both members of the HA configuration in takeover mode. This value must be the same on both HA members to ensure correct takeover and giveback operation.
```

**Set Auto-Revert on Cluster Management Interface**

To set the auto-revert parameter on the cluster management interface, run the following command:

```
network interface modify -va server <<var_clustername>> -lif cluster_mgmt -auto-revert true
```

**Create Failover Group for Cluster Management**

To create a failover group for the cluster management port, run the following commands:

```
network interface failover-groups create -failover-group fg-cluster-mgmt -targets <<var_node01>>:e0M,<<var_node02>>:e0M -va server <<var_clustername>>
```

**Assign Cluster Management Failover Group to Cluster Management LIF**

To assign the cluster management failover group to the cluster management logical interface (LIF), run the following command:

```
network interface modify -va server <<var_clustername>> -lif cluster_mgmt -failover-group fg-cluster-mgmt
```

**Configure NTP**

To configure time synchronization on the cluster, complete the following steps:

1. **Set the time zone.**

   ```
timezone <<var_timezone>>
   
   **Note:** For example, in the eastern United States, the time zone is America/New_York.
   ```

2. **Set the date.**

   ```
date <ccyymmddhhmm.ss>
   
   **Note:** The format for the date is <<[century][year][month][day][hour][minute].[second]>>; for example, 201309081735.17.
   ```

3. **Configure the Network Time Protocol (NTP).**

   ```
cluster time-service ntp server create -server <<var_global_ntp_server_ip>>
   ```

**Configure Simple Network Management Protocol**

To configure the Simple Network Management Protocol (SNMP), complete the following steps:

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

   ```
   snmp contact <<var_snmp_contact>>
   snmp location "<<var_snmp_location>>"
   ```
Configure SNMPv1 Access

To configure SNMPv1 access, set the shared secret plain-text password, which is called a community.

```
snmp community delete all
snmp community add ro <<var_snmp_community>>
```

**Note:** Use the `delete all` command with caution. If community strings are used for other monitoring products, the `delete all` command removes them.

Create SNMPv3 User

SNMPv3 requires that a user be defined and configured for authentication. To create and configure a user for SNMPv3, complete the following steps:

1. Create a user called `snmpv3user`.

   ```
   security login create -username snmpv3user -authmethod usm -application snmp
   ```

2. Enter the authoritative entity’s engine ID and select `md5` as the authentication protocol.
3. Run the `security snmpusers` command to view the engine ID.
4. When prompted, enter a password for the authentication protocol. The password must have a minimum of eight characters.
5. Select `des` as the privacy protocol.
6. When prompted, enter a password for the privacy protocol. The password must have a minimum of eight characters.

Configure AutoSupport HTTPS

AutoSupport sends support summary information to NetApp through HTTPS. To configure AutoSupport, run the following command:

```
system node autosupport modify -node * -state enable -mail-hosts <<var_mailhost>> -transport https -support enable -to <<var_storage_admin_email>>
```

**Note:** To enable AutoSupport to send messages using SMTP, change the `-transport` value in the previous command to `smtp`. When configuring AutoSupport to use SMTP, be sure to enable mail relay on the mail server for the cluster management and node management IP addresses.

Configure Remote Support Agent

The Remote Support Agent (RSA) is configured directly on the storage controller’s remote management device firmware. It can only be installed on systems with an onboard service processor or a remote LAN module. To configure the RSA, complete the following steps:

1. Obtain SSH access to the first node’s service processor.
2. Run the `rsa setup` command.

```
SP <<node01_SF_ip>> rsa setup
```
3. Enter yes to enable the RSA.

Would you like to enable Remote Support Agent? [yes]: yes
Do you use a proxy to connect to the internet? [no]:

4. Enter the cluster management IP address of the cluster and enable SSL. The cluster management IP address is picked up automatically.

Enter the cluster management IP address of your storage cluster []: <<cluster_ip>>
Do you want to use HTTP with SSL? [yes]: yes
Enter HTTPS port number [443]:

5. Enter the credentials for the user with HTTP access on the cluster SVM.

Enter HTTP username []: <<http_user_on_cluster_SVM>>
Enter HTTP password: <<http_password_on_cluster_SVM>>

6. Commit the changes and make sure that all tests pass validation.

Do you want to commit configuration changes entered above? [yes]: yes
Committing configuration changes... done
Remote Support Agent is enabled.
Do you want to test current configuration? [yes]: yes
Testing cluster management LIF HTTP connection .................... ok
Testing Remote Support Enterprise connection .................... ok
All configuration tests passed.

7. Repeat steps 2 through 6 on the other node.

Create Operational SVM

Two SVMs (formerly called Vservers) are created for this reference architecture. To create an operational SVM for SAN booting through iSCSI, complete the following steps:

**Note:** The SVM is referred to as Vserver in the clustered Data ONTAP command-line interface (CLI).

1. Create the SVM.

   vserver create -vserv `openstack-admin` -rootvolume openstackadmin_root -aggregate aggr01_node01-rootvolume-security-style unix

   **Note:** The security style for the SVM becomes the default security style for all volumes created on that SVM. NetApp recommends the UNIX security style for SVMs that primarily support Linux environments. Block access is not affected by security style.

2. Remove protocols that are not needed from this SVM. Because this SVM supports iSCSI booting for only the eventual OpenStack compute nodes, remove all other protocols from the SVM.

   vserver remove-protocols -vserv `openstack-admin` -protocols cifs,fcp,nfs,ndmp

3. Add the two data aggregates to the aggregate list. This allows volumes to be created on these aggregates for this SVM.

   vserver modify -vserv `openstack-admin` -aggr-list aggr01_node01, aggr01_node02

Create Load-Sharing Mirror of SVM Root Volume

To create a load-sharing mirror of an SVM root volume, complete the following steps:

1. Create a volume to be the load-sharing mirror of the root volume of the infrastructure SVM (Vserver) on each node.
2. Create a job schedule to update the root-volume mirror relationships every 15 minutes.

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

3. Create the mirroring relationships.

```bash
snapmirror create -source-path //openstack-admin1/openstackadmin_root -destination-path //openstack-admin1/rootvol_m01 -type LS -schedule 15min
snapmirror create -source-path //openstack-admin1/openstackadmin_root -destination-path //openstack-admin1/rootvol_m02 -type LS -schedule 15min
```

4. Initialize the mirroring relationship.

```bash
snapmirror initialize -is-set -source-path //openstack-admin1/openstackadmin_root
```

**Create iSCSI Logical Interfaces (LIFs)**

To create the network interfaces that are used by the Cisco UCS compute nodes for booting, complete the following steps:

1. Create the interfaces on node 01.

```bash
network interface create -vserver openstack-admin1 -lif iscsi_lif01a -role data -data-protocol iscsi -home-node <<var_node01>> -home-port a0a<<var_iscsi_a_vlan_id>> -address <<var_node01_iscsi_lif01a_ip>> -netmask <<var_node01_iscsi_lif01a_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false
network interface create -vserver openstack-admin1 -lif iscsi_lif01b -role data -data-protocol iscsi -home-node <<var_node01>> -home-port a0a<<var_iscsi_b_vlan_id>> -address <<var_node01_iscsi_lif01b_ip>> -netmask <<var_node01_iscsi_lif01b_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false
```

2. Create the interface on node 02.

```bash
network interface create -vserver openstack-admin1 -lif iscsi_lif02a -role data -data-protocol iscsi -home-node <<var_node02>> -home-port a0a<<var_iscsi_a_vlan_id>> -address <<var_node02_iscsi_lif02a_ip>> -netmask <<var_node02_iscsi_lif02a_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false
network interface create -vserver openstack-admin1 -lif iscsi_lif02b -role data -data-protocol iscsi -home-node <<var_node02>> -home-port a0a<<var_iscsi_b_vlan_id>> -address <<var_node02_iscsi_lif02b_ip>> -netmask <<var_node02_iscsi_lif02b_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false
```

**Create NetApp FlexVol Volume and Enable Deduplication for Boot LUN Volumes**

To create the NetApp FlexVol volume that holds the necessary boot LUNs for each individual RHEL server in this infrastructure, complete the following steps:

1. Create the `rhel7_iscsi_boot` FlexVol volume.

```bash
volume create -vserver openstack-admin1 -volume rhel7_iscsi_boot -aggregate aggr01_node01 -size 750GB -state online -policy default -space-guarantee none -percent-snapshot-space 0
```

2. Enable deduplication on the boot LUN volume to enable space savings.

```bash
volume efficiency on -vserver openstack-admin1 -volume rhel7_iscsi_boot
```

3. Update the mirroring relationship.

```bash
snapmirror update -is-set -source-path //openstack-admin1/openstackadmin_root
```
Create Initiator Groups

Initiator groups (igroups) are tables of Fibre Channel Protocol (FCP) host worldwide port names or iSCSI host-node names. You can define igroups and map them to LUNs to control which initiators have access to LUNs. To do so, run the following commands:

```bash
igroup create -vserver openstack-admin1 -igroup OSP_Director -protocol iscsi -ostype linux -initiator <<var_osp_host0_a_iqn>>,<<var_osp_host0_b_iqn>>
igroup create -vserver openstack-admin1 -igroup osp8_controller_1 -protocol iscsi -ostype linux -initiator <<var_osp_host1_a_iqn>>,<<var_osp_host1_b_iqn>>
igroup create -vserver openstack-admin1 -igroup osp8_controller_2 -protocol iscsi -ostype linux -initiator <<var_osp_host2_a_iqn>>,<<var_osp_host2_b_iqn>>
igroup create -vserver openstack-admin1 -igroup osp8_controller_3 -protocol iscsi -ostype linux -initiator <<var_osp_host3_a_iqn>>,<<var_osp_host3_b_iqn>>
igroup create -vserver openstack-admin1 -igroup osp8_controller_4 -protocol iscsi -ostype linux -initiator <<var_osp_host4_a_iqn>>,<<var_osp_host4_b_iqn>>
```

Create LUNs

To create boot LUNs 50GB in size for each host in the infrastructure and to disable space reservation, run the following commands:

```bash
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_director_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_1_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_2_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_3_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_4_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_1_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_2_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_3_bootlun -size 50GB -ostype linux -space-reserve disabled
lun create -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_4_bootlun -size 50GB -ostype linux -space-reserve disabled
```

Note: It is recommended to have at least a 50GB LUN for RHEL7.

Map LUNs to Initiator Groups

To map the igroups to the actual boot LUN volumes, run the following commands:

```bash
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_director_bootlun -igroup OSP_Director -lun-id 0
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_1_bootlun -igroup osp8_controller_1 -lun-id 0
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_2_bootlun -igroup osp8_controller_2 -lun-id 0
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_3_bootlun -igroup osp8_controller_3 -lun-id 0
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_4_bootlun -igroup osp8_controller_4 -lun-id 0
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_1_bootlun -igroup osp8_controller_2 -lun-id 0
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_3_bootlun -igroup osp8_controller_3 -lun-id 0
lun map -vserver openstack-admin1 -volume rhel7_iscsi_boot -lun osp8_controller_4_bootlun -igroup osp8_controller_4 -lun-id 0
```
Setup for the operational SVM is complete.

Create Infrastructure SVM
To create an infrastructure SVM specifically for OpenStack and its related services, instances, shares, and configuration, complete the following steps:

1. Create the SVM.
   
   
   ```bash
   vserver create -vserver osp8-svm -rootvolume osp8svm_root -aggregate aggr01_node01 -rootvolume-security-style unix
   ```

2. Remove protocols that are not needed for this reference architecture from the SVM.

   ```bash
   vserver remove-protocols -vserver osp8-svm -protocols fcp,iscsi,ndmp
   ```

3. Add the two data aggregates to the aggregate list.

   ```bash
   vserver modify -vserver osp8-svm -aggr-list aggr01_node01, aggr01_node02
   ```

Create Load-Sharing Mirror of SVM Root Volume
To create a load-sharing mirror of an SVM root volume, complete the following steps:

1. Create a volume to be the load-sharing mirror of the root volume of the infrastructure SVM on each node.

   ```bash
   volume create -vserver osp8-svm -volume rootvol_m01 -aggregate aggr01_node01 -size 1GB -type DP
   volume create -vserver osp8-svm -volume rootvol_m02 -aggregate aggr01_node02 -size 1GB -type DP
   ```

2. Create the mirroring relationships.

   ```bash
   snapmirror create -source-path //osp8-svm/osp8svm_root -destination-path //osp8-svm/rootvol_m01 -type LS -schedule 15min
   snapmirror create -source-path //osp8-svm/osp8svm_root -destination-path //osp8-svm/rootvol_m02 -type LS -schedule 15min
   ```

3. Initialize the mirroring relationship.

   ```bash
   snapmirror initialize -ls-set -source-path //osp8-svm/osp8svm_root
   ```

Create Failover Group for SVM Management
To create a failover group for data NAS services for OpenStack, run the following command:

```bash
network interface failover-groups create -failover-group fg-nfs<<var_nfs_vlan_id>> -vserver openstack-svm -targets <<var_node01>>:a0a<<var_nfs_vlan_id>>,<<var_node02>>:a0a<<var_nfs_vlan_id>>
```  

Configure NFSv4
To configure NFS on the SVM, complete the following steps:

1. Set the NFSv4 ID mapping domain to match the DNS domain name used in the overall environment. This should match the default domain configured in /etc/idmapd.conf on each provisioned RHEL7 system, which is the host’s DNS domain name by default.

   ```bash
   nfs server modify -vserver osp8-svm -v4-id-domain rtp.netapp.com
   ```

2. Create a new export policy for the hosts that need storage access.

   ```bash
   vserver export-policy create -vserver osp8-svm -policyname openstack-hosts
   ```

3. Create a new export policy rule for the compute nodes used by this SVM. Set the user ID to which anonymous users are mapped to 0.

   ```bash
   vserver export-policy rule create -vserver osp8-svm -policyname openstack-hosts -type iscsi -export / -map 0
   ```
Note: For each RHEL host being created, create a rule. Each host has its own rule index. Your first RHEL host has rule index 1, your second RHEL host has rule index 2, and so on. Alternatively, you can specify the entire network in CIDR notation or use netgroups.

```bash
vserver export-policy rule create -vserver osp8-svm -policyname openstack-hosts -ruleindex 1 -
protocol nfs -clientmatch <<rhel_host1_nfs_ip>> -rorule sys -rwrule sys -superuser sys -allow-
suid false -anon 0
```

4. Modify the default UNIX user's group ID for the SVM's root user from 1 (the default) to 0.

```bash
unix user-modify -vserver osp8-svm -user root -primary-gid 0
```

5. Assign the new export policy to the infrastructure SVM root volume.

```bash
volume modify -vserver osp8-svm -volume osp8svm_root -policy openstack-hosts
```

Create Flexible Volumes (FlexVol Volumes) for Cinder and Glance

To create thick-provisioned FlexVol volumes (the volume’s name and size and the aggregate on which it exists) to have storage for both Cinder volumes and Glance images, complete the following steps:

1. Run the following command to create the cinder1 FlexVol volume on aggregate aggr01_node02:

   ```bash
   volume create -vserver osp8-svm -volume cinder1 -user 165 -group 165 -aggregate aggr01_node02 -
   size 1TB -state online -policy openstack-hosts -junction-path /cinder1 -space-guarantee volume -
   percent-snapshot-space 0
   ```

   **Note:** The following is displayed on the screen and is normal:

   ```
   Notice: Volume cinder1 now has a mount point from volume osp8svm_root. The load sharing (LS) mirrors of volume osp8svm_root will be updated according to the SnapMirror schedule in place for volume osp8svm_root. Volume cinder1 will not be visible in the global namespace until the LS mirrors of volume osp8svm_root have been updated.
   ```

2. Create the cinder2 FlexVol volume on aggregate aggr01_node01.

   ```bash
   volume create -vserver osp8-svm -volume cinder2 -user 165 -group 165 -aggregate aggr01_node01 -
   size 1TB -state online -policy openstack-hosts -junction-path /cinder2 -space-guarantee volume -
   percent-snapshot-space 0
   ```

3. Create the cinder3 FlexVol volume on aggregate aggr01_node02.

   ```bash
   volume create -vserver osp8-svm -volume cinder3 -user 165 -group 165 -aggregate aggr01_node02 -
   size 1TB -state online -policy openstack-hosts -junction-path /cinder3 -space-guarantee volume -
   percent-snapshot-space 0
   ```

4. Run the following command to create the Glance FlexVol volume:

   ```bash
   volume create -vserver osp8-svm -volume glance -user 161 -group 161 -aggregate aggr01_node02 -
   size 500GB -state online -policy openstack-hosts -junction-path /glance -space-guarantee volume -
   percent-snapshot-space 0
   ```

Enable Deduplication on Glance Volume

To enable deduplication on the Glance image repository volume, run the following command:

```bash
volume efficiency on -vserver osp8-svm -volume glance
```

**Note:** The volume efficiency schedule can be modified per the documentation [here](#) and can be configured to run during off-peak load times.
Create Additional FlexVol Volumes for Cinder (Optional)

This section is optional, but it is a NetApp best practice to have a minimum of three FlexVol volumes for Cinder in order to have the Cinder scheduler effectively load-balance between the different FlexVol volumes (referred to as back ends from a Cinder perspective).

1. Run the following command to create the archived data FlexVol volume, which is used to illustrate the storage service catalog (SSC) concept. Note that this volume is thin provisioned, has compression enabled, and has deduplication enabled.

   ```bash
   volume create -vserver osp8-svm -volume archived_data -user 165 -group 165 -size 500GB -aggregate aggr01_node01 -space-guarantee none -policy openstack-hosts -junction-path /archived_data -percent-snapshot-space 0
   ```

2. Enable deduplication on the archived_data FlexVol volume.

   ```bash
   volume efficiency on -vserver osp8-svm -volume archived_data
   ```

3. Enable compression on the archived_data FlexVol volume.

   ```bash
   volume efficiency modify -vserver osp8-svm -volume archived_data -compression true
   ```

4. Update the SVM root volume load-sharing mirrors. This allows mounts to be accessible by making the new mount points visible to the destination load-sharing mirror volumes.

   ```bash
   snapmirror update-ls-set -source-path osp8-svm:osp8svm_root
   ```

NFS Logical Interfaces

To create NFS logical interfaces (LIFs) for the Cinder and Glance volumes, run the following commands to create LIFs for both nodes:

**Note:** In this example, the specified failover group is the failover group configured in the section “Create Failover Group for SVM Management.”

```bash
network interface create -vserver osp8-svm -lif openstack_nfs_1_lif -role data -data-protocol nfs -home-node <<var_node01>> -home-port a0a<<var_nfs_vlan_id>> -address <<nfs_lif_ip1>> -netmask <<nfs_lif_netmask1>> -status-admin up -failover-policy nextavail -firewall-policy data -auto-revert true -failover-group fg-nfs<<var_nfs_vlan_id>>

network interface create -vserver osp8-svm -lif openstack_nfs_2_lif -role data -data-protocol nfs -home-node <<var_node02>> -home-port a0a<<var_nfs_vlan_id>> -address <<nfs_lif_ip2>> -netmask <<nfs_lif_netmask2>> -status-admin up -failover-policy nextavail -firewall-policy data -auto-revert true -failover-group fg-nfs<<var_nfs_vlan_id>>
```

Start NFS Server and Enable Advanced NFS Options

To start the NFS service, enable NFS v4.1 with parallel NFS (pNFS) support and enable the NetApp copy offload-capable feature, complete the following steps:

1. Run the following command to enable the NFS service on the osp8-svm SVM:

   ```bash
   vserver nfs on -vserver osp8-svm
   ```

2. Enable advanced NFS options.

   ```bash
   vserver nfs modify -vserver osp8-svm -v4.0 enabled -v4.1 enabled -v4.1-pnfs enabled -vstorage enabled
   ```

Gather Target Information from FAS8040

Run the `iscsi show` command to get the target information required for configuration in the Cisco UCS server node section.

The target name for future steps is shown in Table 17.
Table 17) iSCSI target name for Cisco UCS booting.

<table>
<thead>
<tr>
<th>SVM</th>
<th>Target Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>openstack-admin1</td>
<td>iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6</td>
</tr>
</tbody>
</table>

3.6 NetApp E-Series E5660 Setup

For the physical installation of the E-Series E5660, follow the procedures in the E5600 series documentation on the NetApp Support site. When planning the physical location of the storage systems, refer to the following sections in the NetApp E-Series and EF-Series Site Preparation Guide:

- System Connectivity Requirements
- Circuit Breaker, Power Outlet Balancing, System Cabinet Power Cord Plugs, and Console Pinout Requirements
- Specifications of the DE6600 Drive Tray

Disk Shelves

NetApp storage systems support a wide variety of disk shelves and disk drives. Visit the NetApp Support site to view a complete list of supported E-Series Disk Shelves.

This solution is built on a DE6600 disk shelf with 60 3TB SAS disks owned by two E5600 controllers. These disks provide an ample amount of storage at a moderate price point and are recommended for an OpenStack deployment, specifically for object storage (Swift).

For information about E-Series cabling guidelines, see the E-Series and EF-Series Systems Hardware Cabling Guide.

SANtricity OS

This procedure assumes that the storage system has been installed and cabled and is ready for setup. For detailed information about storage system installation, see the resources provided previously.

Complete Configuration Worksheet

Before you begin the setup, review the requirements for configuring SANtricity OS in SANtricity Storage Manager 11.25 Software Installation Reference. Table 18 lists the information that you need to configure the E-Series E5660. You should customize the values in this table with information that is applicable to your deployment.

Table 18) SANtricity OS software configuration worksheet.

<table>
<thead>
<tr>
<th>Array Detail</th>
<th>Array Detail Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Series storage array name</td>
<td>&lt;&lt;var_storagearrayname&gt;&gt;</td>
</tr>
<tr>
<td>Controller A, Port 1 management interface</td>
<td>&lt;&lt;var_contA_management_ip&gt;&gt;</td>
</tr>
<tr>
<td>Controller B, Port 1 management interface</td>
<td>&lt;&lt;var_contB_management_ip&gt;&gt;</td>
</tr>
<tr>
<td>Controller A, Port 1 management netmask</td>
<td>&lt;&lt;var_contA_management_mask&gt;&gt;</td>
</tr>
<tr>
<td>Controller B, Port 1 management netmask</td>
<td>&lt;&lt;var_contB_management_mask&gt;&gt;</td>
</tr>
<tr>
<td>Array management gateway</td>
<td>&lt;&lt;var_management_gateway&gt;&gt;</td>
</tr>
<tr>
<td>iSCSI port Controller A, HIC 1, Port 1</td>
<td>&lt;&lt;var_iscsiConta_hic1_p1_ip&gt;&gt;</td>
</tr>
</tbody>
</table>
### Array Detail

<table>
<thead>
<tr>
<th>Array Detail</th>
<th>Array Detail Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSCSI port Controller A, HIC 1, Port 1 netmask</td>
<td>&lt;&lt;var_iscsi_conta_hic1_p1_netmask&gt;&gt;</td>
</tr>
<tr>
<td>iSCSI port Controller A, HIC 1, Port 3</td>
<td>&lt;&lt;var_iscsi_conta_hic1_p3_ip&gt;&gt;</td>
</tr>
<tr>
<td>iSCSI port Controller A, HIC 1, Port 3 netmask</td>
<td>&lt;&lt;var_iscsi_conta_hic1_p3_netmask&gt;&gt;</td>
</tr>
<tr>
<td>iSCSI port Controller B, HIC 1, Port 1</td>
<td>&lt;&lt;var_iscsi_contb_hic1_p1_ip&gt;&gt;</td>
</tr>
<tr>
<td>iSCSI port Controller B, HIC 1, Port 1 netmask</td>
<td>&lt;&lt;var_iscsi_contb_hic1_p1_netmask&gt;&gt;</td>
</tr>
<tr>
<td>iSCSI port Controller B, HIC 1, Port 3</td>
<td>&lt;&lt;var_iscsi_contb_hic1_p3_ip&gt;&gt;</td>
</tr>
<tr>
<td>iSCSI port Controller B, HIC 1, Port 3 netmask</td>
<td>&lt;&lt;var_iscsi_contb_hic1_p3_netmask&gt;&gt;</td>
</tr>
<tr>
<td>OpenStack Controller-1 IQN A</td>
<td>&lt;&lt;var_controller01_iqn_a&gt;&gt;</td>
</tr>
<tr>
<td>OpenStack Controller-1 IQN B</td>
<td>&lt;&lt;var_controller01_iqn_b&gt;&gt;</td>
</tr>
<tr>
<td>OpenStack Controller-2 IQN A</td>
<td>&lt;&lt;var_controller02_iqn_a&gt;&gt;</td>
</tr>
<tr>
<td>OpenStack Controller-2 IQN B</td>
<td>&lt;&lt;var_controller02_iqn_b&gt;&gt;</td>
</tr>
<tr>
<td>OpenStack Controller-3 IQN A</td>
<td>&lt;&lt;var_controller03_iqn_a&gt;&gt;</td>
</tr>
<tr>
<td>OpenStack Controller-3 IQN B</td>
<td>&lt;&lt;var_controller03_iqn_b&gt;&gt;</td>
</tr>
</tbody>
</table>

### Configure Management Interfaces

By default, E-Series systems ship with DHCP enabled on all management network ports. If the controller is unable to obtain a DHCP address on a given network port, it reverts to a default IP address, which can take up to three minutes.

To perform the initial configuration of the management interfaces, complete the following steps:

1. Wait three minutes after booting the system and then plug a service laptop into controller A, Port 1 with an RJ-45 crossover cable.
2. Configure the service laptop with an IP address of 192.168.128.201 and a subnet mask of 255.255.255.0. Leave the gateway blank.
3. You should be able to ping the system at 192.168.128.101. If not, check the configuration and network cabling before moving on to the next step.
4. Start the SANtricity Storage Manager client and manually add the storage system to the client by clicking Edit and Add Storage Array.
5. The storage array should be discovered.
   **Note:** Disregard warnings about not adding another controller.

6. To rename the storage system with a descriptive name, right-click the current name and select Rename.
   **Note:** Use the <<var_storagearrayname>> value listed in Table 18.

7. If the storage subsystem is not on firmware version 8.25.04.00 or later, refer to the SANtricity Storage Manager 11.25 System Upgrade Guide for instructions about how to upgrade the controller firmware and NVSRAM.
   To upgrade drive firmware to the latest levels where applicable, see the SANtricity Storage Manager 11.25 System Upgrade Guide.
   To download drive firmware, see the E/EF-Series Disk Drive and Firmware Matrix on the NetApp Support site.

8. Click the storage system to launch the Array Management window.

9. From the Setup tab, scroll down to Optional Tasks and click Configure Ethernet Management Ports.

10. Configure the appropriate values for Controller A, Port 1, and Controller B, Port 1. Disable IPv6 if it does not apply to your environment. Click OK and accept any changes.
11. Unplug the service laptop from the storage system and connect the management ports to the upstream data center network. The system should now be accessible through the IP addresses configured in step 10. They are accessible by pinging the controller management interfaces.

Create Disk Pool

Now that the storage array is accessible on the network, relaunch SANtricity Storage Manager to create disk pools based on the number of hosts connected to the subsystem. For this reference architecture, create pools of 20 drives each, with a total of three disk pools. These three disk pools represent the three OpenStack controller systems that are used as Swift proxy nodes.

**Note:** A minimum of 11 drives per drive pool is required.

1. Start the SANtricity Storage Manager client and click the recently discovered storage array. The Array Management window is displayed.
2. Click No to manually configure the disk pools.
3. From the Storage and Copy Services tab, right-click Total Unconfigured Capacity and select Create Disk Pool.

4. Scroll down and select a drive count of 20. Keep the name Disk_Pool_1 and click Create.

5. Repeat step 4 for Disk_Pool_2 and Disk_Pool_3 (with 20 disks for each).
6. Right-click each new disk pool and select Change and then Settings. Deselect the Critical Warning Notification Threshold option and click OK. This configuration silences the warnings that the disk pool is over capacity after the volumes are created.

**Note:** Repeat this step on all three disk pools. Otherwise, Recovery Guru in SANtricity Storage Manager indicates an error condition.

![Image of Change Disk Pool Settings](image)

### Create Volume

Volumes can now be created from each of the disk pools that were formed in the previous section. NetApp recommends creating an even number of LUNs of equal size on a per-controller basis. Swift lays down account, container, and object data equally across both storage controllers to maximize performance using this methodology.

The default mapping for volumes to hosts (through LUN mapping) is to expose all volumes to all hosts. To make sure that multiple hosts are not accessing the same LUN concurrently, map each volume to the appropriate host to which it should mount.

**Note:** If SSDs are present, create separate disk pools that contain only SSDs.

1. Right-click the Free Capacity of Drive_Pool_1 and click Create Volume.
2. Divide the total usable capacity of the volume by four. Enter the size of 9832.500GB and name the volume pool1_vol0.
3. Click Finish and then click Yes to create another volume.
4. Enter the size 9832.500GB and name the volume pool1_vol1.
5. Click Finish and then click Yes to create another volume.
6. Enter the size 9,832.500GB again, this time with a name of pool1_vol2.
7. Click Finish and then click Yes to create another volume.
8. Create a volume with the name pool1_vol3 and input the remainder of space left in the disk pool; in our case it was 9,824.000GB for Disk_Pool_1. Click Finish.
9. Repeat steps 1 through 8, substituting Drive Pool_2 and Drive Pool_3 for Drive Pool_1.

The following information is displayed in the navigation pane of the Array Management window:
Map Host

To define the hosts that connect to the E-Series storage array and their associated host port identifiers, complete the following steps:

1. Select the Host Mappings tab.
2. Select Don’t Display This Dialog Again in the Mapping Start-Up Help window. Click Close.
3. Right-click the volume named Access and click Remove. This volume represents an in-band management drive and is not needed. Enter yes in the text box and click OK.
4. Right-click Default Group, select Define, and then select Host.

5. The Define Host wizard is displayed. Enter Swift_Node0 in the Host Name field.
6. Select No for whether or not to use storage partitions on this storage array.
7. Select Add by Creating a New Host Port Identifier and enter <<var_controller01_iqn_a>> in the New Host Port Identifier text box.
8. For the alias/user label, enter Swift_Node0_A.
9. Click Add to add the host port identifier to the specified host.
10. Repeat step 7, but substitute <<var_controller01_iqn_b>> in the host port identifier text box.
11. For the alias/user label, enter Swift_Node0_B.
12. Click Add. Before clicking Next, verify that the associations are correct for your environment.

<table>
<thead>
<tr>
<th>Host Port Identifier</th>
<th>Alias / User Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;var_controller01_iqn_a&gt;&gt;</td>
<td>Swift_Node0_A</td>
</tr>
<tr>
<td>&lt;&lt;var_controller01_iqn_b&gt;&gt;</td>
<td>Swift_Node0_B</td>
</tr>
</tbody>
</table>

13. Select the host type Linux (DM-MP). This host type represents a Linux host using the DM-Multipath subsystem. Click Next.
14. Preview the new host to be created and click Finish.
15. Click Yes when asked to define other hosts.
16. Repeat steps 5 through 15 for the remaining two hosts to be added to the storage array. Substitute the appropriate values listed in Table 18.

**Map LUN**

After the hosts are defined and the host port identifiers are associated, map the volumes as LUNs to hosts. Volume-to-LUN mapping gives hosts access to the storage as host LUNs by using the SCSI device driver in Linux in tandem with the device-mapper multipath subsystem for HA. The LUNs are accessible as SCSI /dev/mapper/mpathX[1-3] devices, where X is a unique identifier assigned by the SCSI subsystem during the device discovery process.
To map the LUNs, complete the following steps:

1. From the Host Mapping tab, expand Undefined Mappings. All of the volumes that were previously created are displayed.
   
   **Note:** The question marks are associated with LUN numbers because they are currently undefined.

2. Right-click `pool1_vol0` and select Add LUN Mapping.

![Diagram showing undefined mappings]

3. The Define Additional Mapping window is displayed. Select `pool1_vol0` and map it to host `Swift_Node0`, LUN 0. Click Add.

4. Select `pool1_vol1` and map it to host `Swift_Node0`, LUN 1. Click Add.

5. Select `pool1_vol2` and map it to host `Swift_Node0`, LUN 2. Click Add.

6. Repeat steps 3 through 5 for `Swift_Node1` and `Swift_Node2`. The final configuration should appear similar to the following example:
Configure iSCSI Host Ports

In this section, you configure network parameters for the iSCSI host ports on the controllers, such as IP addressing and MTU. For this reference architecture, four ports were chosen, two from each controller spread across application-specific integrated circuits (ASICs) on the board.

1. From the Setup tab, select the Configure iSCSI Host Ports option.
2. The Configure iSCSI Ports dialog box appears.
3. For the iSCSI port named Controller A, HIC 1, Port 1, enter the pertinent IPv4 configuration listed in Table 18.
4. Deselect the Enable IPv6 option if it does not apply to your environment.
5. Click Advanced Port Settings.
6. Under Jumbo Frames, select Enable Jumbo Frames and enter 9000 bytes/frame.
7. Click OK in the Advanced Port Settings window.
8. In the iSCSI port menu, select Controller A, HIC 1, Port 3.
9. Repeat steps 3 through 7, substituting information from Table 18 for this port.
10. In the iSCSI port drop-down menu, select Controller B, HIC 1, Port 1.
11. Repeat steps 3 through 7, substituting information from Table 18 for this port.
12. In the iSCSI port drop-down menu, select Controller B, HIC 1, Port 3.
13. Repeat steps 3 through 7, substituting information from Table 18 for this port.
14. Make sure that the Enable ICMP PING Responses option is enabled. This option aids in troubleshooting.
15. Click OK in the Configure iSCSI Ports window.

**Note:** This process can take a few minutes to complete.
16. The window should close automatically when the process is finished.

**Disable Unused iSCSI Host Ports**

If iSCSI host ports are not being used, they should be disabled so that iSCSI-based logins from RHEL7 systems don’t attempt to use them. To disable unused iSCSI host ports, complete the following steps:

1. From the Setup tab, select the Configure iSCSI Host Ports option.
2. The Configure iSCSI Ports dialog box appears.
3. Select iSCSI port Controller A, HIC 1, Port 2 from the drop-down menu.
4. Deselect the Enable IPv4 and Enable IPV6 options.
5. Repeat steps 3 and 4 for all ports that are not used.
   
   **Note:** This example shows the ports that were disabled in the NetApp lab.

   ![Configure iSCSI Ports dialog box](image)

   - **IPv4 Settings**
   - **IPv4 Configuration**
   - **IPv4 Settings**

6. Click Close when the process is complete.

**Verify Write Cache Mirroring**

By default, NetApp E-Series controllers have write cache mirroring enabled per volume. To verify write cache mirroring, complete the following steps:

1. From the Storage & Copy Services tab, expand the Disk Pools drop-down list.
2. Right-click one of the volumes, select Change, and then select Cache Settings.
3. The Change Cache Settings dialog box is displayed.

4. Click any of the respective volumes. Checkboxes for the following cache properties are displayed:
   - Enable read caching
   - Enable dynamic cache read prefetch
   - Enable write caching
   - Enable write caching with mirroring

5. Make sure that all four boxes are selected for each respective volume. In this validation, 12 total volumes exist.

   **Note:** If a failover scenario exists where only one E-Series controller is active, write caching with mirroring adversely affects system performance. If the storage system operates on a single controller for a prolonged period, repeat the steps in this procedure and make sure that write caching with mirroring is disabled for each volume in the system.

   **Note:** NetApp does not recommend running a storage system on a single controller. Failed controllers should be replaced as soon as possible to return the system to a fully redundant state.

3.7 Cisco UCS Setup

Setting up the Cisco UCS in a step-by-step manner is outside the scope of this document. For step-by-step instructions, see the [Cisco UCS and Server Configuration](#) section of the "FlexPod Datacenter with Red Hat OpenStack Platform Design Guide." This document is loosely based on the Cisco UCS configuration setup, with the following notable differences:
• The Cisco UCS version deployed in this document is newer and is listed in the section titled “Solution Software.” Be sure to upgrade both the infrastructure and server firmware to the version listed.

• The number of VLANs, their respective names, and their IDs (802.1Q tags) are different in this document. Consult the section titled “Necessary VLANs” for guidance.

• vNIC templates are different to accommodate network segmentation:
  – For the undercloud, the service profile template has an iSCSI-A, iSCSI-B, a PXE (to listen and respond to requests from overcloud servers being provisioned), and an OOB-Management vNIC. Figure 18 illustrates the undercloud in the NetApp lab.

  Figure 18) Undercloud vNIC and network segmentation.

  ![Undercloud vNIC and network segmentation diagram]

  – For the overcloud, the service profile template has an iSCSI-A, iSCSI-B, PXE, and an OSP-A and OSP-B vNIC. The OSP-A and OSP-B vNICs have the following VLANs carried to them: NFS, OSP-StorMgmt, OSP-Backend, OOB-Management, Tunnel, and Public. These two vNICs are bonded together in a link aggregation using mode 1, or active backup. The bridge is named br-ex.

  Figure 19 and Figure 20 illustrate the overcloud in the NetApp lab.

  Figure 19) Overcloud controller vNIC and network segmentation.

  ![Overcloud controller vNIC and network segmentation diagram]
An IPMI access profile has been set up with a user name and password combination that accommodates OpenStack Ironic being used to turn on and off associated service profile templates for overcloud deployments.

4 Solution Deployment

After the infrastructure components are configured, the software components must be configured.

4.1 Deploy Red Hat OpenStack Platform Director

The Red Hat OpenStack Platform director node is commonly referred to as the undercloud. This node is responsible for deploying RHEL OpenStack Platform 8 on FlexPod in a highly available, automated manner.

Install RHEL 7.2

To install RHEL 7.2, complete the following steps:

1. After logging in to Cisco Unified Computing System Manager (UCSM), associate the RHEL7.2 DVD with the intended service profile and boot it. When the RHEL7.2 splash screen is displayed, quickly press the Tab key to override the listing of default entries.

2. Pass rd.scsi.ibft=1 to the kernel command line and press Enter.
3. Click Continue.
4. Click Installation Destination and then click Done in the upper left corner.
5. Click Network & Host Name.
6. Enter a host name for the system, enable any pertinent NICs, and assign the static IP addressing information for this server. Click Done.
7. Click Begin Installation.
8. Click Root Password to set a root password while the system is installing.
9. After the system is finished building, reboot it by clicking Reboot.
10. After the system boots for the first time, verify that you can reach external entities. If you can't, fix the server's network configuration in the `/etc/sysconfig/network-scripts` directory before proceeding further.

```plaintext
[root@osp-director ~]# ping flexpod.com -c 4
PING flexpod.com (66.6.44.4) 56(84) bytes of data.
64 bytes from 66.6.44.4: icmp_seq=1 ttl=52 time=16.3 ms
64 bytes from 66.6.44.4: icmp_seq=2 ttl=52 time=16.4 ms
64 bytes from 66.6.44.4: icmp_seq=3 ttl=52 time=16.4 ms
64 bytes from 66.6.44.4: icmp_seq=4 ttl=52 time=16.4 ms
--- flexpod.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 16.366/16.417/16.444/0.095 ms
```

**Add Stack Account and Check for Host Name Problems**

Configure the server for important prerequisite information before running the undercloud deployment script. To add the stack account and check for any host name problems, complete the following steps:

1. Create the user `stack` and give it a strong password.
   
   **Note:** For security purposes, configure the undercloud using the `stack` account, not the `root` account.

   ```plaintext
   [root@osp-director ~]# useradd stack
   [root@osp-director ~]# passwd stack
   Changing password for user stack.
   New password: 
   Retype new password:
   ```
passwd: all authentication tokens updated successfully.

2. Give the stack user the ability to run commands as root by using sudo:

   [root@osp-director ~]# echo "stack ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/stack
   stack ALL=(root) NOPASSWD:ALL

3. Fix file ownership in the /etc/sudoers.d/stack file.

   [root@osp-director ~]# chmod 0440 /etc/sudoers.d/stack

4. Change to the stack account.

   Note: Complete the rest of this section as the stack user.

   [root@osp-director ~]# su - stack
   [stack@osp-director ~]$ 

5. Create the images directory to house tailored Red Hat image files that help with the overcloud installation and deployment.

   [stack@osp-director ~]$ mkdir ~/images

6. Set the system host name for proper functionality with Puppet-based orchestration that is used later in this installation procedure.

   [stack@osp-director ~]$ sudo hostnamectl set-hostname osp-director.rtp.netapp.com
   [stack@osp-director ~]$ sudo hostnamectl set-hostname --transient osp-director.rtp.netapp.com

7. Make sure that the system recognizes its host name by running the following commands:

   [stack@osp-director ~]$ hostname
   osp-director.rtp.netapp.com
   [stack@osp-director ~]$ hostname -f
   hostname: Name or service not known

   If hostname -f returns with the message Name or service not known, make sure that an entry exists in /etc/hosts for the undercloud server.

   Note: Create an entry if one does not exist.

   127.0.0.1   osp-director.rtp.netapp.com localhost localhost.localdomain localhost4
   localhost4.localdomain4 :1   localhost localhost.localdomain localhost6 localhost6.localdomain6

9. Rerun the hostname -f command.

   [stack@osp-director ~]$ hostname -f
   osp-director.rtp.netapp.com

Register System and Perform Miscellaneous Server Configurations

The system must be registered with Red Hat's Subscription Manager Tool in order to install the undercloud and its associated packages. Other configurations are also required before launching the undercloud deployment script.

To register the system and perform the miscellaneous configurations, complete the following steps:

1. Register the system, substituting the user name and password information that are specific to your environment.

   Note: The final command output is truncated because each repository is being disabled. Only the repositories pertinent to Red Hat OpenStack Platform 8 are specifically enabled.

   [stack@osp-director ~]$ sudo subscription-manager register --username userName --password mySecretPassword
   Registering to: subscription.rhn.redhat.com:443/subscription
   The system has been registered with ID: <snip>
2. Disable the repositories that are currently enabled.

```
[stack@osp-director ~]$ sudo subscription-manager repos --disable=* 
```

3. Enable the following repositories:

```
[stack@osp-director ~]$ sudo subscription-manager repos --enable=rhel-7-server-rpms--
enable=rhel-7-server-extras-rpms--enable=rhel-7-server-openstack-8-rpms--enable=rhel-7-server-
openstack-8-director-rpms--enable=rhel-7-server-rh-common-rpms 
```

4. Verify that the system has the latest base packages before installing the director-related packages.

```
[stack@osp-director ~]$ sudo yum -y update 
```

5. Accurate time is critical. Install and configure the NTP.

```
[stack@osp-director ~]$ sudo yum -y install ntp 
```

6. Paste the following information into /etc/ntp.conf, substituting time.netapp.com for an NTP server specific to your environment:

```
driftfile /var/lib/ntp/drift 
restrict default nomodify notrap nopeer noquery 
restrict 127.0.0.1 
server time.netapp.com 
includefile /etc/ntp/crypto/pw 
keys /etc/ntp/keys 
disable monitor 
logfile /var/log/ntp.log 
```

7. Allow NTP outbound through the firewall and start the NTP daemon.

```
[stack@osp-director ~]$ sudo firewall-cmd --add-service=ntp --permanent 
[stack@osp-director ~]$ sudo firewall-cmd --reload 
[stack@osp-director ~]$ sudo systemctl start ntpd 
[stack@osp-director ~]$ sudo systemctl enable ntpd 
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to 
/usr/lib/systemd/system/ntpd.service. 
[stack@osp-director ~]$ sudo systemctl status ntpd 
```

```
ntpd.service - Network Time Service 
Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled) 
Active: active (running) since Thu 2016-04-07 21:19:55 EDT; 17s ago 
Main PID: 30248 (ntpd) 
CGProcess: /system.slice/ntpd.service 
   30248 /usr/sbin/ntpd -u ntpd:ntp -g 
Apr 07 21:19:55 osp-director.rtp.netapp.com systemd[1]: Starting Network Time Service... 
Apr 07 21:19:55 osp-director.rtp.netapp.com ntpd[30248]: proto: precision = 0.628 usec 
Apr 07 21:19:55 osp-director.rtp.netapp.com ntpd[30248]: 0.0.0.0 c01d 0d kern kernel time sync 
   enabled 
```

8. Verify the synchronization with the NTP server.
9. Synchronize the hardware clock to the system clock.

```
[stack@osp-director ~]$ ntpq -p
```

```
remote refid st t when poll reach delay offset jitter
*ns11.corp.netapp 10.32.32.20 2 u 55 64 1 62.102 0.033 0.000
```

10. Verify that the hardware clock and the system clock are synchronized.

```
[stack@osp-director ~]$ sudo hwclock -systohc
```

```
Thu Apr 7 21:20:46 EDT 2016
Thu 07 Apr 2016 09:20:47 PM EDT
```

11. Install the supported version of Cisco eNIC firmware. Download the most recent version from the Cisco Support site.

**Note:** See the hardware and software certification links in the References section to find the supported version of the eNIC firmware, which is determined by the UCSM, the adapter, and the underlying OS version. The hardware combination in the NetApp lab required version 2.3.0.18.

```
[stack@osp-director ~]$ sudo rpm -ivh kmod-enic-2.3.0.18-rhel7u2.el7.x86_64.rpm
Preparing...
Updating / installing...
1:kmod-enic-2.3.0.18-rhel7u2.el7                             [100%]
```

12. After the installation is complete, reboot the system.

```
[stack@osp-director ~]$ sudo reboot
```

13. When the system is back up, log in as root and verify that it has the intended version of the Cisco eNIC driver firmware.

```
[root@osp-director ~]$ dmesg | grep enic | grep ver
[ 4.103159] enic: Cisco VIC Ethernet NIC Driver, ver 2.3.0.18
```

14. Make sure that the system sees a total of four paths from the NetApp FAS device.

```
root@osp-director ~]$ iscsiadm --mode discovery --type sendtargets --portal "$(< /sys/firmware/ibft/target0/ip-addr)"
192.168.188.5:3260,1038 iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6
192.168.189.5:3260,1039 iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6
192.168.189.4:3260,1037 iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6
192.168.189.5:3260,1036 iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6
```

```
[root@osp-director ~]$ iscsiadm --mode node --login
Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6, portal: 192.168.189.5,3260] (multiple)
Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6, portal: 192.168.188.4,3260] (multiple)
Login to [iface: default, target: iqn.1992-08.com.netapp:sn.2cbc21e16bc111e5a02d00a0985e420a:vs.6, portal: 192.168.188.4,3260] successful.
```

15. Verify that DM multipath has updated information for the new paths.

```
[root@osp-director ~]$ multipath -ll
360a0a98038330346562b47369693048 dm-0 NETAPP ,LUN C-Mode
size=50G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-- policy='service-time 0' prio=50 status=active
| `- 2:0:0:0 sdb 8:16 active ready running
| `- 4:0:0:0 sdd 8:48 active ready running
`- policy='service-time 0' prio=10 status=enabled
| `- 1:0:0:0 sda 8:0 active ready running
| `- 3:0:0:0 sdc 8:32 active ready running
```
Install Director Packages

After the system updates with the base packages and contains the recommended number of paths to the storage system, proceed with the director package installation.

To install the director packages, complete the following step:

1. Run the following command:

   ```bash
   [stack@osp-director ~] $ sudo yum -y install python-tripleoclient
   ```

Download and Configure FlexPod Heat Templates and Postdeployment Scripts

As described in section 2.6, “Heat Orchestration Templates (HOT) and Overcloud Customization,” NetApp provides several Heat templates and postdeployment scripts to help the reader configure the OSP director with the proper parameters for the eventual OpenStack deployment.

**Note:** The script files are located at [https://github.com/NetApp/snippets/tree/master/RedHat/osp8-liberty/tr](https://github.com/NetApp/snippets/tree/master/RedHat/osp8-liberty/tr).

To download and configure FlexPod HOT templates and postdeployment scripts, complete the following steps on the director machine:

1. Install Subversion (SVN) to clone the code repository hosted on GitHub to the director machine. Keep the folder structure present on GitHub the same as in the `/home/stack` directory.

   ```bash
   [stack@osp-director ~] $ sudo yum -y install svn
   ```

2. Clone the repository to the `/home/stack` directory on the director machine.

   **Note:** Be aware of the period (.) at the end of the following command:

   ```bash
   [stack@osp-director ~] $ cd /home/stack;
   ```

3. Modify the `/home/stack/undercloud.conf` file. Modify the variables listed in Table 19 to match your respective environment.
Table 19) Undercloud.conf variable definitions.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Variable Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>local_ip</td>
<td>The IP address that is defined for the director’s provisioning NIC in CIDR notation. The director also uses this IP address for its DHCP and PXE boot services.</td>
</tr>
<tr>
<td>local_interface</td>
<td>The chosen interface for the director’s provisioning NIC. The director uses this device for its DHCP and PXE boot services.</td>
</tr>
<tr>
<td>masquerade_network</td>
<td>Input the provisioning network here in CIDR notation.</td>
</tr>
<tr>
<td>dhcp_start, dhcp_end</td>
<td>The start and end of the DHCP allocation range for overcloud nodes. Make sure that this range contains enough IP addresses to allocate to your nodes.</td>
</tr>
<tr>
<td>network_cidr</td>
<td>The provisioning network that the director uses to manage overcloud instances. It is specified in CIDR notation.</td>
</tr>
<tr>
<td>network_gateway</td>
<td>The gateway for the overcloud instances while provisioning. It is the director host that forwards traffic to the external network. Later in the provisioning process, we use the gateway on the OOB-Management network for external access so that all outbound traffic does not traverse through the director host.</td>
</tr>
<tr>
<td>discovery_iprange</td>
<td>A range of IP addresses that the director’s discovery service uses during the PXE boot and provisioning process. NetApp chose the higher order IP addresses in the 172.21.19.0/24 range. <strong>Note:</strong> Use comma-separated values to define the start and end of the IP address range.</td>
</tr>
</tbody>
</table>

In the NetApp lab, the following definitions existed in the `/home/stack/undercloud.conf` file.

**Note:** As a part of this document, `undercloud.conf` is available on GitHub. It should reside in the `/home/stack` directory after running the commands listed in the section titled “Download and Configure FlexPod Heat Templates and Postdeployment Scripts.”

```
[DEFAULT]
local_ip = 172.21.19.18/24
local_interface = enp6s0
masquerade_network = 172.21.19.0/24
dhcp_start = 172.21.19.21
dhcp_end = 172.21.19.200
network_cidr = 172.21.19.0/24
network_gateway = 172.21.19.18
```

**Install Undercloud**

To install the undercloud, complete the following steps:

1. Before launching the undercloud deployment script, verify that the interfaces specified in the `/home/stack/undercloud.conf` file are functioning properly.
   **Note:** Unnecessary output is truncated.

2. Run the following commands and verify the output carefully:

```
[stack@osp-director ~]$ ip addr list
2: enp6s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
   link/ether 00:07:00:aa:00:00 brd ff:ff:ff:ff:ff:ff
```

©2016 NetApp, Inc. All rights reserved.
Note: It is now safe to launch the undercloud deployment script to configure the director node with the configurations in /home/stack/undercloud.conf.

3. Prevent Puppet from overriding entries in the /etc/resolv.conf file (presumably due to the iBFT interfaces), which prevents the deployment script from functioning properly because DNS resolution would be prevented from functioning at this stage.

```bash
[stack@osp-director ~]$ sudo chattr +i /etc/resolv.conf
```

4. Run the following command as the stack user:

```bash
[stack@osp-director ~]$ openstack undercloud install
```

Note: At this point, packages are installed from relevant yum repositories, SELinux is configured, Puppet is running on the undercloud, and other configuration elements are configured for the undercloud.

5. After the installation is complete, the following message is displayed:

```
#########################################################################
# Undercloud install complete.
The file containing this installation's passwords is at /home/stack/undercloud-passwords.conf.
There is also a stackrc file at /home/stack/stackrc.
These files are needed to interact with the OpenStack services, and should be secured.
#########################################################################
```

6. Proceed to the next section to deploy the overcloud on FlexPod.

### 4.2 Customize Red Hat OpenStack Platform Director

The Red Hat OpenStack Platform director requires the following disk images in order to provision overcloud nodes:

- **Discovery kernel and ramdisk.** Used for bare metal system discovery over PXE boot.
- **Deployment kernel and ramdisk.** Used for system provisioning and deployment.
- **Overcloud kernel, ramdisk, and full image.** A base overcloud system written to the node’s hard disk.

Note: Additional Red Hat OpenStack Platform director customization is provided in the remainder of this section.
Download Overcloud Images

To download overcloud images, complete the following steps:

1. Obtain the overcloud images from the 4.7 Obtaining Images for Overcloud Nodes section of Director Installation and Usage on the Red Hat Customer Portal.

   **Note:** The Ironic Python Agent (IPA) does not support discovery or deployment to servers with iSCSI-backed root disks. The OSP7 discovery and deployment images must be used for node discovery and deployment through OpenStack Ironic instead of the default IPA in OSP8. For the latest information, see https://bugzilla.redhat.com/show_bug.cgi?id=1283436 and https://bugzilla.redhat.com/show_bug.cgi?id=1317731.

2. Go to the Red Hat Customer Portal. Download the latest OSP7-based discovery and deployment ramdisk to the images directory in the stack user's home directory (on the director's host /home/stack/images).

   **Note:** In the NetApp lab environment, the discovery ramdisk was discovery-ramdisk-7.3.1-59.tar, and the deployment ramdisk was deploy-ramdisk-ironic-7.3.0-39.tar.

3. Install the rhosp-director-images and rhosp-director-images-ipa packages.

   ```bash
   [stack@osp-director images]$ sudo yum install rhosp-director-images rhosp-director-images-ipa
   ``

   **Note:** In the NetApp lab environment, the overcloud image was distributed as a part of the OSP8 GA. The IPA is required for uploads to Glance, which is detailed later in this section.

4. Copy the new image archives to the images directory on the stack user’s home directory.

   ```bash
   [stack@osp-director images]$ cp /usr/share/rhosp-director-images/overcloud-full-latest-8.0.tar ~/images/
   [stack@osp-director images]$ cp /usr/share/rhosp-director-images/ironic-python-agent-latest-8.0.tar ~/images/
   ```

5. Verify that the images are in the /home/stack/images directory.

   ```bash
   [stack@osp-director images]$ ls -lh
   -rw-r--r-- 1 stack stack 59M  Apr 20 15:34 discovery-ramdisk-7.3.0-39.tar
   -rw-r--r-- 1 stack stack 152M Apr 20 15:31 discovery-ramdisk-7.3.1-59.tar
   -rw-r--r-- 1 stack stack 334M Apr 20 11:45 ironic-python-agent.tar
   -rw-r--r-- 1 stack stack 1.1G Apr 20 15:41 overcloud-full.tar
   ```

6. Extract the images from the archives.

   ```bash
   [stack@osp-director images]$ for tarfile in *.tar; do tar -xf $tarfile; done
   ```

7. Verify that the extraction was successful.

   ```bash
   [stack@osp-director images]$ ls -lh
   -rw-r--r-- 1 stack stack 59M Mar 9 17:10 discovery-ramdisk-7.3.0-39.tar
   -rw-r--r-- 1 stack stack 152M Mar 17 11:15 discovery-ramdisk-7.3.1-59.tar
   -rw-r--r-- 1 stack stack 334M Mar 17 11:53 discovery-ramdisk-initramfs
   -rw-r--r-- 1 stack stack 1.1G Apr 20 15:41 overcloud-full.tar
   ```

8. Proceed to the next section to customize the director and the overcloud-full.qcow2 file to support iSCSI boot LUNs hosted on the NetApp FAS.
Customize Director and Overcloud Images

To customize director and overcloud images, complete the following steps to support iSCSI-backed root disks in the eventual OpenStack deployment:

1. Modify the `/httpboot` directory for the discovery image and the iPXE process so that OpenStack Ironic can discover the boot LUN during introspection and discovery.

2. Modify the Python `ipxe_config.template` source code so that the corrected information for recognizing iSCSI boot LUNs is properly written to the `/tftpboot` directory during introspection and with the deployment image.

3. Customize the `overcloud-full.qcow2` image to support iSCSI and multipath in the initial ramdisk so that the system sees the remote LUN on bootup of RHEL 7.2.

4. Set a password for the root user on the `overcloud-full.qcow2` image for troubleshooting.

5. Install the packages on the director server that is used for customization.

   ```bash
   [stack@osp-director images]$ sudo yum -y install libguestfs-tools libvirt
   ```

6. Restart the `libvirtd` service.

   ```bash
   [stack@osp-director images]$ sudo systemctl restart libvirtd.service
   ```

7. Move the `overcloud-full.qcow2` image to the `/var/lib/libvirt/images` directory.

   ```bash
   [stack@osp-director images]$ sudo mv overcloud-full.qcow2 /var/lib/libvirt/images/
   ```

8. Run the `virt-edit` command to modify the `/etc/sysconfig/grub` file in the `overcloud-full.qcow2` image.

   ```bash
   [stack@osp-director images]$ sudo virt-edit -a /var/lib/libvirt/images/overcloud-full.qcow2 /etc/sysconfig/grub
   ```

9. Append the options `rd.iscsi.firmware=1` and `rd.iscsi.ibft=1` to the end of the `GRUB_CMDLINE_LINUX` line. The NetApp lab environment resembled the following:

   ```bash
   GRUB_TIMEOUT=5
   GRUB_DISTRIBUTOR="\$(sed 's, release .*,g' /etc/system-release)"
   GRUB_DEFAULT=saved
   GRUB_DISABLE_SUBMENU=true
   GRUB_TERMINAL_OUTPUT="console"
   GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8 crashkernel=auto rd.iscsi.firmware=1 rd.iscsi.ibft=1" GRUB_DISABLE_RECOVERY="true"
   ```

10. Save and close the file by using the `:wq` key combination.

11. Run the `virt-edit` command to modify the `/etc/fstab` file in the `overcloud-full.qcow2` image.

   ```bash
   [stack@osp-director images]$ sudo virt-edit -a /var/lib/libvirt/images/overcloud-full.qcow2 /etc/fstab
   ```

12. Append the option `_netdev` to the first uncommented line. The NetApp lab environment resembled the following output:

   ```bash
   # /etc/fstab
   # Created by anaconda on Mon Apr  4 19:29:08 2016
   # Accessible filesystems, by reference, are maintained under '/dev/disk'
   # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
   # LABEL-img-rootfs / xfs defaults,_netdev 0 0
   ```

13. Save and close the file by using the `:wq` key combination.
   
   **Note:** This image was supplied as an exhibit from GitHub and downloaded earlier to the director server. The image is located in the `/home/stack/flexpod-templates/netapp-extra` directory.

```
[stack@osp-director images]$ sudo virt-customize -a /var/lib/libvirt/images/overcloud-full.qcow2 --upload /home/stack/flexpod-templates/netapp-extra/multipath.conf:/etc/
[  0.0] Examining the guest ...
[  3.0] Setting a random seed
[  3.0] Uploading: /home/stack/flexpod-templates/netapp-extra/multipath.conf to /etc/
[  3.0] Finishing off
```

15. Boot the image in rescue mode to recreate the initial ramdisk (used to find the iSCSI LUN during bootup). Add the iSCSI and multipath modules in the initramfs (they are not there by default).
   
   **Note:** If you do not perform this step, the systemd daemon freezes upon bootup.

```
[stack@osp-director images]$ sudo virt-rescue --rw /var/lib/libvirt/images/overcloud-full.qcow2
```

16. While in the rescue shell for the `overcloud-full.qcow2` image, mount `/dev/sda` to the `/sysroot` directory.

```
<rescue> mount /dev/sda /sysroot
```

17. Change root to the newly mounted partition.

```
<rescue> chroot /sysroot
```

18. Regenerate the `initialramfs` image by using Dracut to include necessary modules so that the system can find the remote LUN.

```
<rescue> dracut -f -v -N
```


```
<rescue> exit
<rescue> exit
```

20. Set a root password on the `overcloud-full.qcow2` image to help with troubleshooting.

```
[stack@osp-director images]$ sudo virt-customize -a /var/lib/libvirt/images/overcloud-full.qcow2 --root-password myNewRootPassword
```

21. Move the newly modified image back into the `/home/stack/images` directory on the director server.

```
[stack@osp-director images]$ sudo mv /var/lib/libvirt/images/overcloud-full.qcow2 .
```

**Upload Images to Glance**

To upload the images to Glance, complete the following steps:
1. **Run the** `openstack overcloud image upload` **command.**

2. **Import the images contained in the images directory to the director.**

   **Note:** The `--old-deploy-image` parameter is critical to use the OSP7-based deployment image when the overcloud is deployed. Disregard the warning about the bash-based ramdisk.

   ```bash
   [stack@osp-director images]$ openstack overcloud image upload --image-path /home/stack/images/ --old-deploy-image
   DEPRECATED: The old bash-based ramdisks are no longer supported. You should move to the agent-based ramdisk as soon as possible.
   ```

3. **Verify that the images are contained in the image store.**

   ```bash
   [stack@osp-director images]$ openstack image list
   +---------------------------------+-------------------------+
   | ID                              | Name                    |
   +---------------------------------+-------------------------+
   | 45529dd0-d88c-411d-aeb6-fce0b113de65 | bm-deploy-ramdisk       |
   | 3e03ae8b-a468-4f34-bd33-b379de9dc09d | bm-deploy-kernel        |
   | a87a2493-5d8a-4be6-9e13-b32095069914 | overcloud-full          |
   | 9ca4d072-0c6f-4b1f-95bc-547d61b4da00 | overcloud-full-initrd   |
   | e5068c74-8e76-4bb8-a31d-c4d00a9d0991 | overcloud-full-vmlnuz   |
   +---------------------------------+-------------------------+
   ```

**Modify iPXE Discovery Configuration**

To configure iPXE to use the discovery ramdisk and kernel from OSP7 and to configure the PXE server options while performing introspection to account for the discovered process in OpenStack Ironic, complete the following steps:

**Note:** The IPA does not work during the discovery or introspection phase of the overcloud creation.

1. **Move the IPA-based agent files to a different location so they are not used.**

   ```bash
   [stack@osp-director httpboot]$ cd /httpboot/
   [stack@osp-director httpboot]$ sudo mv agent.kernel agent.kernel.old
   [stack@osp-director httpboot]$ sudo mv agent.ramdisk agent.ramdisk.old
   [stack@osp-director httpboot]$ sudo mv inspector.ipxe inspector.ipxe.old
   ```

2. **Copy the OSP7-based discovery kernel and ramdisk to the original IPA-based locations.**

   ```bash
   [stack@osp-director httpboot]$ cd /home/stack/images/
   [stack@osp-director images]$ sudo cp discovery-ramdisk.kernel /httpboot/agent.kernel
   [stack@osp-director images]$ sudo cp discovery-ramdisk.initramfs /httpboot/agent.ramdisk
   ```

3. **Copy the `inspector.pxe.old` file to `inspector.ipxe` and modify it for your environment. The **`lab inspector.ipxe`** is listed below for reference. You can substitute the IP address of your respective director server and place the following into `/httpboot/inspector.ipxe`:

   ```bash
   #!ipxe
dhcp
discoverd_callback_url=http://172.21.19.18:5050/v1/continue ip=ip;${ip}:${{next-server}};${gateway};${netmask}; BOOTIF=${mac} rd.iscsi.firmware=1
boot
   ```

**Modify iPXE Deployment Configuration**

To modify the iPXE source code to pass options that help find the iSCSI LUN used for the root disk for the server to the bash-based deployment image, complete the following steps:

**Note:** The IPA does not work during the deployment phase of the overcloud creation. Be careful when completing the steps in the section titled “Upload Images to Glance”; make sure to pass the --
old-deploy-image parameter when uploading the deployment kernel and ramdisk to OpenStack Glance.

1. From the director server, open the following file with a text editor:

```
(stack@osp-director ~)$ sudo vim /usr/lib/python2.7/site-packages/ironic/drivers/modules/pxe_config.template
```

2. Insert the `rd.iscsi.firmware=1` to the end of the stanza with the `:deploy` and `:kernel` lines. The NetApp file resembles the following:

```bash
#!/ipxe

dhcp
goto deploy

:deploy
kernel {{ pxe_options.deployment_aki_path }} selinux=0 disk-{{ pxe_options.disk }}
iscsi_target_iqn-{{ pxe_options.iscsi_target_iqn }}
deployment_key-{{ pxe_options.deployment_key }}
ironic_api_url-{{ pxe_options.ironic_api_url }}
troubleshoot-0 text {{ pxe_options.pxe_append_params|default("", true) }}
boot_option-{{ pxe_options.boot_option }}
lp=$(/ip):~/next-server:$gateway:$netmask)
BOOTIP=$(/mac) {% endif %} ipa-api-url-{{ pxe_options[ipa-api-url] }}
ipa-driver-name-{{ pxe_options[ipa-driver-name] }}
boot_mode-{{ pxe_options[boot_mode] }}
initrd-{{ pxe_options.ari_path }}
boot

:boot_partition
kernel {{ pxe_options.aki_path }} root-{{ ROOT }}
initrd {{ pxe_options.ari_path }}
boot

:boot_whole_disk
sanboot --no-describe
```

3. Restart OpenStack ironic services.

```
(stack@osp-director ~)$ sudo systemctl restart openstack-ironic-(api,conductor,inspector-dnsmasq,inspector)
```

Modify TripleO Heat Templates to Accommodate NetApp E-Series

By default, director-installed overcloud deployments use the local root disk under the `/srv/node/d1` directory for OpenStack Swift data placement.

The templates in this document are designed to automatically configure NetApp E-Series as a target for OpenStack Swift data. However, one file must be modified on the director server to accommodate an array of strings passed to it. When configured, Swift uses the configured E-Series LUNs presented to the OpenStack controller systems `/srv/node/eserieslun[0-3]` instead of `/srv/node/d1`.

1. Open the `controller.yaml` file under the `openstack-tripleo-heat-templates/puppet` directory.

```
(stack@osp-director ~)$ sudo vim /usr/share/openstack-tripleo-heat-templates/puppet/controller.yaml
```

2. For the `swift_device` resource, change the template string from `r1z1-IP:%PORT%/d1` to the following:

```yaml
swift_device:
  description: Swift device formatted for swift-ring-builder
  value: str_replace:
```

©2016 NetApp, Inc. All rights reserved.
Updated subnet: 311019e9

dns_nameservers
cid

dirw
dirw
dirw

template: "r121-IP:%PORT%/eserieslun0,r121-IP:%PORT%/eserieslun1,r121-IP:%PORT%/eserieslun2,r121-IP:%PORT%/eserieslun3"
params:
IP: {get_attr: [NetIpMap, net_ip_uri_map], {get_param: [ServiceNetMap, SwiftMgmtNetwork]}}

Note: This template parameter can be modified for the customer’s environment, using more or fewer LUNs as needed. This parameter must match the SwiftNetappEseriesLuns parameter in /home-stack/flexport-templates/flexport.yaml.

3. Save and close the file by using the :wq key combination.

Download NetApp Copy Offload Tool

To download the NetApp copy offload tool, complete the following step:

1. Download the NetApp Toolchest from the NetApp Support site.

   Note: NetApp recommends the Toolchest download for the overcloud deployment. Be sure to download the binary (named na_copyoffload_64 at the time of this writing) and copy it to the /home-stack/postdeploy-flexport-scripts directory on the director server.

   Note: The postdeployment scripts available on the NetApp GitHub site can automatically copy the NetApp copy offload tool to the proper /usr/local/bin directory on OpenStack controller systems that are to be provisioned by the director.

   [stack@osp-director ~]$ ls -lah /home-stack/postdeploy-flexport-scripts/
drwxrwxr-x. 2 stack stack 4.0K Apr 20 20:56 .
drwxrwxr-x. 11 stack stack 4.0K Apr 20 20:55 ..
-rwxrwxr-x. 1 stack stack 1.1K Apr 20 12:13 flexpodupdate-allsystems.sh
-rwxrwxr-x. 1 stack stack 3.7K Apr 20 12:13 flexpodupdate-controllers.sh
-rwxrwxr-x. 1 stack stack 11K Apr 20 12:13 flexpodupdate-start.sh
-rwxr-xr-x. 1 stack stack 365K Apr 20 20:58 na_copyoffload_64

Set Name Server for Overcloud

Overcloud nodes require a name server in order to resolve host names through DNS. The name server is defined in the undercloud’s neutron subnet. To set a name server for the overcloud, complete the following steps:

1. Run the following commands to define the name server for the environment:

   [stack@osp-director ~]$ neutron subnet-list
   +-----------------------------------------+-----------------------------------------+-----------------------------+-----------------------------+
   | id | name | cidr | allocation_pools |
   +-----------------------------------------+-----------------------------------------+-----------------------------+-----------------------------+
   +-----------------------------------------+-----------------------------------------+-----------------------------+-----------------------------+

   [stack@osp-director ~]$ neutron subnet-update 311019e9-bc2c-4739-a8d2-ed6857b814f9 --dns-nameserver 172.21.11.254
   Updated subnet: 311019e9-bc2c-4739-a8d2-ed6857b814f9

2. Verify that the subnet information is updated.

   [stack@osp-director ~]$ neutron subnet-show 311019e9-bc2c-4739-a8d2-ed6857b814f9
   +-----------------------------------------+-----------------------------------------+-----------------------------+-----------------------------+
   | Field | Value |
   +-----------------------------------------+-----------------------------------------+-----------------------------+-----------------------------+
   | allocation_pools | {"start": "172.21.19.21", "end": "172.21.19.200"} |
   | cidr | 172.21.19.0/24 |
   | dns_nameservers | 172.21.11.254 |
### 4.3 Launch Introspection of Servers

The Cisco UCS service profiles that were created earlier must be discovered, acknowledged, and cataloged by the OpenStack Ironic service for an OpenStack deployment. Important characteristics are gathered by the discovery image sent through PXE down to these servers, for example:

- How much memory the server has
- CPU architecture of the server
- Disk size for the OS installation (in this example, RHEL 7.2)
- Number of CPUs the server has
- Number of NICs the server has
- Ironic driver used (in this example, pxe_ipmitool)

#### JSON Attributes

Server definition templates must be created and define how the director adds the seven servers identified in this document to the database that OpenStack Ironic uses. These values are represented in a JavaScript Object Notation (JSON)–format file and contain the hardware and power management details for the seven servers. For example:

```json
{

    "nodes": [
        {
            "mac": [
                "00:00:de:ad:be:af"
            ],
            "cpu": "4",
            "memory": "6144",
            "disk": "40",
            "arch": "x86_64",
            "pm_type": "pxe_ipmitool",
            "pm_user": "myipmiUsername",
            "pm_password": "myipmiPassword",
            "pm_addr": "192.168.70.125"
        }
    ]
}
```

The server definition template uses the JSON attributes listed in Table 20.

#### Table 20) Introspection JSON attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>mac</td>
<td>A list of MAC addresses for the network interfaces on the node. <strong>Note:</strong> Use only the MAC address for the provisioning NIC of each system.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Attribute Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pm_type</td>
<td>The power management driver. Note: The example in this document uses the IPMI driver (pxe_ipmitool).</td>
</tr>
<tr>
<td>pm_user, pm_password</td>
<td>The IPMI user name and password as defined by the IPMI access profile in Cisco UCSM.</td>
</tr>
<tr>
<td>pm_addr</td>
<td>The IP address of the IPMI device as defined by the IPMI access profile in Cisco UCSM.</td>
</tr>
<tr>
<td>cpu</td>
<td>The number of CPUs on the node. Note: Set this value lower than what is actually in the server because the introspection process updates it when discovering facts about the server.</td>
</tr>
<tr>
<td>memory</td>
<td>The amount of memory, in megabytes (MB). Note: Set this value lower than what is actually in the server because the introspection process updates it when discovering facts about the server.</td>
</tr>
<tr>
<td>disk</td>
<td>The size of the hard disk, in gigabytes (GB). Note: Set this value lower than what is actually in the server because the introspection process updates it when discovering facts about the server.</td>
</tr>
<tr>
<td>arch</td>
<td>The system architecture. Note: The system architecture example in this document is x86_64.</td>
</tr>
</tbody>
</table>

**Customize and Import JSON File to Director**

The `blades-ipmi.json` file is supplied as an exhibit from GitHub. This file was downloaded to the director earlier in the procedure and is located in the `/home/stack/` directory. To customize and import the JSON file to the director, complete the following steps:

1. Customized the `./home/stack/blades-ipmi.json` file to match your environment and verify it in Cisco UCS Manager.

2. Import the file into the director.

   ```bash
   [stack@osp-director ~]$ openstack baremetal import --json ~/blades-ipmi.json
   ```

3. Verify that all seven servers successfully imported into the OpenStack Ironic database.

   ```bash
   [stack@osp-director ~]$ ironic node-list
   +--------------------------------------+
   | UUID                                 | Name | Instance UUID | Power State | Provisioning State |
   | Maintenance |                                  |               |             |                    |
   +--------------------------------------+
   | bfd9cf63-4e1c-43e7-94a2-7aca1ef1e2b4 | None | None          | power off   | available          |
   | False       |                                  |               |             |                    |
   | 9fbfd72a-dc00-4547-a8d5-cdb29d2f1fbf | None | None          | power off   | available          |
   | False       |                                  |               |             |                    |
   | de4a61ce-a0af-46e6-b8ef-3e2916e79094 | None | None          | power off   | available          |
   | False       |                                  |               |             |                    |
   ```
4. Assign the kernel and ramdisk images to all nodes.

```bash
[stack@osp-director ~]$ openstack baremetal configure boot
```

**Launch Introspection of Server Hardware**

After registering the nodes, the hardware attributes of each server must be inspected. To launch introspection of the server hardware, complete the following steps:

1. Inspect the hardware attributes of each server.

```bash
[stack@osp-director ~]$ openstack baremetal introspection bulk start
```

Waiting for introspection to finish...

**Note:** Do not interrupt the inspection process; it can take a several minutes to complete. This process took approximately seven minutes in the NetApp lab.

This example shows the introspection of one of the Cisco UCS server consoles.

2. Look for the following output on the director to verify that the introspection completed successfully.

```
Introspection for UUID de4a61ce-a0af-46e6-b8ef-3e2916e79094 finished successfully.
```
Introspection for UUID 8b7cb1ab-69df-4c33-9793-c33eab82f941 finished successfully.
Introspection for UUID 995c2e79-9677-41d8-84a5-79b590d98fbf finished successfully.
Introspection for UUID bdf9cf63-4e1c-43e7-94a2-7acalef1e2b4 finished successfully.
Introspection for UUID 447f6d28-b822-4823-bd21-ee3ef0bc75e0 finished successfully.
Introspection for UUID c11f95d-c764-467b-997b-0899bea16d4a finished successfully.

Setting manageable nodes to available...
Node bfd9cf63-4e1c-43e7-94a2-7acalef1e2b4 has been set to available.
Node 9fbfd72a-dc00-4547-84a5-79b590d98fbf finished successfully.
Node de4a61ce-a9af-46ee-b8ef-3e2916e79094 has been set to available.
Node 995c2e79-9677-41d8-84a5-79b590d98fbf has been set to available.
Node c11f95d-c764-467b-997b-0899bea16d4a has been set to available.
Node 8b7cb1ab-69df-4c33-9793-c33eab82f941 has been set to available.
Node 447f6d28-b822-4823-bd21-ee3ef0bc75e0 has been set to available.
Introspection completed.

3. Run the following commands on one of the nodes in the ironic node-list output.

   Note: Make sure that memory_mb, local_gb, and cpus reflect what is installed on the server.

```bash
[stack@osp-director ~]$ ironic node-show bfd9cf63-4e1c-43e7-94a2-7acalef1e2b4
```

```bash
+----------------------------------------+------------------------+
| Property            | Value                  |
+------------------------+------------------------+
| target_power_state    | None                   |
| extra                 | {u'hardware_swift_object': u'extra_hardware-bfd9cf63-4e1c-43e7-94a2-7acalef1e2b4'} |
| last_error            | None                   |
| updated_at            | 2016-03-30T01:50:48+00:00 |
| maintenance_reason    | None                   |
| provision_state       | available              |
| clean_step            | {}                     |
| uuid                  | bfd9cf63-4e1c-43e7-94a2-7acalef1e2b4 |
| console_enabled       | False                  |
| target_provision_state| None                   |
| provision_updated_at  | 2016-03-30T01:50:48+00:00 |
| maintenance           | False                  |
| inspection_started_at | None                   |
| inspection_finished_at| None                   |
| power_state           | power off              |
| driver                | pxe_ipmitool           |
| reservation           | None                   |
| properties            | {u'memory_mb': u'262144', u'cpu_arch': u'X86_64', u'local_gb': u'49', |
|                       | u'cpus': u'32', u'capabilities': u'boot_option:local'} |
| instance_uuid         | None                   |
```
Rename Servers in OpenStack Ironic

To rename the servers in the OpenStack Ironic database, complete the following steps:

1. Assign a friendly system name to the nodes in OpenStack Ironic.

   `[stack@osp-director ~]$ ironic node-update bfd9cf63-4e1c-43e7-94a2-7aca1ef1e2b4 replace name=controller1`

2. Repeat step 1 for each additional node. The OpenStack Ironic node list output command is as follows:

   ```
   [stack@osp-director ~]$ ironic node-list
   +--------------------------------------+-+----------------+-----------------+----------------+-----------------------------+
   | UUID | Name        | Instance UUID | Power State | Provisioning |
   +--------------------------------------+-+----------------+-----------------+----------------+-----------------------------+
   | bfd9cf63-4e1c-43e7-94a2-7aca1ef1e2b4 | controller1  | None          | power off     | available       |
   | False | 9fbd72a-da00-4547-a8d5-cdb292df1fbf | controller2  | None          | power off     | available       |
   | False | de4a61ce-a0af-46e6-b8ef-3e2916e79094 | controller3 | None          | power off     | available       |
   | False | 995c2e79-9677-41d8-84a5-79b590d98bfc | compute1    | None          | power off     | available       |
   | False | cc11f95d-c764-467b-997b-0899bea16d6a | compute2    | None          | power off     | available       |
   | False | 8b7c61ab-69df-4c33-9793-c33eab82f941 | compute3    | None          | power off     | available       |
   | False | 447f6d28-bb22-4823-bd21-ee3ef0bc75e0 | compute4    | None          | power off     | available       |
   +--------------------------------------+-+----------------+-----------------+----------------+-----------------------------+
   ```

Associate Servers in OpenStack Ironic with Controller or Compute Role

Tag the discovered or introspected nodes with either the controller or compute role in OpenStack Ironic. As a result, the node is associated with a specific role when the overcloud is deployed.

**Note:** For the examples in this document, three servers were chosen as OpenStack controller systems, and four servers were chosen as OpenStack compute systems.

To associate servers in OpenStack Ironic with either the controller or compute role, complete the following step:
1. Run the following commands:

```bash
[stack@osp-director ~]$ ironic node-update controller1 add properties/capabilities="profile:control,boot_option:local"
[stack@osp-director ~]$ ironic node-update controller2 add properties/capabilities="profile:control,boot_option:local"
[stack@osp-director ~]$ ironic node-update controller3 add properties/capabilities="profile:control,boot_option:local"
[stack@osp-director ~]$ ironic node-update compute1 add properties/capabilities="profile:compute,boot_option:local"
[stack@osp-director ~]$ ironic node-update compute2 add properties/capabilities="profile:compute,boot_option:local"
[stack@osp-director ~]$ ironic node-update compute3 add properties/capabilities="profile:compute,boot_option:local"
```

4.4 Deploy Overcloud

The final step in creating the OpenStack environment is to deploy the overcloud. Before deploying the overcloud, verify the following prerequisites:

- The FlexPod HOT templates are modified for the customer’s environment. These modified templates help administrators take advantage of customized templates and automation specifically written for this technical report to enable the FlexPod value proposition in the resulting OpenStack deployment.
- The FlexPod postdeployment scripts are modified for the customer’s environment. These scripts perform tasks that are not suited as HOT templates. They are scripts that launch after the overcloud is deployed.

Modify FlexPod HOT Templates

The `flexpod.yaml` file is available on GitHub. It should be located in the `/home/stack/flexpod-templates` directory after running the commands in the section titled “Download and Configure FlexPod Heat Templates and Postdeployment Scripts.”

**Note:** The various file parameters are located in the section titled “FlexPod Templates.”

To modify the FlexPod HOT templates to suit the customer’s environment, complete the following steps:

1. Verify that you are the `stack` user in the stack user’s home directory.
   
   **Note:** The `stackrc` file must be sourced as a part of your profile.

   ```bash
   [root@osp-director ~]# su - stack; cd /home/stack; source stackrc
   ```

2. Open the `flexpod.yaml` file under the `/home/stack/flexpod-templates/` directory.

   ```bash
   [stack@osp-director ~]# vim flexpod-templates/flexpod.yaml
   ```

3. Modify the following variables in the `flexpod.yaml` file:
   - `CinderNetappLogin` is the cluster administrator account name, which is typically `admin`.
   - `CinderNetappPassword` is the password for the cluster administrator user used by the NetApp unified Cinder driver.
   - `CinderNetappServerHostname` is the IP address or host name of the cluster admin LIF.
   - `CinderNetappServerPort` is the port used to communicate with the cluster admin LIF, either 80 or 443.
   - `CinderNetappVserver` is the SVM used for Cinder in the resulting OpenStack deployment.
   - `CinderNetappNfsShares` is the FlexVol volumes used as an NFS standpoint. This notation is `IP:/export`, where `IP =` the IP address and `export =` the NFS export path.
   - `GlanceNetappCopyOffloadMount` is the IP address and mount point for the NFS export used by Glance. The `GlanceNetappCopyOffloadMount` variable is used by the NetApp copy offload tool to quickly clone images to volumes in the resulting OpenStack deployment. This variable is typically the same as the `GlanceFilePcmkDevice` variable.
   - `SwiftNetappEseriesHicP1` is the IP address of the NetApp E-Series controller HIC1, port 1.
   - `SwiftNetappEseriesLuns` is the space-delimited LUN numbers that reflect the LUNs used for OpenStack Swift by the NetApp E-Series storage system.
CloudDomain is the DNS domain name of the overcloud servers.

GlanceFilePcmkDevice the FlexVol volume IP address and mount point for Glance.

4. Save and close the flexpod.yaml file by using the :wq key combination.
   
   **Note:** If you are following the guidance presented in this document, leave the rest of the variables in this file alone.

5. Open the network-environment.yaml file under the /home/stack/flexpod-templates/network-environment.yaml directory.

   ```bash
   [stack@osp-director ~]$ vim flexpod-templates/network-environment.yaml
   ```

6. Modify any of the predefined variables in this file to suit your environment.
   
   **Note:** This file mirrors the previously configured VLAN and subnet information described in the section titled “Necessary VLANS.”

7. Save and close the network-environment.yaml file by using the :wq key combination.
   
   **Note:** If you are following the guidance presented in this document, leave the rest of the variables in this file alone.

8. Open the controller.yaml file under the /home/stack/flexpod-templates/nic-configs/controller.yaml directory.

   ```bash
   [stack@osp-director ~]$ vim flexpod-templates/nic-configs/controller.yaml
   ```

9. Modify any of the predefined variables in this file to suit your environment, specifically if you want to configure a different NIC segmentation on the controller servers in the overcloud.
   
   **Note:** Pay particular attention to the resources: section in this file. This file takes advantage of the previously configured vNIC information described in the section titled “Cisco UCS Setup.”

10. Open the compute.yaml file under the /home/stack/flexpod-templates/nic-configs/compute.yaml directory.

    ```bash
    [stack@osp-director ~]$ vim flexpod-templates/nic-configs/compute.yaml
    ```

11. Modify any of the predefined variables in this file to suit your environment, specifically if you want to configure a different NIC segmentation on the compute servers in the overcloud.
   
   **Note:** Pay particular attention to the resources: section in this file. This file takes advantage of the previously configured vNIC information described in the section titled “Cisco UCS Setup.”

12. Save and close the file by using the :wq key combination.

After you modify the files identified in this section, you should not need to modify any other files.

**Deploy Overcloud**

To deploy the overcloud, complete the following steps:

1. Verify that the flexpod.yaml, network-environment.yaml, compute.yaml, and controller.yaml files were successfully modified.

2. Create and deploy the overcloud by running the following command:

   ```bash
   ```

   Where terms have the following meanings:

   - **--templates.** Creates the overcloud and uses the OpenStack Heat template collection location in the /usr/share/openstack-tripleo-heat-templates directory on the director server.
- -e /home/stack/flexpod-templates/network-environment.yaml. An environment file that represents the customized networking, subnet, and VLAN information consumed in the overcloud.
- -e /home/stack/flexpod-templates/flexpod.yaml. An environment file that serves as a main template for modifying the overcloud for FlexPod enhancements in OpenStack. Several child templates that are executed at dedicated stages during the overcloud deployment (called from flexpod.yaml). They use the NodeExtraConfig, ControllerExtraConfigPre, and NodeExtraConfigPost resource registries. NetApp unified Cinder driver, NetApp E-Series for OpenStack Swift, NetApp copy offload tool, and extra paths for DM-Multipath for the resulting servers in the overcloud are configured automatically.
- --control-scale. The number of controller systems that are configured during the deployment process.
- --compute-scale. The number of compute systems that are configured during the deployment process.
- --ntp-server. The NTP server used by the servers in the overcloud.
- --neutron-network-type. The network segmentation used for OpenStack Neutron.
- --neutron-tunnel-types. The tunneling mechanism used in the OpenStack Neutron deployment process.
- -t. The time allotted for a successful deployment. One hour should be sufficient.

3. The following message is displayed, indicating a successful deployment process:

```
Overcloud Endpoint: http://172.21.11.176:5000/v2.0
Overcloud Deployed
clean_up DeployOvercloud:
END return value: 0
```

**Note:** The deployment process took approximately 35 minutes in the NetApp lab.

**Note:** The overcloud endpoint in this example provides access to Horizon. Credentials for the dashboard are located in the /home/stack/overcloudrc file on the director host. To access Horizon, omit :5000/v2.0 from the endpoint.

Launch FlexPod Postdeployment Scripts

Postdeployment scripts created by NetApp are available on GitHub. They should be in your /home/stack/postdeploy-flexpod-scripts/ directory after running the commands listed in the section titled “Download and Configure FlexPod Heat Templates and Postdeployment Scripts.”

These scripts help deploy OpenStack Manila in the resulting overcloud environment.

**Note:** The various file parameters are located in the section titled “Postdeployment Scripts (Non-Heat).”

To modify and launch the FlexPod postdeployment scripts, complete the following steps:

1. Verify that you are the stack user in the stack user’s home directory.

   **Note:** The stackrc file must be sourced as a part of your profile.

   ```
   [root@osp-director ~]# su - stack; cd /home/stack; source stackrc
   ```

2. Open the flexpodupdate-controllers.sh file under the /home/stack/postdeploy-flexpod-scripts directory.

   ```
   [stack@osp-director ~]# vim postdeploy-flexpod-scripts/flexpodupdate-controllers.sh
   ```

3. Modify the following variables in the flexpodupdate-controllers.sh file:
   - NETAPP_CLUSTERADMIN_LIF is the IP address or host name of the cluster admin LIF.
- **NETAPP_CLUSTERADMIN_USER** is the cluster administrator account name, which is typically admin.

- **NETAPP_CLUSTERADMIN_PASS** is the password for the cluster administrator user used by the NetApp unified Cinder driver.

- **NETAPP_MANILA_SVM** is the SVM used for Manila in the resulting OpenStack deployment.

4. Save and close the file by using the :wq key combination.

5. Make sure that the NetApp copy offload tool is in the `/home/stack/postdeploy-flexpod-scripts` directory. If it is not, reread the section titled “Download NetApp Copy Offload Tool.”

```
[stack@osp-director ~]$ ls -lah postdeploy-flexpod-scripts/
total 400K
drwxrwxr-x. 2 stack stack 4.0K Apr 20 21:55 ..
drwx------. 12 stack stack 4.0K Apr 20 21:55 ..
-rwxrwxr-x. 1 stack stack 1.4K Apr 20 00:14 flexpodupdate-allsystems.sh
-rwxrwxr-x. 1 stack stack 3.7K Apr 20 12:13 flexpodupdate-cont0.sh
-rwxrwxr-x. 1 stack stack 1.1K Apr 20 08:36 flexpodupdate-controllers.sh
-rwxrwxr-x. 1 stack stack 2.4K Apr 20 12:13 flexpodupdate-start.sh
-rwx-r-x-. 1 stack stack 365K Apr 20 20:58 na_copyoffload_64
```


```
[stack@osp-director ~]$ cd postdeploy-flexpod-scripts/
```

7. Launch the FlexPod postdeployment script.

```
[stack@osp-director postdeploy-flexpod-scripts]$ ./flexpodupdate-start.sh
```

8. Monitor the output from the script.

**Note:** OpenStack Manila services should be created inside the overcloud.

```
+ openstack role add --project service --user manila admin
+ openstack service create --name manila --description 'Manila File Share Service' share
+ openstack service create --name manilav2 --description 'Manila File Share Service V2' sharev2
```

9. Continue monitoring output from the script.

**Note:** OpenStack Manila endpoints should be created in Keystone inside the overcloud.

```
```
10. Continue monitoring output from the script.

Note: Pacemaker resource records for OpenStack Manila API, scheduler, and share service should be created inside the overcloud.

+ pcs resource create openstack-manila-api systemd:openstack-manila-api -clone interleave=true
+ pcs resource create openstack-manila-scheduler systemd:openstack-manila-scheduler -clone interleave=true
+ pcs resource create openstack-manila-share systemd:openstack-manila-share
+ pcs constraint order start openstack-manila-api-clone then openstack-manila-scheduler-clone
+ pcs constraint order start openstack-manila-api-clone then openstack-manila-scheduler-clone
+ pcs constraint colocation add openstack-manila-scheduler-clone with openstack-manila-api-clone
+ pcs constraint order start openstack-manila-scheduler-clone then openstack-manila-share
+ pcs constraint colocation add openstack-manila-scheduler-clone with openstack-manila-share

11. The FlexPod postdeployment is complete.

5 Solution Operations: Functionality Verification

The functionality of the overcloud environment must now be verified.

5.1 Log in to Horizon

To log in to Horizon, complete the following steps:

1. Verify that you are the stack user in the stack user's home directory.

   Note: The overcloudrc file must be sourced as a part of your profile.

   [root@osp-director -]# su - stack; cd /home/stack; source stackrc

2. Make a note of the password for the admin user in the overcloud.

   [stack@osp-director -]$ cat overcloudrc | grep OS_PASSWORD
   export OS_PASSWORD="BBuYN9bwfWZE9uebDsus7pR3"
3. If the overcloud message is truncated, run the following command to discover the Horizon dashboard URL.

```
(stack@osp-director ~)$ openstack endpoint show horizon
+----------------------------------++----------------------------------+
| Field       | Value                                               |
+----------------------------------++----------------------------------+
| adminurl    | http://172.21.11.176:80/dashboard/admin             |
| enabled     | True                                                |
| id          | 3b37780e1b9344e6adde5014215ca104                    |
| internalurl | http://172.21.11.176:80/dashboard/                  |
| publicurl   | http://172.21.11.176:80/dashboard/                  |
| region      | regionOne                                           |
| service_id  | 9ec70d0862384d04a2c341276af9887c                    |
| service_name| horizon                                             |
| service_type| dashboard                                           |
+----------------------------------++----------------------------------+
```

4. Grab the `publicurl` value and open a web browser.

5. Log in to Horizon with the `admin` user name and the password noted in step 2.

5.2 Upload Image to Image Store

The OpenStack image service called Glance provides discovery, registration, and delivery services for disk and server images.

To upload an image to the OpenStack image store, complete the following steps:
1. On Horizon, select the Admin tab and click Images.

2. On the Images page, click Create Image.

3. Enter Fedora23_Cloud as the image name.

4. For the image location, go to the Fedora website and search for a 64-bit base cloud image in qcow2 format. Copy the link and paste it into the Image Location field.

5. In the Format column, select QCOW2.

6. Make sure that the Public option is selected so that the image can be used by different tenants.

7. Click Create Image at the bottom of the page. The new image should now be displayed in the image store.
5.3 Create Flavors

In OpenStack, virtual hardware templates are called flavors. They define the RAM sizes, disk, number of cores, and so on. To create flavors, complete the following steps:

1. On Horizon, select the Admin tab and then click Flavors.
2. Click Create Flavor.
3. For the flavor name, enter m0.toaster.
4. For the VCPUs value, enter 1.
5. For the RAM value, enter 512.
6. For the root disk (GB) value, enter 40GB.
7. Click Create Flavor at the bottom of the page. The new flavor should now be displayed.

5.4 Create Project and User

In OpenStack, a group of users is referred to as a project or tenant. These terms are interchangeable. Users must be associated with at least one project; however, they can belong to many. You must add a project before adding users. To create a project and users, complete the following steps:

1. On Horizon, select the Identity tab and then click Create Project.
2. For the project name, enter netapp.
3. Click Create Project at the bottom of the page.

4. From the Identity tab, click Users.

5. Click Create User.

6. For the project username, enter netapp.

7. For the password, enter a strong password of your liking and then confirm the password.

8. From the Primary Project drop-down menu, select netapp.

9. Click Create User at the bottom of the page. The new user netapp should now be created.

10. From the Admin drop-down menu, select Sign Out.

5.5 Create Tenant Network and Router

The tenant network provides internal network access for instances. The architecture isolates this type of network from other tenants. A router must be added for outbound access so instances on this network can communicate to hosts outside of the network. To create a tenant network and router, complete the following steps:
1. Log in to Horizon using the netapp user login credentials.
2. Select the Network tab and click Networks.

3. Click Create Network.
4. For the network name, enter default.
5. Select the Subnet tab.
6. For the subnet name, enter default_subnet.
7. For the network address, enter 10.10.10.0/20.
8. For the gateway IP address, enter 10.10.10.1.
9. Select the Subnet Details tab.
10. For the allocation pools, enter 10.10.10.5, 10.10.15.254.
11. For the DNS name servers, enter 172.21.11.254.
12. Click Create on the bottom of the page.
13. Select the Network tab and click Routers.
14. Click Create Router.
15. For the router name, enter `router1`.
16. Click Create Router at the bottom of the page.
17. After the router is created, click the name `router1`.
18. Select the Interfaces tab.
19. Click Add Interface.
20. From the Subnet drop-down menu, select the default subnet.
21. Click Add Interface at the bottom of the page.
22. An internal interface is added to `router1`.

23. Log out of the `netapp` user account by selecting Sign Out from the NetApp drop-down menu.

5.6 Set Gateway and Create Floating IP Network and Subnet

To set external connectivity for any tenant router, administrator rights are required. However, a floating IP address network must first be set up with Horizon using the administrator login credentials.

1. Click Networks.
2. Click Create Network.
3. For the network name, enter `floating`.
4. For the project, select `admin`.
5. For the provider network type, select `VLAN`.
6. For the physical network, enter `datacentre` (the proper spelling is important).
7. For the segmentation ID (otherwise known as the 802.1Q VLAN ID), enter `3270`.

**Note:** This value corresponds with the public VLAN illustrated in the section titled “Necessary VLANs.”
8. Make sure that the External Network option is selected.
9. Click Create Network at the bottom of the page. After the floating network is created, the following network information is displayed:

![Networks](image)

10. In the Network Name column, click Floating.
11. Click Create Subnet.
12. For the subnet name, enter floating_subnet.
   
   **Note:** This network address corresponds with the public VLAN illustrated in the section titled “Necessary VLANs.”
15. Select the Subnet Details tab.
16. Deselect the Enable DHCP option.
18. Click Create at the bottom of the page. After the floating subnet is created, the network overview information is displayed.
19. Log out by selecting Sign Out from the Admin drop-down menu.

After the floating network and subnet are created, the netapp tenant can set a gateway as its own router created earlier, router1.

1. Log in to Horizon by using the netapp user credentials.
2. Select the Network tab and click Routers.
3. Under the Actions column, click Set Gateway.
4. From the External Network drop-down menu, select Floating.
5. Click Set Gateway at the bottom of the page.
6. The floating network is displayed in the External Network column.

5.7 Create and Boot Instances from Volume

To create and boot instances from a volume, complete the following prerequisites:

- **Create a security group.** Security groups are a set of network traffic filtering rules that are applied to compute instances.
- **Create a key pair for the netapp user.** When an instance is launched, inject a key pair, which provides SSH access to the instance.

To create a security group, complete the following steps:

2. Click Create Security Group.
3. For the security name, enter `netapp-secgroup`.
4. After the security group is created, click Manage Rules.
5. Click Add Rule.
6. From the Rule drop-down menu, select ALL ICMP.
7. Click Add at the bottom of the page.
8. Click Add Rule again.
9. From the Rule drop-down menu, select SSH.
10. Click Add at the bottom of the page. After the security group is created, the following security information is displayed.

To create a key pair, complete the following steps:
2. Click Key Pairs.
3. Click Create Key Pair.
4. For key pair name, enter `netapp-key`.
5. Click Save Key Pair at the bottom of the page.
6. Save the key pair on your local machine for later use. Click OK.
To boot instances from a volume, complete the following steps:

1. Select the Compute tab and click Instances.
2. Click Launch Instance.
3. For the instance name, enter system1.
4. From the Flavor drop-down menu, select m0.toaster.
5. From the Instance Boot Source drop-down menu, select Boot from Image (creates a new volume).
6. For the image name, select Fedora23_Cloud.
7. Select the Delete on Terminate option to automatically clean up the volume in case the instance is deleted.
8. From the Access & Security tab, verify that the netapp-secgroup security group option is selected.
9. From the Networking tab, verify that the default network option is selected for NIC1.
10. Click Launch at the bottom of the page. After the instance is created, the following instance information is displayed.
5.8 Associate Floating IP Address with Instance

Floating IP addresses are associated with instances, in addition to their fixed IP addresses. Unlike fixed IP addresses, the floating IP address associations can be modified at any time regardless of the state of the instances involved. They allow incoming traffic to reach instances using one-to-one IP address mapping between the floating IP address and the IP address actually assigned to the instance in the tenant network.

Before you assign an IP address to an instance, allocate the floating IP address to a NetApp project. To allocate a floating IP address, complete the following steps:

1. From the Compute tab, click Access & Security.
2. From the Floating IPs tab, click Allocate IP To Project.
3. Click Allocate IP at the bottom of the page. After the floating IP address is allocated to the NetApp project, the following IP information is displayed.

To allocate a floating IP address to the system1 instance, complete the following steps:

1. From the Compute tab, click Instances.
2. Under the Actions column, select Associate Floating IP from the drop-down menu.
4. Click the associate.
5. After the floating IP address is associated to the system1 instance, the following instance information is displayed:
5.9 Verify Inbound and Outbound Network Traffic to Instance

You can verify network connectivity to the system1 instance in the following manner:

- For inbound connectivity, the instance can be pinged from outside of the overcloud environment by using the floating IP address (from the previous section). You can obtain SSH access directly to the instance by using the key pair that was downloaded to the client in a previous section by the same floating IP address.
- For outbound connectivity, the instance can forward traffic to its own default gateway on the tenant subnet (10.10.10.1) and have it sent through the overcloud infrastructure. You can connect the instance to a Manila-provisioned share in a later step.

**Note:** Client traffic from the tenant subnet can reach the Internet through an outbound NAT in the NetApp lab.

To verify inbound connectivity from the client, complete the following steps:

1. Ping the floating IP address associated with the system1 instance.
   
   **Note:** In this example, the myclientmachine system is outside of the infrastructure and is communicating directly with the instance through OpenStack Neutron and the physical data center network infrastructure (Cisco Nexus 9000 pair).

   ```bash
   [dcain@myclientmachine ~]$ ping 172.21.14.105 -c 4
   64 bytes from 172.21.14.105: icmp_seq=1 ttl=62 time=48.1 ms
   64 bytes from 172.21.14.105: icmp_seq=2 ttl=62 time=48.5 ms
   64 bytes from 172.21.14.105: icmp_seq=3 ttl=62 time=53.0 ms
   64 bytes from 172.21.14.105: icmp_seq=4 ttl=62 time=46.3 ms
   --- 172.21.14.105 ping statistics ---
   4 packets transmitted, 4 received, 0% packet loss, time 3004ms
   rtt min/avg/max/mdev = 46.304/48.992/53.013/2.483 ms
   ```

2. Before you obtain SSH access to the instance, secure the permissions of the key file that was downloaded earlier.

   ```bash
   [dcain@myclientmachine ~]$ chmod 600 Downloads/netapp-key.pem
   ```

3. Obtain SSH access to the instance by using the key file downloaded from Horizon earlier (saved in the Downloads folder).
Note: For the Fedora 23 cloud image, obtain SSH access to the user as `fedora`.

```
[dcain@myclientmachine ~]$ ssh -i Downloads/netapp-key.pem fedora@172.21.14.105
```

4. Verify both of the IP addresses and the default route on the eth0 interface in the instance.

```
[fedora@system1 ~]$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1400
    inet 10.10.10.8 netmask 255.255.192.0 broadcast 10.10.63.255
    inet6 fe80::f816:3eff:fe55:a9ac prefixlen 64 scopeid 0x20<link>
      ether fa:16:3e:55:a9:ac txqueuelen 1000 (Ethernet)
    RX packets 285 bytes 34003 (33.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 365 bytes 36862 (35.9 KiB)
    TX errors 0 dropped 0 carrier 0 collisions 0

[fedora@system1 ~]$ ip route
default via 10.10.10.1 dev eth0
10.10.0.0/18 dev eth0 proto kernel scope link src 10.10.10.8
169.254.169.254 via 10.10.10.1 dev eth0 proto static
```

5. Install system updates in the system1 instance.

Note: Outbound RPM updates are available on the Internet.

```
[fedora@system1 ~]$ sudo dnf -y update
Fedora 23 - x86_64 - Updates
5.4 MB/s | 21 MB 00:03
Fedora 23 - x86_64
11 MB/s | 43 MB 00:03
Last metadata expiration check performed 0:00:18 ago on Fri Apr 22 02:46:27 2016.
Dependencies resolved.
```

<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Installing:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kernel-core</td>
<td>x86_64</td>
<td>4.4.6-300.fc23</td>
</tr>
<tr>
<td>updates</td>
<td>20 M</td>
<td></td>
</tr>
</tbody>
</table>

```
```

5.10 Provision and Access Share Using File Share Service (Manila)

To provision a file share, first define the share type as the admin user. Then log in as the netapp user and create a share that is accessible to the system1 instance created earlier.

Create Share Type

To create a share type, complete the following steps:

1. Log in to Horizon as the admin user.
2. From the System tab, click Shares.
3. Click Share Types.
4. Click Create Share Type.
5. For the share name, enter general.
6. Click Create Share Type at the bottom of the page. After the share type is created, the following share information is displayed:
7. Log out of the admin user account by selecting Sign Out from the Admin drop-down menu.

**Request Share**

After the share type is created, request a share by completing the following steps:

1. Log in to Horizon with the `netapp` user login credentials.
2. From the Compute tab, click Shares.
3. Click Create Share.
4. For the share name, enter `share1`.
5. For size, enter 7GB.
6. From the Share Type drop-down menu, select General.
7. From the Availability Zone drop-down menu, select Nova.
8. Click Create Share at the bottom of the page. After the share is created, the following information is displayed:
9. From the clustered Data ONTAP CLI, the share is displayed in the osp8-svm.

```bash
fas8040-openstack::> volume show -vserver osp8-svm

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Volume</th>
<th>Aggregate</th>
<th>State</th>
<th>Type</th>
<th>Size</th>
<th>Available</th>
<th>Used%</th>
</tr>
</thead>
<tbody>
<tr>
<td>osp8-svm</td>
<td>share_887374bf_af97_4192_9a89_14d5a6ae59be</td>
<td>aggr01_node02</td>
<td>online</td>
<td>RW</td>
<td>7GB</td>
<td>6.65GB</td>
<td>5%</td>
</tr>
</tbody>
</table>
```

**Define Share Rules**

Defining share rules allows you to dictate which clients have access to the newly created share. By default, access to the share is a deny condition; therefore, access must be whitelisted through an access policy before client machines (such as system1) can access it. The NetApp Manila driver configures the NetApp FAS8040's export policy to whitelist access for the system1 machine through NetApp Manageability SDK calls.

The NetApp environment uses VXLAN network segmentation at the tenant level. It is deencapsulated at the neutron router layer and sent to the NetApp FAS8040 device through the public subnet (172.21.14.0/24). Therefore, a specific rule must be added to permit access for the floating IP address of system1 (172.21.14.105). As a result, the instance can successfully mount the share directly from the CLI through standard Linux mechanisms.

After a share is created, add an access rule by completing the following steps:

1. From the Actions drop-down menu, select Manage Rules.
2. Click Add Rule.
3. For access to, enter 172.21.14.105, which is the floating IP address assigned to the system1 instance.
4. Verify that the access rule was created on the FAS8040 by using the clustered Data ONTAP CLI.

```bash
fas8040-openstack::> export-policy rule show -vserver osp8-svm -policyname policy_887374bf_af97_4192_9a89_14d5a6ae59be

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Policy</th>
<th>Rule</th>
<th>Access</th>
<th>Client</th>
<th>RO</th>
</tr>
</thead>
<tbody>
<tr>
<td>osp8-svm</td>
<td>policy_887374bf_af97_4192_9a89_14d5a6ae59be</td>
<td>1</td>
<td>any</td>
<td>172.21.14.105</td>
<td>sys</td>
</tr>
</tbody>
</table>
```

5. Verify the share1 share. From the Compute tab, click Shares.
6. Click share1 and review the new share overview.
To verify that the system1 instance can mount the share named share1, complete the following steps:

1. Log back into the system1 instance by using the floating IP address from the client system named myclientmachine.

```
[dcain@myclientmachine ~]$ ssh -i Downloads/netapp-key.pem fedora@172.21.14.105
Last login: Fri Apr  1 20:54:51 2016 from 10.254.66.10
[fedora@system1 ~]$ 
```

2. Create a directory to serve as the mount point for the share.

```
[root@system1 ~]$ mkdir /mnt/share1/
```

3. Mount the share.

```
[root@system1 /]$ mount -o nfsvers=3 192.168.67.1:/share_887374bf_af97_4192_9a89_14d5a6ae59be /mnt/share1
```

4. Verify the mount’s free space and availability.

```
[root@system1 /]$ df -h
Filesystem Size Used Avail Use% Mounted on
devtmpfs 233M 0 233M 0% /dev
```

Share Details: share1

Share Overview

- **Info**
  - **Name**: share1
  - **ID**: abe73be-145f-4f2f-bc1e-60c6e7b1b105
  - **Status**: Available
  - **Export locations**:
    - 192.168.67.2:/share_887374bf_af97_4192_9a89_14d5a6ae59be
    - 192.168.67.1:/share_887374bf_af97_4192_9a89_14d5a6ae59be
  - **Visibility**:
    - Private
    - Availability zone: nova
  - **Specs**:
    - **Size**: 7.08
    - **Protocol**: NFS
    - **Share type**: generic
  - **Created**: April 1, 2016, 6:19 p.m.
  - **Host**: overcloud-controller-0.localdomain@netappstorage01_node02
  - **Access Rules**:
    - **ip**
      - 172.21.14.105
5. Create a file called `bigfile` and copy it to the share.

```
[root@system1 ~]# cd /root;
[root@system1 ~]# dd if=/dev/urandom of=bigfile bs=1M count=100
100+0 records in
100+0 records out
10485760 bytes (105 MB) copied, 6.47864 s, 16.2 MB/s
```

```
[root@system1 ~]# cp bigfile /mnt/myshare
```

6. Verify that `bigfile` was successfully copied to the share.

```
[root@system1 ~]# ls -lh /mnt/myshare/
total 101M
-rw-r--r-- 1 root root 100M Apr 22 2016 bigfile
```

### 5.11 Upload to and Access Objects in Object Store (Swift)

To upload files to the object store as a test and verify that the previous steps were implemented successfully, complete the following steps on the director server:

1. Verify that you are the stack user in the stack users home directory, with the `overcloudrc` file sourced as a part of your profile.

```
[root@osp-director ~]# su - stack; cd /home/stack; source overcloudrc
```

2. Run the following commands to upload three random objects to the c1 container.

```
[stack@osp-director tmp]$ cd /tmp
[stack@osp-director tmp]$ head -c 1024 /dev/urandom > data1.random; swift upload c1 data1.random
[stack@osp-director tmp]$ head -c 1024 /dev/urandom > data2.random; swift upload c1 data2.random
[stack@osp-director tmp]$ head -c 1024 /dev/urandom > data3.random; swift upload c1 data3.random
```

3. Verify that the objects were added to the c1 container.

```
[stack@osp-director tmp]$ swift list c1
data1.random
data2.random
data3.random
```

4. To verify that the same information is present in Horizon, log back in to the dashboard using the admin user credentials.

5. Select the Project tab at the top of the dashboard (to the right of the Red Hat OpenStack Platform banner).

6. Select the Object Store tab and click Containers.

7. Click the c1 container. The three objects uploaded in previous steps should be displayed.

**Note:** Any user (including the admin user) can upload objects.
6 Solution Operations: Load Testing

The OpenStack deployment can have a series of automated tests run against it to make sure that the control plane is functioning properly. Early detection of functional and/or performance degradations should be key factors of the change management processes through continuous monitoring of infrastructure and cloud resources. Rally is an automated toolset that can be deployed in tandem with the resulting OpenStack cloud.

User requests (RESTful API calls, interaction through the OpenStack dashboard, other custom tooling, and so on) can be bursty in nature to the control plane. There might be a period of constant requests associated with a workflow, or there might be situations in which larger Glance image sizes (20GB or more) are required. The turnaround time for the storage requests plays a significant role in creating a positive user experience and meeting established SLAs.

The goals for demonstrating this automated test suite in the resulting OpenStack deployment are as follows:

- Can the infrastructure successfully stand up to constant requests for instances (VMs) using an image in Glance, such as a Fedora 23 Cloud image?
- How much faster can we spin up instances utilizing the NetApp NFS Cinder driver versus the Generic NFS Cinder driver on the same infrastructure?
- How much physical disk space can we save using the NetApp NFS Cinder driver versus the Generic NFS Cinder driver on the same infrastructure?
- What about an even larger-sized image, such as a Fedora 23 cloud image filled with 35GB of random data? How is time and space utilization affected by using the NetApp NFS Cinder driver versus the Generic NFS Cinder driver on the same infrastructure?

OpenStack Rally can be used to answer these questions and demonstrate why NetApp storage for OpenStack is compelling in terms of time-efficient operations and space savings on the storage itself.

6.1 OpenStack Rally

OpenStack Rally is a benchmark-as-a-service (BaaS) project for OpenStack. It is a tool that automates and unifies multinode OpenStack cloud verification, benchmarking, and profiling. Rally can be used to
continuously improve cloud operating conditions, performance, stability through infrastructure upgrades, and so on.

Rally is written in Python and uses relational databases such as MySQL, PostgreSQL, or SQLite to store the results of test runs. It contains predefined tasks that in most cases can be used as-is to benchmark or demonstrate capabilities or atomic actions in the resulting deployment.

**Note:** For more information, see the Rally documentation.

**Note:** Step-by-step instructions to install Rally are outside of the scope of this document. For detailed steps, see the Rally installation and configuration steps.

### Load-Testing Scenarios

To test the OpenStack cloud performance under specific load conditions, the following load-testing scenarios were performed in the NetApp lab environment:

- **Scenario 1.** Subject the control plane to a constant load of 35 requests concurrently by requesting that 2,000 persistent instances (VMs) be booted from volume.
- **Scenario 2.** Request instances from a large image (60GB RAW image).

You can go back and forth between the NetApp NFS Cinder driver and the Generic NFS Cinder driver on the same infrastructure and measure the results using OpenStack Rally.

The goal of these testing scenarios is to prove the efficiency of using NetApp storage paired with the NetApp NFS Cinder driver.

### Initial Prerequisites and Configuration

One of NetApp's goals in using OpenStack Rally is to establish conditions that are present and common across the scenarios in order to have a fair baseline for comparison when launching those tasks.

NetApp used three controllers with four compute nodes and the backend NFS shares listed in Table 21.

#### Table 21) NFS shares used by Rally.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS Shares</td>
<td>192.168.67.1:/cinder1</td>
</tr>
<tr>
<td></td>
<td>192.168.67.2:/cinder2</td>
</tr>
<tr>
<td></td>
<td>192.168.67.1:/cinder3</td>
</tr>
</tbody>
</table>

**Note:** Storage efficiency is not enabled on the backend.

Since reliable connections to MariaDB are essential to perform an OpenStack operation, NetApp increased the maximum allowable connections, as shown in Table 22.

#### Table 22) Configuration changes required on controller systems.

<table>
<thead>
<tr>
<th>File</th>
<th>Parameter and Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/my.cnf.d/galera.cnf</td>
<td>max_connections = 10240</td>
</tr>
</tbody>
</table>

NetApp created the following entities before launching Rally in the existing overcloud OpenStack environment, as shown in Table 23.

#### Table 23) Overcloud configuration addition required.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flavor</td>
<td>m0.toaster(RAM = 512 MB, vCPU = 1)</td>
</tr>
</tbody>
</table>
Note: The `Fedora_data.raw` image is 60GB and is filled with 35GB of random data to simulate a larger than normal image.

### 6.2 Scenario Definitions and Tasks

After the necessary configuration changes were made to support Rally, NetApp defined the scenarios that needed to be run and explained what Rally would do in the resulting OpenStack deployment.

#### Scenario 1: Control Plane Load and 2,000 Persistent Instances Creation

NetApp used the existing Nova scenario, `NovaServers.boot_server_from_volume_and_delete`, that is defined in Rally. Scenario 1 consisted of the following operations:

- **Atomic operations:**
  - Created a bootable Cinder volume using an image defined in Glance.
  - Booted the instance from the Cinder volume previously created.
  - Deleted the instance and its associated volume.

- **Context set up:**
  - Created a tenant and a user associated with the tenant.
  - Created a neutron network in the tenant and associated a subnet with the network.
  - Set appropriate quotas for the tenant (which supported the atomic actions in the task).

- **Scenario operations:**
  - Created 2,000 bootable Cinder volumes from the Fedora 23 Cloud image.
  - Booted 2,000 instances from the Cinder volumes.
  - Deleted 2,000 instances.

In order to gauge the performance of the NetApp NFS Cinder driver in load conditions, NetApp set the concurrency parameter to 35, which translated to 35 concurrent requests against the OpenStack control plane while the scenario was executed. In other words, 35 requests were active throughout the entire run.

The individual task file for Scenario 1 is defined as follows:

```json
{
    "NovaServers.boot_server_from_volume_and_delete": {
        "args": {
            "flavor": {
                "name": "m0.toaster"
            },
            "min_sleep": 2,
            "max_sleep": 4,
            "image": {
                "name": "Fedora23_Cloud"
            },
            "volume_size": 60,
            "auto_assign_nic": true
        },
        "runner": {
            "type": "constant",
            "times": 2000,
            "concurrency": 35
        },
        "context": {
            "users": {
                "tenants": 1,
                "users_per_tenant": 1
            }
        }
    }
}
```
"network": {
    "start_cidr":"10.0.0.0/16",
    "networks_per_tenant": 1,
    "subnets_per_network": 1
},
"quotas":{
    "nova":{
        "instances": -1,
        "cores": -1,
        "ram": -1,
    },
    "neutron":{
        "network": -1,
        "port": -1,
        "subnet": -1,
    },
    "cinder":{
        "gigabytes": -1,
        "volumes": -1,
    }
}
}

Note: NetApp assumed that the flavor m0.toaster and image Fedora23_Cloud already existed.
NetApp ran the task with the NetApp NFS Cinder driver and the Generic NFS Cinder driver. It finished with a success rate of 100% for both drivers.

Results
The following parameters reflect the performance of both drivers:

- **Total time taken to create volume.** Since the load conditions were maintained in each case with fixed concurrency, the total time taken to create the 2,000 bootable volumes would reflect the behavior of each system under the load.

- **Total amount of space consumed.** The amount of space consumed on the backend shares for 2,000 bootable volumes.

Table 24 summarizes the readings for creating and booting 2,000 persistently backed instances.

<table>
<thead>
<tr>
<th>Type of Driver</th>
<th>Total Time Taken</th>
<th>Total Space Consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp NFS Cinder driver</td>
<td>1,187 seconds</td>
<td>42.90GB</td>
</tr>
<tr>
<td>Generic NFS Cinder driver</td>
<td>4,102 seconds</td>
<td>1209.49GB</td>
</tr>
</tbody>
</table>

Note: The total time also includes the instance boot time, but we found this to be independent of the Cinder driver being used at the time.

The NetApp NFS Cinder driver achieved space efficiency by creating a point-in-time, space-efficient, writable copy of the parent volume.
Figure 21 compares the total time it took to create 2,000 bootable volumes with 35 concurrent active requests. As shown in Figure 21, the NetApp NFS Cinder driver completed the task in 1,187 seconds while the Generic NFS Cinder driver completed the task 4,102 seconds.

**Summary:** The NetApp NFS Cinder driver creates volumes 71.06% faster than the Generic NFS Cinder driver.
Figure 22 compares the total disk space consumed by 2,000 bootable volumes. When the NetApp NFS Cinder driver was used, the total amount of disk space consumed was 42.9GB, but when the Generic NFS Cinder driver was used, the total amount of disk space consumed was 1209.49GB.

**Summary:** The NetApp NFS Cinder driver uses 99.97% less physical disk space than the Generic NFS Cinder driver.

**Scenario 2: Volume Creation with Large Image Size**

NetApp used the same Nova scenario as the previous test, *NovaServers.boot_from_volume_and_delete*, but changed the following parameters:

- Used a RAW image file. The RAW image was based on the Fedora 23 Cloud image, except that it had 35GB of randomized data inserted into the image. The image disk size was 60GB.
- Set the concurrency set to one. Since the load was applied in terms the image size, NetApp was not as concerned in benchmarking a 35 concurrency.

Scenario 2 set up a similar context and performed similar operations as Scenario 1; however, Scenario 2 profiled the differences only using the larger RAW image and not the load on the OpenStack control plane.

The individual task file Scenario 2 is as follows:

```json
{
    "NovaServers.boot_server_from_volume_and_delete": {
        "args": {
            "flavor": {
                "size": 35,
                "name": "fedora-23-cloud-image-with-35GB-data"}
        }
    }
}```
Results

The following parameters reflect the performance of the NetApp NFS Cinder driver:

- **Individual time taken to create a volume.** The concurrency was set to one; therefore, each request received a similar timeshare, which resulted in close readings for each iteration.
- **Total space consumed.** The amount of space that was consumed on the storage to host the volumes.

NetApp observed that the median, 90% ile, 95% ile and the average readings for the creation of the volume are almost similar, which reflects no anomalies in the control plane during respective runs. 95% ile reflects the behavior of the largest subset of the population; therefore, NetApp selected it as an indicator for individual time taken to create a volume, as shown in Table 25.
Table 25) Scenario 2 results.

<table>
<thead>
<tr>
<th>Type of Driver</th>
<th>Individual Time Taken</th>
<th>Total Space Consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp NFS Cinder driver</td>
<td>32.88 seconds</td>
<td>87GB</td>
</tr>
<tr>
<td>Generic NFS Cinder driver</td>
<td>743.52 seconds</td>
<td>6000GB</td>
</tr>
</tbody>
</table>

Figure 23) Comparison of time taken to create a single bootable volume in Scenario 2.

Figure 23 compares the time it took to create a single bootable volume from a 60GB RAW image. The NetApp NFS Cinder driver took 32.88 seconds to create a bootable volume while the Generic NFS Cinder driver took 743.52 seconds.

The NetApp NFS Cinder driver creates volumes 95.57% faster than the Generic NFS Cinder driver.

---

4 We observed that when using the Generic NFS Cinder driver that there were failures associated with copies of the image (since it is a large image). Out of 100 runs, 56 runs were successful. The amount of space consumed was 3436GB, therefore each Cinder volume took approximately 60GB. With these empirical readings we can extrapolate the data to 100 runs which would consume a total of 6000GB.
Figure 24 compares the total disk space consumed by 100 bootable volumes created from 60GB RAW images. The volumes created by using the NetApp NFS Cinder driver consumed 87GB of disk space while the volumes created by using the generic NFS Cinder driver consumed 6000GB of disk space.

**Summary**: Even with using large RAW images, the NetApp NFS Cinder driver used 98.55% less physical disk space than the Generic NFS Cinder driver.

### 7 Solution Maintenance

The undercloud and overcloud must be updated over time.

#### 7.1 Update Undercloud

You can upgrade any package that is installed on the undercloud machine. To update the undercloud, complete the following step:

1. Run the `yum` command.

```
[stack@osp-director ~]$ sudo yum -y update
```

**Note**: It is not necessary to restart services after the update.

#### 7.2 Update Overcloud

The overcloud relies on standard RPM methods to update the environment. This process involves updating all nodes using the `openstack overcloud update` from the director.
Running an update on all nodes in parallel might cause problems. For example, an update of a package might involve restarting a service, which can disrupt other nodes. For this reason, each node is updated using a set of breakpoints; they are updated one by one. When one node completes a package update, the update process moves to the next node.

The update process also requires the -i option. This option puts the command in interactive mode, which requires confirmation at each breakpoint. Without the -i option, the update process pauses at the first breakpoint. If this happens, you can resume the update process by running the command again with the -i option.

**Note:** Updating the overcloud node packages requires an update using the local Heat template collection.

To update the overcloud stack, complete the following steps:

1. Run the following command:

   ```
   ```

2. Review the update progress by using the CLI.

   ```
   starting package update on stack overcloud
   IN_PROGRESS
   IN_PROGRESS
   IN_PROGRESS
   ```

### 8 Conclusion

With enterprise organizations processing billions of I/O instructions each day, every microsecond counts. FlexPod converged infrastructure solutions represent the latest generation of high-performance servers, networking switches, and storage arrays, bringing the sustained and consistent performance required by next-generation platforms such as OpenStack.

This document demonstrates how operators and administrators can more easily deploy a functional and highly available OpenStack cloud infrastructure using NetApp storage, Cisco UCS compute, Cisco Nexus networking, and the Red Hat OpenStack Platform software.

The goal of this document is to illustrate how these technologies can be used to complete the following tasks:

- Set up network, storage, and server infrastructure
- Deploy the Red Hat OpenStack Platform director
- Deploy Red Hat OpenStack Platform 8 on FlexPod (using director)
- Validate the resulting deployment

This reference architecture details each topic so that users can reproduce them in their own environments. The configuration described across various tools and technologies can be customized and expanded to meet specific requirements.

For cloud architects and managers evaluating an OpenStack-powered infrastructure, FlexPod significantly removes configuration and deployment complexity, eliminates lock-in risk, and accelerates time to value for the business. By capturing the value of robust technologies from Red Hat, Cisco, and NetApp, enterprises stand to benefit from a significantly more agile, efficient, automated, and predictable cloud infrastructure.
Acknowledgments

Dave Cain of NetApp is the primary writer and technical lead in the creation of this document. He would like to thank and acknowledge the following persons, who had a positive impact on the quality of this document:

- Amit Borulkar, Technical Marketing Engineer, NetApp
- Sean Cohen, Principal Product Manager, Red Hat
- Steve Hardy, Principal Software Engineer, Red Hat
- Mike Orazi, Software Engineering Manager, Red Hat
- Steve Reichard, Consulting Engineer and Manager, Red Hat

References

The resources in this section complement the material presented in this document and serve as additional reference sources.

FlexPod

The following links provide additional information about FlexPod:

- TR-4036: FlexPod Datacenter Technical Specifications
- FlexPod Landing Page
  http://flexpod.com/
- FlexPod Datacenter
- FlexPod Validated Designs
- FlexPod Red Hat OpenStack 8 Technical Report GitHub Collateral
  https://github.com/NetApp/snippets/tree/master/RedHat/osp8-liberty/tr

NetApp FAS Storage

The following links provide additional information about NetApp FAS storage:

- Clustered Data ONTAP 8.3 Documentation
- Clustered Data ONTAP 8.3 High-Availability Configuration Guide
  https://library.netapp.com/ecm/ecm_download_file/ECMP1610209
- TR-3982: NetApp Clustered Data ONTAP 8.3.x and 8.2.x
- TR-4067: Clustered Data ONTAP NFS Best Practice and Implementation Guide
- TR-4063: Parallel Network File System Configuration and Best Practices for Clustered Data ONTAP 8.2 and Later
- TR-4379: Name Services Best Practices Guide for Clustered Data ONTAP
- TR-4393: Clustered Data ONTAP Security Guidance
NetApp E-Series Storage
The following links provide additional information about NetApp E-Series storage:

- E5600 Series Documentation
- SANtricity Storage Manager 11.25 Express Guide: For Linux and iSCSI
  https://library.netapp.com/ecm/ecm_download_file/ECMP12409284
- SANtricity Storage Manager 11.25: Multipath Drivers Guide
  https://library.netapp.com/ecm/ecm_download_file/ECMP12404601
- TR-4494: Introduction to NetApp E-Series E5600 with SANtricity 11.25

Cisco UCS
The following links provide additional information about Cisco UCS:

- Cisco Unified Computing System Overview
- Cisco Unified Computing System Technical References
- Cisco UCS 6200 Series Fabric Interconnects
- Cisco UCS 5100 Series Blade Server Chassis
- Cisco UCS B-Series Blade Servers
- Cisco UCS Adapters
- Cisco UCS Manager

Cisco Nexus Networking
The following links provide additional information about Cisco Nexus 9000 Series switches:

- Cisco Nexus 9000 Series Switches
- Cisco Nexus 9000 Series Configuration Guides

Red Hat Enterprise Linux 7
The following links provide additional information about Red Hat Enterprise Linux 7:

- Red Hat Enterprise Linux 7 Product Documentation
  https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/
- DM Multipath Administration and Configuration
  https://access.redhat.com/documentation/en-
Red Hat OpenStack Platform 8

The following links provide additional information about Red Hat OpenStack Platform 8:

- Red Hat OpenStack Platform
  https://access.redhat.com/products/red-hat-enterprise-linux-openstack-platform
- Red Hat OpenStack Platform 8 Documentation Home Page
- Red Hat OpenStack Platform Director Installation and Usage
- Red Hat OpenStack Platform Life Cycle
  https://access.redhat.com/support/policy/updates/openstack/platform/
- Red Hat OpenStack Platform Director Life Cycle
  https://access.redhat.com/support/policy/updates/openstack/platform/director

OpenStack at NetApp

For more information about OpenStack at NetApp, the following resources are available:

- OpenStack at NetApp Landing Page
  http://netapp.github.io/openstack-deploy-ops-guide/
- OpenStack Deployment and Operations Guide for Liberty
  http://netapp.github.io/openstack-deploy-ops-guide/liberty/
- OpenStack at NetApp Blog
  http://netapp.github.io/openstack/

OpenStack Upstream

- OpenStack Documentation for the Liberty Release
  http://docs.openstack.org/liberty/

Hardware and Software Certification

For hardware and software certifications with respect to running OpenStack on FlexPod, see the following resources:

- Cisco UCS Hardware and Software Interoperability Matrix
- NetApp Interoperability Matrix Tool
  http://mysupport.netapp.com/matrix/
- Red Hat Certified Products
  https://access.redhat.com/ecosystem/
- Red Hat Component, Plug-In, and Driver Support in Red Hat OpenStack Platform
  https://access.redhat.com/articles/1535373

Version History

The latest version of this document always located at the following URL:
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Document Version History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0</td>
<td>April 2016</td>
<td>Initial release with NetApp clustered Data ONTAP 8.3.2, NetApp SANtricity OS 8.25.04.00, Cisco NX-OS 7.0(3)I2(2a), Cisco UCS Manager 3.1(1e), Red Hat Enterprise Linux 7.2, and Red Hat OpenStack Platform 8.0.</td>
</tr>
</tbody>
</table>
Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer’s installation in accordance with published specifications.

Copyright Information

Copyright © 1994–2016 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, Go Further, Faster, AltaVault, ASUP, AutoSupport, Campaign Express, Cloud ONTAP, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Flash Accel, Flash Cache, Flash Pool, FlashRay, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, GetSuccessful, LockVault, Manage ONTAP, Mars, MetroCluster, MultiStore, NetApp Fitness, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANshare, SANtricity, SecureShare, Simplicity, Simulate ONTAP, SnapCenter, SnapCopy, Snap Creator, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, SolidFire, StorageGRID, Tech OnTap, Unbound Cloud, WAFL, and other names are trademarks or registered trademarks of NetApp Inc., in the United States and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web at http://www.netapp.com/us/legal/netapptmlist.aspx. TR-4506-0416