NetApp Verified Architecture

Converged Infrastructure Solution with NetApp E-Series and Splunk

NVA Design and Deployment

Dave Derry, NetApp
Mitch Blackburn, NetApp

September 2017 | NVA-1115-DESIGN | Version 1.0

Abstract

This solution provides architectural guidance for inclusion of converged infrastructure techniques in a NetApp® E-Series deployment to support the Splunk analytics application.

NetApp®

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

The purpose of this solution is to provide architectural guidance and sample deployment steps for the inclusion of converged infrastructure techniques in a NetApp E-Series deployment to support the Splunk analytics application.

Note: This document does not cover component sizing or solution performance. Excellent documents and tools that cover those topics already exist. This document focuses on how to connect the various components (NetApp E-Series storage, Splunk forwarding servers, Splunk indexing peer servers, and Splunk search head servers) using converged infrastructure techniques.

2 Program Summary

The term converged infrastructure originally referred to data center technologies that allowed storage traffic to be carried as a payload over a data network. Those technologies (primarily iSCSI and FCoE) were developed in the mid-2000s to improve the return on investment of data center capital expenditures by eliminating the need for a dedicated, Fibre Channel-based storage network.

Because of its beneficial economics, converged infrastructure has become a very popular data center architecture paradigm in the years since its introduction. Over those years, various best practices have emerged for design and deployment of converged infrastructure architectures; many of those best practices are now considered to be under the overall rubric of converged infrastructure. The converged infrastructure best practices that are relevant to this solution include:

- Redundant components: for example, two network switches, two storage controllers, and so on.
- High-availability configuration: for example, configuration such that load handling automatically fails over to the remaining redundant partner in response to a component outage.
- Redundant power distribution.
- Advanced link management technologies: ALUA multipathing, link aggregation (LACP), and jumbo frames should be used in the data paths (where supported); settings should be uniform throughout the network.
- Short storage paths: Hosts and storage controllers should have layer 2 adjacency to minimize network-related storage latency. FCoE enforces that inherently; however, iSCSI does not, so it is up to the designer to make sure that each iSCSI fabric is composed of one dedicated VLAN and its corresponding IP subnet.
- Separate out-of-band management network.

Note that there is a lot of flexibility in this definition; that is deliberate to accommodate the diverse needs of data center operators. The best outcomes occur when a specific deployment is designed by an architect who is familiar with both converged infrastructure best practices and the specific needs of the applications that are running in the data center and implemented on data center-grade equipment.

NetApp E-Series enables Splunk environments to maintain the highest levels of performance and uptime for Splunk workloads by providing advanced fault recovery features and easy in-service growth capabilities to meet ever-changing business requirements. The E-Series is designed to handle the most extreme application workloads with very low latency. Typical use cases include application acceleration; improving the response time of latency-sensitive applications; and improving the power, environmental, and capacity efficiency of overprovisioned environments. E-Series storage systems leverage the latest solid-state disk (SSD) and SAS drive technologies and are built on a long heritage of serving diverse workloads to provide superior business value and enterprise-class reliability. For more information, see the NetApp E-Series Solutions for Splunk page.

Splunk is the leading operational intelligence software that enables you to monitor, report, and analyze live streaming and historical machine-generated data, whether it is located on the premises or in the cloud. An organization's IT data is a definitive source of intelligence because it is a categorical record of

activity and behavior, including user transactions, customer behavior, machine behavior, security threats, and fraudulent activity. Splunk helps users gain visibility into this machine data to improve service levels, reduce IT operations costs, mitigate security risks, enable compliance, and create new product and service offerings. Splunk offers solutions for IT operations, applications management, security and compliance, business analytics, and industrial data. The purpose of this solution is to provide architectural guidance for inclusion of converged infrastructure techniques in an E-Series deployment for Splunk.

3 Solution Overview

This document discusses the following four storage topologies:

- iSCSI network-attached topology
- FC network-attached topology
- FC direct-attached topology
- SAS direct-attached topology

Note that a variety of different components were selected to illustrate the topologies. That does not mean that the illustrated topology only works on these components. Rather, it is to illustrate some of the many possible ways to reach the desired outcome. Different components can be used as needed to accommodate the organization's goals, polices, and preferences.

3.1 Target Audience

The target audience for the solution includes the following groups:

- Solution architects
- Deployment engineers

3.2 Solution Technology

Splunk Architecture Overview

Splunk's architecture provides linear scalability for indexing and distributed search. Splunk's implementation of MapReduce allows large-scale search, reporting, and alerting. Splunk takes a single search and enables you to query many indexers in massive parallel clusters. With the addition of index replication, you can specify how many copies of the data you want to make available to meet your availability requirements.

The Splunk platform is open and has SDKs and APIs, including a REST API and SDKs for Python, Java, JavaScript, PHP, Ruby, and C#. This capability enables developers to programmatically interface with the Splunk platform. With Splunk you can develop your own applications or templates to deploy on your infrastructure.

Splunk can write the indexed data to clustered servers to add additional copies of the raw file and metadata using replication so that the data is available even from indexer node failures.

Before getting into the details about various storage topologies, here is a brief overview of the Splunk components. Figure 1 shows how the following Splunk components interact:

- **Peer nodes.** Also referred to as indexers, the peer nodes receive and index the incoming data, send or receive replicated data in the cluster, and search across indexed data in response to search requests from the search head. There are typically at least three peer nodes in a production cluster.
- **Search head.** The search head manages searches by distributing queries to the peer nodes and then consolidating the results for transmittal back to the requesting application. A cluster must have at least one search head.

- **Master node.** The master node performs management tasks for the cluster. It coordinates replication between the peer nodes and initiates remedial action when a peer node goes offline. It also advises the search heads on which peer nodes to direct their searches to. There is only one master node in a cluster.
- **Forwarders.** Forwarders consume data from external sources and forward it to the peer nodes.

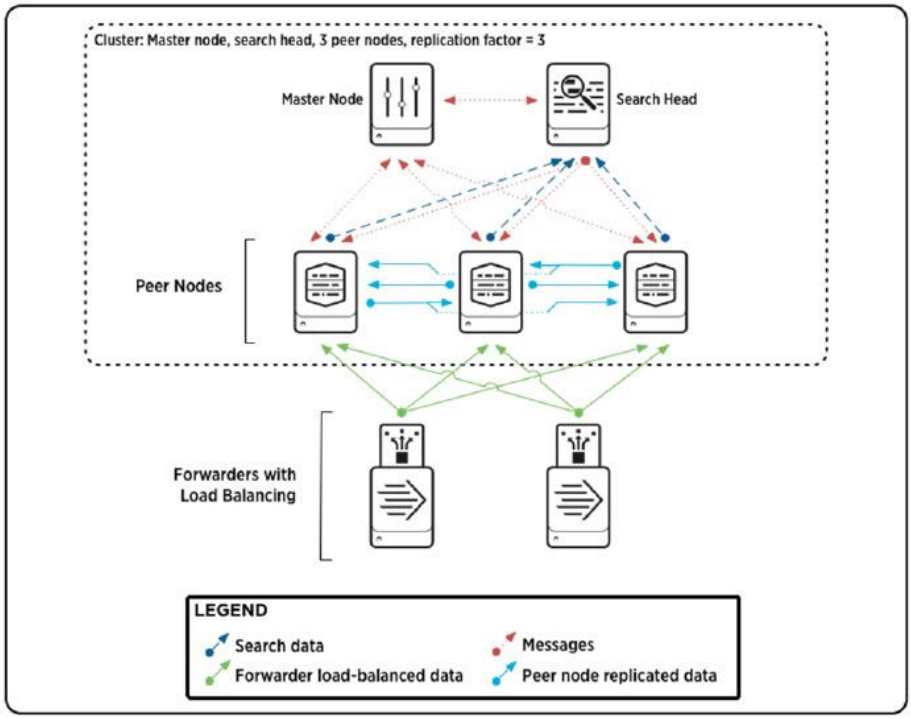
These Splunk components communicate with each other using TCP/IP over Ethernet, so there is always a redundant pair of Ethernet switches in a converged infrastructure Splunk cluster.

When using Splunk clustering to manage server redundancy, the servers do not need to boot from SAN; a local HDD or SSD, of at least 100GB capacity, suffices for booting.

The indexers are the only Splunk component that communicates with the NetApp E-Series storage. That is where the four different storage topologies come from. They correspond to various ways to connect the block protocols that the NetApp E-Series supports.

Note: For this environment, the master node and the forwarder are on the same physical server. In a larger cluster, the master node would be on its own physical server.

Figure 1) Splunk cluster components.



For a complete description of the Splunk architecture, see the Splunk Enterprise Documentation.

The machine log data from the Splunk forwarders sent to the indexer peer nodes uses the recommended data replication factor of three, which makes available three copies of data. The ingested data is compressed and indexed as raw data files and metadata, which are then distributed among the indexer peer nodes for redundancy.

Splunk places your indexed data in directories, also referred to as buckets, as it moves through its lifecycle in Splunk. When data is first indexed, it goes in db-hot. Then, according to your data policy definitions, it moves into the warm buckets, then cold, and finally frozen (which by default means it is deleted). Each time db-hot is rolled to warm, it creates a new directory (known as a warm bucket) named to indicate the time range of the events in that bucket. Rolling to warm occurs automatically when the

specified bucket size is reached, so the buckets are all typically the same size unless you have rolled manually at some point. In this configuration, the hot/warm tier uses a volume created on the SSDs. Each time the warm bucket is rolled to the cold bucket, it creates a new directory on the E-Series volume containing the cold tier. In this case the cold tier is created using the NL-SAS drives. For more information about how buckets work in Splunk, see Understanding how "buckets" work.

NetApp E-Series Overview

NetApp E-Series E2800 storage systems address wide-ranging data storage requirements with balanced performance that is equally adept at handling large sequential I/O for video, analytical, and backup applications, as well as small random I/O requirements for small and medium-sized enterprise mixed workloads. The E2800 brings together the following advantages:

- Support for all-flash and hybrid drive configurations
- Modular host interface flexibility (SAS, FC, and iSCSI)
- High reliability (99.999% reliability)
- Intuitive management: simple administration for IT generalists and detailed drill-down for storage specialists

E2800 System Options

As shown in Table 1, the E2800 is available in three shelf options, which support both hard-disk drives (HDDs) and SSDs, to meet a wide range of performance and application requirements.

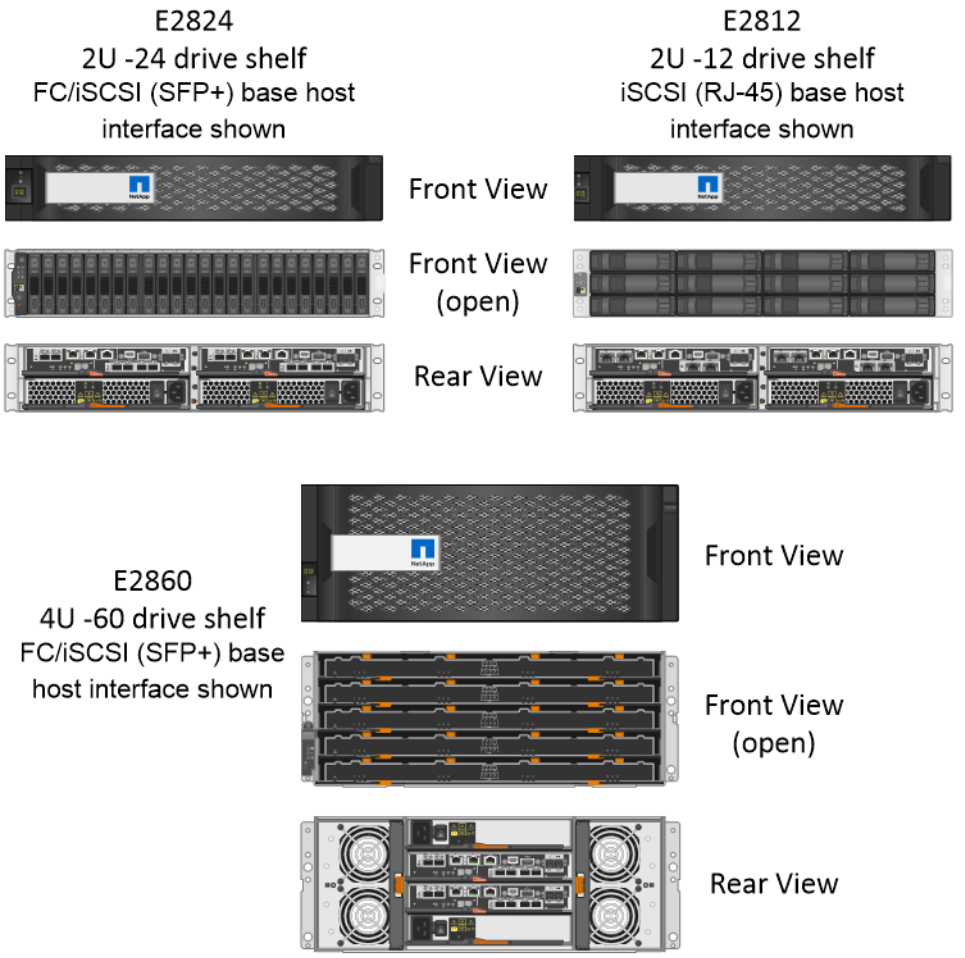
Table 1) E2800 controller shelf and drive shelf models.

Controller Shelf Model	Drive Shelf Model	Number of Drives	Type of Drives
E2812	DE212C	12	3.5" NL-SAS drives 2.5" SAS SSDs
E2824	DE224C	24	2.5" SAS drives (HDDs and SSDs)
E2860	DE460C	60	3.5" NL-SAS drives 2.5" SAS drives (HDDs and SSDs)

The E2812 and E2824 shelf options support one or two controller canisters, while the E2860 supports only two controller canisters. All shelves support dual power supplies and dual fan units for redundancy (the shelves have an integrated power fan canister). The shelves are sized to hold 12 drives, 24 drives, or 60 drives, as shown in Figure 2.

Note: In a duplex configuration, both controllers must be identically configured.

Figure 2) E2800 shelf options (duplex configurations shown).



Each E2800 controller provides two Ethernet management ports for out-of-band management and has two 12Gbps (x4 lanes) wide-port SAS drive expansion ports for redundant drive expansion paths. The E2800 controllers also include two built-in host ports, either two 16Gb FC/10Gb iSCSI or two 10Gb iSCSI RJ-45, but one of the following host interface cards (HICs) can be installed in each controller:

- 4-port 12Gb SAS (SAS 3 connector)
- 2-port 12Gb SAS (SAS 3 connector)
- 4-port optical HIC (SFP+), which can be configured as either 16Gb Fibre Channel or 10Gb iSCSI
- 2-port optical HIC (SFP+), which can be configured as either 16Gb Fibre Channel or 10Gb iSCSI

  Note: A software feature pack can be applied in the field to change the host protocol of the optical baseboard ports and the optical HIC ports from FC to iSCSI or from iSCSI to FC.

- 2-port 10Gb iSCSI (Cat6e/Cat7 RJ-45)

Decoupling Storage from Compute

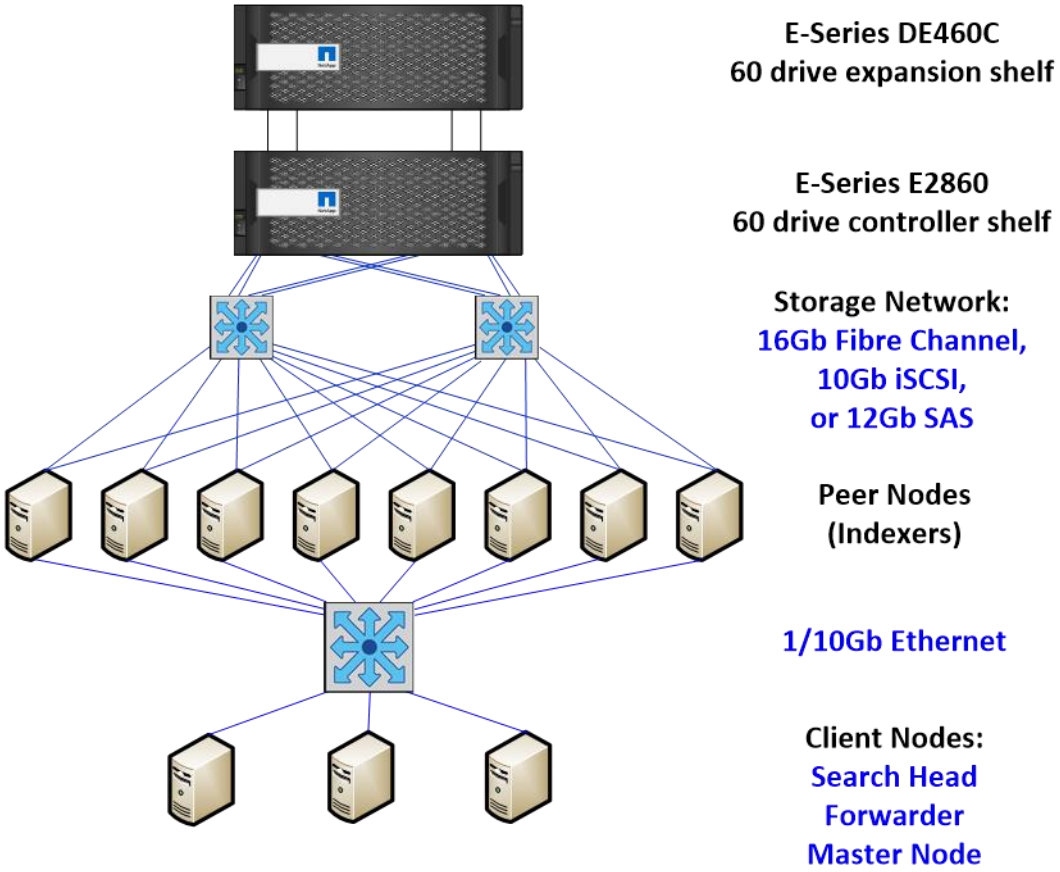
Splunk ingest rates are increasing day by day, retention periods are being extended, SSDs are getting larger, and the use of specialized compute for data analytics is expanding. The ability to decouple storage from compute with Splunk is becoming an economic necessity.

Figure 3 shows a sample Splunk architecture where an E-Series E2860 is connected over iSCSI to eight peer nodes, and the client nodes (search head, forwarder, and master node) are connected over Ethernet to the peer nodes. With this architecture, it is possible to begin with 20 SSDs and scale all the way up to 120 SSDs without needing to add more peer nodes in an all-SSD system. If HDDs are being deployed for the cold tier, it is possible to scale up to 180 drives by adding additional drive shelves to the system shown.

Decoupling storage from compute includes the following advantages:

- Ability to scale capacity and compute separately, saving the cost of overprovisioning one or the other
- Ability to nondisruptively scale capacity and compute as demanding requirements change
- Ability to refresh compute (which happens more frequently than storage) without a performance-affecting data migration effort
- Flexibility to use excess top-of-rack switches bandwidth for the storage network, use a wholly different storage network such as Fibre Channel or SAS, or connect the array as direct-attached storage (DAS)

Figure 3) Sample of storage and compute separation.



This type of decoupling, for example, can allow a company with 100 nodes to reduce its number of nodes substantially if 100 nodes of compute are not required. This change provides a significant reduction in the rack space needed and the associated cooling and power requirements. In contrast, if the need is more compute, then less expensive servers can be purchased that do not require space for additional storage and have a smaller footprint. It also enables the use of blade server technology with their environmental savings.

This NVA describes using an E2824 with 12 SSDs for both the hot/warm tier and the cold tier as an example of what can be done. A different option involves using an E2860 with SSDs for the hot/warm tier and large-capacity NL-SAS drives for the cold tier.

4 Storage Topologies

4.1 iSCSI Network-Attached Topology

iSCSI encapsulates the native SCSI disk messages in a header and then uses TCP/IP to reliably transport them to their destination.

Because it is built on TCP/IP, iSCSI can be implemented entirely in software; indeed, virtually every operating system (including consumer versions of Microsoft Windows) includes software iSCSI. Software implementations generally are not appropriate for data center use because of low performance and because they consume server CPU and memory. However, there are many models of network adapter cards that have portions of the iSCSI/TCP/IP stack implemented in hardware to offload that processing from the server CPU. When deployed with hardware iSCSI and using the best practice of layer 2 adjacency, the performance of iSCSI approaches that of native FC or FCoE.

iSCSI does not require any special network configuration or equipment. If the network supports TCP/IP traffic, it supports iSCSI.

Its reliance on TCP/IP also creates the biggest problem for iSCSI: the temptation to route it. Introducing router hops to the iSCSI data path always adds latency. That latency can also be variable, depending on router load, queuing delays, and so on. That can result in the intermittent loss of storage connectivity, which is very difficult to troubleshoot because of its ephemeral nature. The best practice is layer 2 adjacency. The iSCSI target and initiator should be in the same VLAN/subnet on the same Ethernet switch. The subnet does not require any gateway or route out, because you are not routing the iSCSI. In this case, IP is only providing logical addressing.

An iSCSI network-attached topology is the cheapest and easiest of the four topologies to implement because the indexer-to-storage traffic goes over the same Ethernet connections that the indexer uses to communicate with the other Splunk components.

The advantages of the iSCSI network-attached topology are:

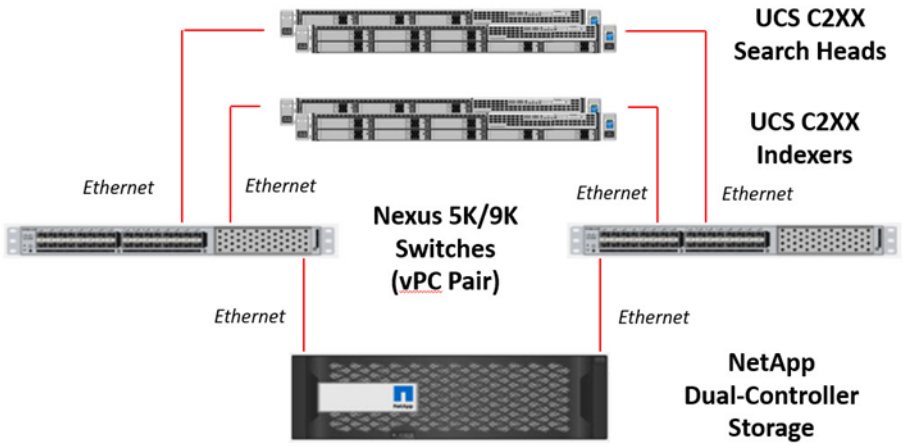
- Excellent performance to cost ratio, even with hardware iSCSI.
- Minimal cabling.
- Easy scale up. 40Gbps Ethernet has been available for several years and is becoming increasingly common; products that support up to 100Gbps are available.
- Easy scale out. A typical data center Ethernet switch has 48 ports in one rack unit. Deploying two such switches in a redundant pair (according to NetApp converged infrastructure best practices) typically leaves many available ports for adding additional devices.
- Operational flexibility. Each indexer can reach any storage controller connected to the network with only configuration changes required. Cabling changes are typically not required.

Caveats of the iSCSI network-attached topology are:

- Port bandwidth utilization should be monitored to make sure that adequate margins from saturation always exist, particularly on the indexer links, which are carrying incoming data traffic, replication traffic, and intracluster communications traffic on the same links.

Figure 4 depicts a notional iSCSI network-attached topology.

Figure 4) iSCSI network-attached topology.



4.2 FC Network-Attached Topology

Fibre Channel (FC) encapsulates the native SCSI disk messages in a header and then uses a variety of means to reliably transport them to their destination. When discussing converged infrastructure and NetApp E-Series with Splunk, there are two variants that differ in how the indexers connect to the FC network:

- The FC indexer link might be an FCoE link, running on the same converged network adapter (CNA) as the indexer's Ethernet traffic, connecting to an FCoE-capable Ethernet switch (for example, Cisco Nexus 5000 series).
- The FC indexer link might be a native FC link, running on a dedicated FC HBA in the indexer, connecting to an FC switch (for example, Cisco MDS).

Note that any FC link to the E-Series storage must be native FC (that is, not FCoE) because the NetApp E-Series does not support FCoE connections.

FCoE takes incoming FC frames (including their payload of native SCSI disk messages), encapsulates them in a specialized Ethernet frame, and uses some FCoE-specific technologies to reliably transport them to their destination. That might sound similar to iSCSI, but it is actually quite different. FCoE operates at the data link layer (layer 2, with no TCP/IP involved) and preserves the capabilities of FC because it preserves the FC frame. iSCSI operates at the session layer (layer 5) and uses iSCSI and TCP capabilities as substitutes for the missing FC capabilities.

FC and FCoE are implemented in hardware and tend to have better performance than iSCSI when running on similar hardware. However, the performance advantage of FC/FCoE becomes tiny when the iSCSI is deployed on data center-grade hardware, using best practices such as layer 2 adjacency. From an economic standpoint, FC/FCoE is typically more expensive than iSCSI. It also requires specialized network configuration such as zoning.

The results are that FC and FCoE are typically preferred by organizations that already have FC-based storage, and iSCSI is typically preferred by organizations that do not already have FC-based storage.

The advantages of the FC/FCoE network-attached topology are:

- Minimal cabling.
- Easy scale up. 40Gbps Ethernet is available in FC-enabled Ethernet switches and CNAs.
- Easy scale out. A typical data center FC-enabled Ethernet switch or a native FC switch has 48 ports in one rack unit. Deploying two such switches in redundant pairs (according to NetApp's converged infrastructure best practices) typically leaves many available ports for adding additional devices.

- Operational flexibility. Each indexer can reach any storage controller connected to the network, with only configuration changes required. Cabling changes are typically not required.

Caveats of the FC/FCoE network-attached topology are:

- Port bandwidth utilization should be monitored to make sure that adequate margins from saturation always exist, particularly on the indexer FCoE links, which are carrying incoming data traffic, replication traffic, and intracluster communications traffic on the same links.

Figure 5 and Figure 6 depict notional FCoE and native FC network-attached topologies.

Figure 5) FCoE network-attached topology.

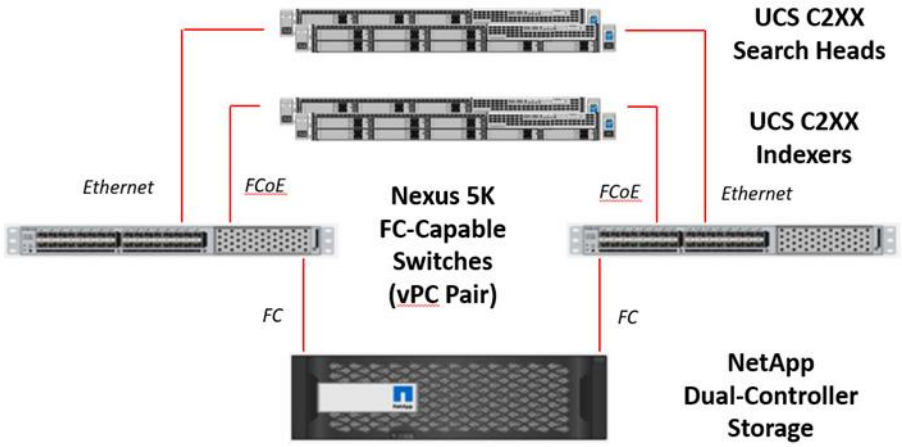
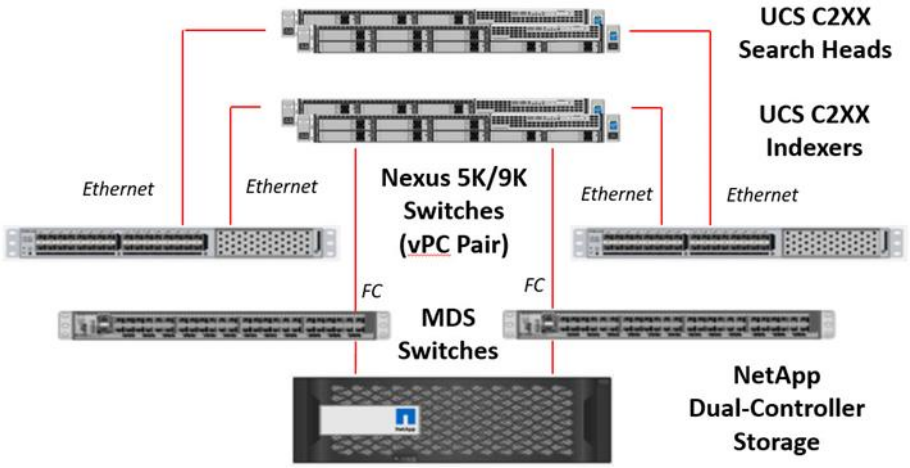


Figure 6) Native FC network-attached topology.



4.3 FC Direct-Attached Topology

The FC direct-attached topology uses dedicated FC links from ports on each indexer to ports on each storage controller that serves it. This provides the excellent performance of FC while eliminating the cost and complexity of FC/FCoE network components. The trade-offs are reduced scalability and operational flexibility.

This is a topology suitable for organizations that are not affected by the reduced scalability and operational flexibility. For example, they might include enough capacity in the initial deployment to

accommodate several years' growth, or their growth might be predictable enough that they can scale out in chunks with matched compute and storage capacity.

An easy way to eliminate the scalability limitations is to simply add a pair of FC switches, such as the Cisco MDS. This addition can be done nondisruptively, one link at a time. This might be a good strategy if the initial size of the installation is small enough to be a good fit with FC direct-attached and only add the FC switches if and when the installation has grown to a large enough size to warrant it.

The advantages of the FC direct-attached topology are:

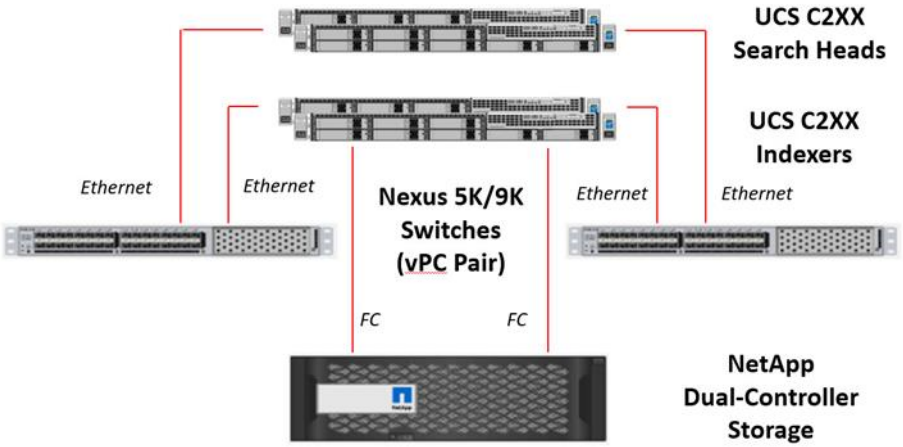
- Excellent performance
- Good performance to cost ratio
- Lower risk of a saturated link
- Easy to improve scalability by adding FC switches

Caveats of the FC direct-attached topology are:

- More cabling than network-attached topologies.
- Scalability is limited by the smaller number of ports available on the storage controller relative to a switch. This is mitigated by adding optional HIC cards to the storage controller to provide more data ports.
- Operational flexibility is more limited than a networked topology, because each indexer can only reach storage to which it is connected. This can result in stranded resources. Both cabling and configuration changes are typically required to change connectivity.

Figure 7 depicts a notional FC direct-attached topology.

Figure 7) FC direct-attached topology.



4.4 SAS Direct-Attached Topology

The SAS direct-attached topology uses dedicated SAS links from ports on each indexer to ports on each storage controller that serves it. It provides the lowest cost option for very small installations because it avoids the cost of FC HBAs in the servers. It has very limited scalability, like the FC direct-attached topology; however, unlike the FC direct-attached topology, it does not have an easy upgrade path to improve scalability.

The advantages of the SAS direct-attached topology are:

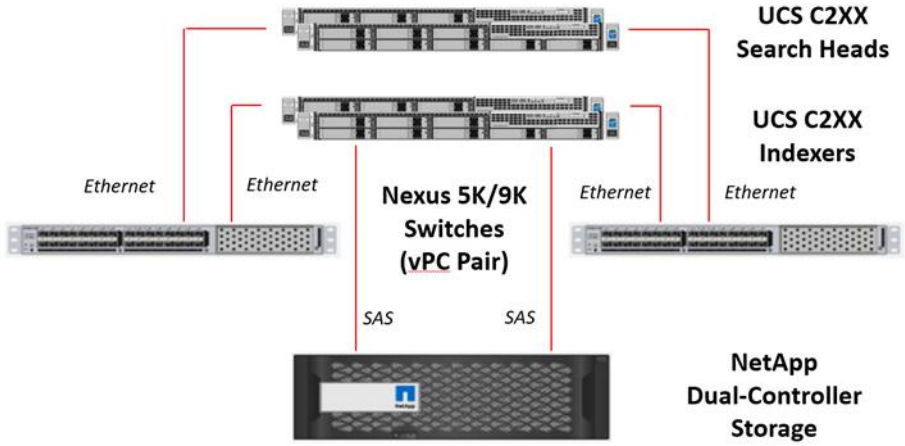
- Good performance
- Excellent performance to cost ratio
- Low risk of a saturated link

Caveats of the SAS direct-attached topology are:

- More cabling than network-attached topologies
- Limited scalability, with no easy path to improve scalability
- Limited operational flexibility

Figure 8 depicts a notional SAS direct-attached topology.

Figure 8) SAS direct-attached topology.



5 Technology Requirements

This section covers the technology requirements for the converged infrastructure solution with NetApp E-Series and Splunk.

5.1 Hardware Requirements

Table 2 lists the hardware components that were used to verify all topologies discussed in this solution in the NetApp labs. Because a production deployment only uses one of the topologies, the hardware components that are used in any particular deployment of the solution are a subset of this list. The models deployed should be tailored to actual customer requirements.

Table 2) Hardware requirements.

Component	Model	Quantity	Software
Splunk search head	Cisco UCS C220M4 Cisco VIC 1227 CNA	1 1 each	Cisco UCS firmware 3.0(3a) RHEL 7.3 VIC 1227: enic-2.3.0.39
Splunk indexer	Cisco UCS C240M4 Cisco VIC 1227 CNA Qlogic QLE2672 HBA Cisco 9300-8e SAS HBA	3 1 each 1 each 1 each	Cisco UCS firmware 3.0(3a) RHEL 7.3 VIC 1227: enic-2.3.0.39
Ethernet switch	Cisco Nexus 9372PX	2	NX-OS 7.0(3)I4(6)
FC switch	Cisco MDS 9148	2	NX-OS 6.2(19)

Component	Model	Quantity	Software
Storage	NetApp E2824	1	SANtricity® 11.40

5.2 Software Requirements

Table 3 lists the software components that were used to verify the solution in the NetApp labs. The software components that are used in any particular implementation of the solution might vary based on customer requirements.

Table 3) Software requirements.

Software	Version	Notes
Splunk Enterprise	6.6.2	Enterprise Edition
NetApp SANtricity Performance App for Splunk Enterprise	1.04	Requires Splunk 6.2 or later
NetApp Technology Add-On for NetApp SANtricity	1.0.0	Requires Splunk 6.1 or later
NetApp SANtricity Web Services Proxy	2.1	2.0 or later for best feature set with Splunk app

6 Deployment Procedures

For the verification testing, we tested the following topologies:

- iSCSI network-attached topology
- FC/FCoE network-attached topology
- FC direct-attached topology
- SAS direct-attached topology

Table 4 lists the VLANs that were used in the development of this solution. Some are topology-specific, such as the iSCSI VLANs, which are only used for the iSCSI topology. The VLANs can be adjusted as needed for specific deployments.

Table 4) Necessary VLANs.

VLAN Name	VLAN Purpose	ID Used in Validating This Document
Management in band	VLAN for inband management interfaces	1172
Native	VLAN to which untagged frames are assigned	2
iSCSI-A	VLAN for ISCSI traffic for fabric A	91
iSCSI-B	VLAN for ISCSI traffic for fabric B	92
Splunk traffic	VLAN for Splunk application traffic	99

Table 5 lists the configuration variables (IP addresses and so on) that are referred to in this document. Deployments typically go more smoothly when this information is determined before execution. The following configuration variables are related to the networking configuration.

Table 5) Network configuration variables.

Variable	Description	Customer Implementation Value
<<var_password>>	Administrative password	
<<var_global_ntp_server_ip>>	NTP server IP address	
<<var_dns1_ip>>	Primary DNS server IP address	
<<var_dns2_ip>>	Secondary DNS server IP address	
<<var_nexus_A_hostname>>	Host name for first Cisco Nexus switch	
<<var_nexus_A_mgmt0_ip>>	Out-of-band management IP address for first Cisco Nexus switch	
<<var_nexus_A_mgmt0_netmask>>	Out-of-band management IP netmask for first Cisco Nexus switch	
<<var_nexus_A_mgmt0_gw>>	Out-of-band management IP gateway for first Cisco Nexus switch	
<<var_nexus_B_hostname>>	Host name for second Cisco Nexus switch	
<<var_nexus_B_mgmt0_ip>>	Out-of-band management IP address for second Cisco Nexus switch	
<<var_nexus_B_mgmt0_netmask>>	Out-of-band management IP netmask for second Cisco Nexus switch	
<<var_nexus_B_mgmt0_gw>>	Out-of-band management IP gateway for second Cisco Nexus switch	
<<var_native_vlan>>	Native VLAN ID	
<<var_splunk_traffic_vlan>>	Splunk traffic VLAN ID	
<<var_iscsi_a_vlan>>	Fabric A iSCSI VLAN ID	
<<var_iscsi_b_vlan>>	Fabric B iSCSI VLAN ID	
<<var_vsan_a_id>>	Fabric A FC VSAN ID	
<<var_vsan_b_id>>	Fabric B FC VSAN ID	
<<var_fabric_a_fcoe_vlan_id>>	Fabric A FCoE VLAN ID	
<<var_fabric_b_fcoe_vlan_id>>	Fabric B FCoE VLAN ID	

Table 6 lists the configuration variables related to the server configuration.

Table 6) Server configuration variables.

Variable	Description	Customer Implementation Value
<<var_indexer1_name>>	Host name for first indexer server	
<<var_indexer1_cimc_ip>>	CIMC IP address for first indexer	
<<var_indexer1_cimc_netmask>>	CIMC IP netmask for first indexer	
<<var_indexer1_cimc_gw>>	CIMC IP gateway for first indexer	

Variable	Description	Customer Implementation Value
<<var_indexer1_mgmt_ip>>	Inband management IP address for first indexer	
<<var_indexer1_mgmt_netmask>>	Inband management IP netmask for first indexer	
<<var_indexer1_mgmt_gw>>	Inband management IP gateway for first indexer	
<<var_indexer1_splunk_ip>>	Splunk IP address for first indexer	
<<var_indexer1_splunk_netmask>>	Splunk IP netmask for first indexer	
<<var_indexer1_iSCSI_A_ip>>	iSCSI fabric A IP address for first indexer	
<<var_indexer1_iSCSI_A_netmask>>	iSCSI fabric A IP netmask for first indexer	
<<var_indexer1_iSCSI_A_mac>>	iSCSI fabric A MAC address for first indexer	
<<var_indexer1_iSCSI_B_ip>>	iSCSI fabric B IP address for first indexer	
<<var_indexer1_iSCSI_B_netmask>>	iSCSI fabric B IP netmask for first indexer	
<<var_indexer1_iSCSI_B_mac>>	iSCSI fabric B MAC address for first indexer	
<<var_indexer1_A_wwpn>>	FC/FCoE fabric A WWPN for first indexer	
<<var_indexer1_B_wwpn>>	FC/FCoE fabric A WWPN for first indexer	
<<var_indexer2_name>>	Host name for second indexer server	
<<var_indexer2_cimc_ip>>	CIMC IP address for second indexer	
<<var_indexer2_cimc_netmask>>	CIMC IP netmask for second indexer	
<<var_indexer2_cimc_gw>>	CIMC IP gateway for second indexer	
<<var_indexer2_mgmt_ip>>	Inband management IP address for second indexer	
<<var_indexer2_mgmt_netmask>>	Inband management IP netmask for second indexer	
<<var_indexer2_mgmt_gw>>	Inband management IP gateway for second indexer	
<<var_indexer2_splunk_ip>>	Splunk IP address for second indexer	
<<var_indexer2_splunk_netmask>>	Splunk IP netmask for second indexer	
<<var_indexer2_iSCSI_A_ip>>	iSCSI fabric A IP address for second indexer	
<<var_indexer2_iSCSI_A_netmask>>	iSCSI fabric A IP netmask for second indexer	
<<var_indexer2_iSCSI_A_mac>>	iSCSI fabric A MAC address for second indexer	
<<var_indexer2_iSCSI_B_ip>>	iSCSI fabric B IP address for second indexer	
<<var_indexer2_iSCSI_B_netmask>>	iSCSI fabric B IP netmask for second indexer	
<<var_indexer2_iSCSI_B_mac>>	iSCSI fabric B MAC address for second indexer	

Variable	Description	Customer Implementation Value
<<var_indexer2_A_wwpn>>	FC/FCoE fabric A WWPN for second indexer	
<<var_indexer2_B_wwpn>>	FC/FCoE fabric A WWPN for second indexer	
<<var_indexer3_name>>	Host name for third indexer server	
<<var_indexer3_cimc_ip>>	CIMC IP address for third indexer	
<<var_indexer3_cimc_netmask>>	CIMC IP netmask for third indexer	
<<var_indexer3_cimc_gw>>	CIMC IP gateway for third indexer	
<<var_indexer3_mgmt_ip>>	Inband management IP address for third indexer	
<<var_indexer3_mgmt_netmask>>	Inband management IP netmask for third indexer	
<<var_indexer3_mgmt_gw>>	Inband management IP gateway for third indexer	
<<var_indexer3_splunk_ip>>	Splunk IP address for third indexer	
<<var_indexer3_splunk_netmask>>	Splunk IP netmask for third indexer	
<<var_indexer3_iSCSI_A_ip>>	iSCSI fabric A IP address for third indexer	
<<var_indexer3_iSCSI_A_netmask>>	iSCSI fabric A IP netmask for third indexer	
<<var_indexer3_iSCSI_A_mac>>	iSCSI fabric A MAC address for third indexer	
<<var_indexer3_iSCSI_B_ip>>	iSCSI fabric B IP address for third indexer	
<<var_indexer3_iSCSI_B_netmask>>	iSCSI fabric B IP netmask for third indexer	
<<var_indexer3_iSCSI_B_mac>>	iSCSI fabric B MAC address for third indexer	
<<var_indexer3_A_wwpn>>	FC/FCoE fabric A WWPN for third indexer	
<<var_indexer3_B_wwpn>>	FC/FCoE fabric A WWPN for third indexer	
<<var_search1_name>>	Host name for first search head	
<<var_search1_cimc_ip>>	CIMC IP address for first search head	
<<var_search1_cimc_netmask>>	CIMC IP netmask for first search head	
<<var_search1_cimc_gw>>	CIMC IP gateway for first search head	
<<var_search1_mgmt_ip>>	Inband management IP address for first search head	
<<var_search1_mgmt_netmask>>	Inband management IP netmask for first search head	
<<var_search1_mgmt_gw>>	Inband management IP gateway for first search head	
<<var_search1_splunk_ip>>	Splunk IP address for first search head	
<<var_search1_splunk_netmask>>	Splunk IP netmask for first search head	
<<var_forwarder1_name>>	Host name for first forwarder	

Variable	Description	Customer Implementation Value
<<var_forwarder1_cimc_ip>>	CIMC IP address for first forwarder	
<<var_forwarder1_cimc_netmask>>	CIMC IP netmask for first forwarder	
<<var_forwarder1_cimc_gw>>	CIMC IP gateway for first forwarder	
<<var_forwarder1_mgmt_ip>>	Inband management IP address for first forwarder	
<<var_forwarder1_mgmt_netmask>>	Inband management IP netmask for first forwarder	
<<var_forwarder1_mgmt_gw>>	Inband management IP gateway for first forwarder	
<<var_forwarder1_splunk_ip>>	Splunk IP address for first forwarder	
<<var_forwarder1_splunk_netmask>>	Splunk IP netmask for first forwarder	

Table 7 lists the configuration variables related to the storage configuration.

Table 7) Storage configuration variables.

Variable	Description	Customer Implementation Value
<<var_admin_password>>	Administrator password	
<<var_e2800_name>>	Host name for storage controller	
<<var_e2800_A_mgmt_ip>>	Out-of-band management IP address for first E2800 controller	
<<var_e2800_A_mgmt_netmask>>	Out-of-band management IP netmask for first E2800 controller	
<<var_e2800_A_mgmt_gw>>	Out-of-band management IP gateway for first E2800 controller	
<<var_e2800_B_mgmt_ip>>	Out-of-band management IP address for second E2800 controller	
<<var_e2800_B_mgmt0_netmask>>	Out-of-band management IP netmask for second E2800 controller	
<<var_e2800_B_mgmt0_gw>>	Out-of-band management IP gateway for second E2800 controller	
<<var_storage_workload_name>>	Workload name	
<<var_e2800_A_iSCSI_A_ip>>	Fabric A iSCSI IP address for first E2800 controller	
<<var_e2800_A_iSCSI_A_netmask>>	Fabric A iSCSI IP address for first E2800 controller	
<<var_e2800_A_iSCSI_B_ip>>	Fabric B iSCSI IP address for first E2800 controller	
<<var_e2800_A_iSCSI_B_netmask>>	Fabric B iSCSI IP address for first E2800 controller	

Variable	Description	Customer Implementation Value
<<var_e2800_B_iSCSI_A_ip>>	Fabric A iSCSI IP address for second E2800 controller	
<<var_e2800_B_iSCSI_A_netmask>>	Fabric A iSCSI IP address for second E2800 controller	
<<var_e2800_B_iSCSI_B_ip>>	Fabric B iSCSI IP address for second E2800 controller	
<<var_e2800_B_iSCSI_B_netmask>>	Fabric B iSCSI IP address for second E2800 controller	
<<var_e2800_A_A_wwpn>>	Fabric A WWPN for first E2800 controller	
<<var_e2800_A_B_wwpn>>	Fabric B WWPN for first E2800 controller	
<<var_e2800_B_A_wwpn>>	Fabric A WWPN for second E2800 controller	
<<var_e2800_B_B_wwpn>>	Fabric B WWPN for second E2800 controller	
<<var_e2800_iqn>>	Target IQN for E2800 controller	
<<var_indexer1_hot_wwi>>	WWI for hot/warm volume on first indexer	
<<var_indexer1_cold_wwi>>	WWI for cold volume on first indexer	
<<var_indexer2_hot_wwi>>	WWI for hot/warm volume on second indexer	
<<var_indexer2_cold_wwi>>	WWI for cold volume on second indexer	
<<var_indexer3_hot_wwi>>	WWI for hot/warm volume on third indexer	
<<var_indexer3_cold_wwi>>	WWI for cold volume on third indexer	
<<var_indexer1_hot_partition>>	Partition identifier for hot/warm volume on first indexer	
<<var_indexer1_cold_partition>>	Partition identifier for cold volume on first indexer	
<<var_indexer2_hot_partition>>	Partition identifier for hot/warm volume on second indexer	
<<var_indexer2_cold_partition>>	Partition identifier for cold volume on second indexer	
<<var_indexer3_hot_partition>>	Partition identifier for hot/warm volume on third indexer	
<<var_indexer2_cold_partition>>	Partition identifier for cold volume on third indexer	
<<var_hot_mount>>	Mount point for hot/warm volume on indexer	
<<var_cold_mount>>	Mount point for cold volume on indexer	

6.1 Cisco Nexus Switch Initial Configuration (All Topologies)

Initial switch configuration is performed through the console port. If there is no existing startup config on the switch when it powers on, the switch attempts to automatically retrieve its configuration through the power on autoprovisioning (POAP) feature.

1. For manual configuration, abort the POAP process as directed by the prompt; set the admin password; and enter the basic configuration dialog box, which prompts for parameters such as host name, management IP address, and so on.

   Here is an example:

```
Abort Power on Auto Provisioning and continue with normal setup? (yes/no) [n]: yes
Do you want to enforce secure password standard (yes/no): yes
Enter the password for "admin": <<var_password>>
Confirm the password for "admin": <<var_password>>
Would you like to enter the basic configuration dialog (yes/no): yes
Create another login account (yes/no) [n]: no
Configure read-only SNMP community string (yes/no) [n]: no
Configure read-write SNMP community string (yes/no) [n]: no
Enter the switch name: <<var_nexus_A_hostname>>
Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: yes
Mgmt0 IPv4 address: <<var_nexus_A_mgmt0_ip>>
Mgmt0 IPv4 netmask: <<var_nexus_A_mgmt0_netmask>>
Configure the default gateway? (yes/no) [y]: yes
IPv4 address of the default gateway: <<var_nexus_A_mgmt0_gw>>
Configure advanced IP options? (yes/no) [n]: no
Enable the telnet service? (yes/no) [n]: no
Enable the ssh service? (yes/no) [y]: yes

Type of ssh key you would like to generate (dsa/rsa) [rsa]: rsa
Number of rsa key bits <1024-2048> [1024]: 1024
Configure the ntp server? (yes/no) [n]: yes
NTP server IPv4 address: <<var_global_ntp_server_ip>>
Configure default interface layer (L3/L2) [L2]: L2
Configure default switchport interface state (shut/noshut) [noshut]: shut
Configure CoPP system profile (strict/moderate/lenient/dense/skip) [strict]: strict
  Would you like to edit the configuration? (yes/no) [n]: no
```

2. Review the configuration summary before enabling the configuration.

```
Use this configuration and save it? (yes/no) [y]: yes
```

3. Repeat steps 1 and 2 for the other Cisco Nexus switch.

6.2 Enable FCoE on Cisco Nexus Switch (FCoE Network Topology Only)

Note: The Cisco Nexus FC features require a license to be installed before they can be enabled.

Note: We did not include the FCoE scenario in our lab testing for this NVA. However, we are including the FCoE steps, at the appropriate place in the config process, to assist those who might be including FCoE in their deployment.

For the FCoE network-attached topology using Cisco Nexus 5000 series switches, the FCoE software features must be enabled, and then any switch ports that are running native FC (such as the ones connected to the E2800 FC ports) must be changed from Ethernet mode to FC mode. Here are some things to know before changing the port mode:

- Ethernet ports must be in one contiguous block, starting from the first port of the module. FC ports must be in one contiguous block, starting from the last port of the module. For example, if the switch has 32 ports, and you need 8 FC ports, then the Ethernet ports would be e1/1–24, and the FC ports would be fc1/25–32.
- If the ports being changed are in the fixed module, the switch must be rebooted to make the port mode change effective. Add-in modules that support the poweroff command might be able to change the mode without rebooting the entire switch.

Here are the commands to enable the FC features and change some ports to FC:

```
config t
feature fcoe
feature npiv
end
copy run start
```



```
config t
slot 1
port 31-32 type fc
end
copy run start
reload
```

6.3 Cisco Nexus Switch Global Configuration (All Topologies)

1. Global config is used to set switchwide parameters. Use the following commands to enable software features, configure spanning tree protocol defaults and NTP, and configure inband management and Splunk traffic VLANs:

```
config t
feature interface-vlan
feature lacp
feature vpc
feature lldp
spanning-tree port type network default
spanning-tree port type edge bpduguard default
spanning-tree port type edge bpdufilter default
port-channel load-balance src-dst l4port
ntp server <<var_global_ntp_server_ip>> use-vrf management
ntp master 3
vlan <<var_native_vlan>>
name Native-VLAN
exit
vlan <<var_splunk_traffic_vlan>>
name Splunk-Traffic-VLAN
end
copy run start
```

2. Repeat step 1 for the other Cisco Nexus switch.

6.4 Cisco Nexus Switch Port and Port Channel Configuration (All Topologies)

A port channel is an aggregation of physical links that act as one logical link. Cisco extended the basic port channel technology by allowing the network end of the links to be split between two switches, which provides device redundancy in the network switches. This technology is called a virtual port channel (vPC). In this section we configure the ports and port channels on each switch independently; then, in the next section, we combine the two switches into a vPC domain.

1. If the Cisco Nexus switches are in the Cisco Nexus 5000 series, set the default MTU using policies using the following commands (do this on both switches):

```
config t
policy-map type network-qos jumbo
class type network-qos class-default
mtu 9216
exit
class type network-qos class-fcoe
pause no-drop
mtu 2158
exit
exit
system qos
service-policy type network-qos jumbo
end
copy run start
```

2. Create each port channel interface (by entering “interface PoX” in global config mode, where “X” is the port channel number) and add the physical ports to it (through the channel-group parameter on the physical interface). Then configure the parameters on the port channel interface.

   This order is preferable because the physical port’s parameters must match those of the port channel it is joining for the port add operation. It is easier to do this when there are no parameters on either. After the physical ports are added, most config to the physical port is blocked, and parameters configured on the port channel interface are automatically pushed out to its member ports.

   Note: A warning message is generated when the port channel interfaces are configured for “spanning-tree port type edge.”:That is expected and can be ignored for this step.

   Use the following commands:

```
config t
interface Po10
description vPC peer-link
exit
interface Eth1/21-22
description vPC peer-link
channel-group 10 mode active
no shutdown
exit
interface Po11
description <<var_indexer1_name>>
exit
interface Eth1/1
description <<var_indexer1_name>>:Port1
channel-group 11 mode active
no shutdown
exit
interface Po12
description <<var_indexer2_name>>
exit
interface Eth1/2
description <<var_indexer2_name>>:Port1
channel-group 12 mode active
no shutdown
exit
interface Po13
description <<var_indexer3_name>>
exit
interface Eth1/3
description <<var_indexer3_name>>:Port1
channel-group 13 mode active
no shutdown
exit
interface Po21
description <<var_search1_name>>
exit
interface Eth1/11
description <<var_search1_name>>:Port1
```



```
channel-group 21 mode active
no shutdown
exit
interface Po10
switchport mode trunk
spanning-tree port type network
exit
interface Po11
switchport mode trunk
switchport trunk native vlan <<var_native_vlan>>
switchport trunk allowed vlan <<var_splunk_traffic_vlan>>
spanning-tree port type edge trunk
exit
interface Po12
switchport mode trunk
switchport trunk native vlan <<var_native_vlan>>
switchport trunk allowed vlan <<var_splunk_traffic_vlan>>
spanning-tree port type edge trunk
exit
interface Po13
switchport mode trunk
switchport trunk native vlan <<var_native_vlan>>
switchport trunk allowed vlan <<var_splunk_traffic_vlan>>
spanning-tree port type edge trunk
exit
interface Po21
switchport mode trunk
switchport trunk native vlan <<var_native_vlan>>
switchport trunk allowed vlan <<var_splunk_traffic_vlan>>
spanning-tree port type edge trunk
end
copy run start
```

3. Repeat step 2 for the other Cisco Nexus switch.
4. If the Cisco Nexus switches are in the Cisco Nexus 9000 series, set the MTU on the server port channel interfaces using the following commands (do this on both switches):

```
config t
interface Po11
mtu 9216
exit
interface Po12
mtu 9216
exit
interface Po13
mtu 9216
exit
interface Po21
mtu 9216
end
copy run start
```

6.5 Cisco Nexus Switch vPC Configuration (All Topologies)

This section explains how to join the two switches in a vPC domain.

1. Use the following commands for the first switch:

```
config t
vpc domain 10
role priority 10
peer-keepalive destination <<var_nexus_B_mgmt0_ip>> source <<var_nexus_A_mgmt0_ip>>
peer-switch
peer-gateway
auto-recovery
delay restore 150
exit
interface Po10
vpc peer-link
```



```
exit
interface Po11
vpc 11
exit
interface Po12
vpc 12
exit
interface Po13
vpc 13
exit
interface Po21
vpc 21
end
copy run start
```

2. Use the following commands for the second switch:

```
config t
vpc domain 10
role priority 20
peer-keepalive destination <<var_nexus_A_mgmt0_ip>> source <<var_nexus_B_mgmt0_ip>>
peer-switch
peer-gateway
auto-recovery
delay restore 150
exit
interface Po10
vpc peer-link
exit
interface Po11
vpc 11
exit
interface Po12
vpc 12
exit
interface Po13
vpc 13
exit
interface Po21
vpc 21
end
copy run start
```

Note that the role priority and peer-keepalive source and destination differ between the two switches.

6.6 Cisco Nexus Switch iSCSI Configuration (iSCSI Topology Only)

If iSCSI is being used, you need to add the two iSCSI VLANs to the VLAN database and add those VLANs to the port channels that serve the storage and the indexers. (The search heads do not participate in the iSCSI, so the iSCSI VLANs should not be added to the port channels that serve search heads. Because you are not restricting VLANs on the peer link, you do not need to explicitly allow them there; they are already implicitly allowed.) In this example, the two ports serving the E2824 are configured as access ports directly on the physical ports because the NetApp E-Series does not use link aggregation. The three port channels go to the indexers.

1. Use the following commands to add iSCSI to switch A:

```
config t
vlan <<var_iscsi_a_vlan>>
name ISCSI-A-VLAN
exit
vlan <<var_iscsi_b_vlan>>
name ISCSI-B-VLAN
exit
interface Eth1/15
description <<var_e2800_name>>-A:0a
switchport mode access
switchport access vlan <<var_iscsi_a_vlan>>
```



```
spanning-tree port type edge
no shutdown
exit
interface Eth1/16
description <<var_e2800_name>>-B:0a
switchport mode access
switchport access vlan <<var_iscsi_b_vlan>>
spanning-tree port type edge
no shutdown
exit

interface Po11
switchport trunk allowed vlan add <<var_iscsi_a_vlan>>,<<var_iscsi_b_vlan>>
exit
interface Po12
switchport trunk allowed vlan add <<var_iscsi_a_vlan>>,<<var_iscsi_b_vlan>>
exit
interface Po13
switchport trunk allowed vlan add <<var_iscsi_a_vlan>>,<<var_iscsi_b_vlan>>
end
copy run start
```

2. Use the following commands to add iSCSI to switch B:

```
config t
vlan <<var_iscsi_a_vlan>>
name ISCSI-A-VLAN
exit
vlan <<var_iscsi_b_vlan>>
name ISCSI-B-VLAN
exit
interface Eth1/15
description <<var_e2800_name>>-A:0b
switchport mode access
switchport access vlan <<var_iscsi_b_vlan>>
spanning-tree port type edge
no shutdown
exit
interface Eth1/16
description <<var_e2800_name>>-B:0b
switchport mode access
switchport access vlan <<var_iscsi_a_vlan>>
spanning-tree port type edge
no shutdown
exit

interface Po11
switchport trunk allowed vlan add <<var_iscsi_a_vlan>>,<<var_iscsi_b_vlan>>
exit
interface Po12
switchport trunk allowed vlan add <<var_iscsi_a_vlan>>,<<var_iscsi_b_vlan>>
exit
interface Po13
switchport trunk allowed vlan add <<var_iscsi_a_vlan>>,<<var_iscsi_b_vlan>>
end
copy run start
```

3. If the Cisco Nexus switches are in the Cisco Nexus 9000 series, set the MTU on the physical ports connected to the E2800 using these commands (do this on both switches):

```
config t
interface e1/11
mtu 9216
exit
interface e1/12
mtu 9216
end
copy run start
```

6.7 Cisco MDS Switch Configuration (Native FC Network Topology Only)

In the FC network-attached topology, the indexers are communicating over the FC network to the E2800 FC ports through a pair of MDS 9148 FC switches. Each switch represents a separate FC fabric; they do not interact in any way. Each indexer has one FC port connected to each fabric, and each E2800 controller has two FC ports connected to each fabric.

1. Configure the switch ports, then populate the VSAN database. Use the following commands for the first switch (also known as fabric A):

```
config t
interface fc1/1
switchport description <<var_indexer1_name>>:Port1
no shutdown
exit
interface fc1/2
switchport description <<var_indexer2_name>>:Port1
no shutdown
exit
interface fc1/3
switchport description <<var_indexer3_name>>:Port1
no shutdown
exit
interface fc1/11
switchport description <<var_e2800_name>>-A:0c
no shutdown
exit
interface fc1/12
switchport description <<var_e2800_name>>-B:0c
no shutdown
exit
vsan database
vsan <<var_vsan_a_id>> name Fabric_A
vsan <<var_vsan_a_id>> interface fc1/1
vsan <<var_vsan_a_id>> interface fc1/2
vsan <<var_vsan_a_id>> interface fc1/3
vsan <<var_vsan_a_id>> interface fc1/11
vsan <<var_vsan_a_id>> interface fc1/12
end
copy run start
```

2. Repeat this step for the second switch (fabric B).

6.8 Cisco Nexus Switch FC/FCoE Configuration (FCoE Network Topology Only)

In an FC/FCoE network-attached topology, the indexer ports are running FCoE by way of the CNA, and the E2800 FC ports are running native FC. The VSAN that defines the FC fabric is associated with an Ethernet VLAN on the links between the indexers and the switch; a virtual FC (vfc) port is bound to the corresponding Ethernet port on the switch. At the indexer end, the CNA presents both a virtual HBA (vHBA, for FC connectivity) and a virtual NIC (vNIC, for Ethernet connectivity) to the server. Each switch represents a separate FC fabric.

1. Configure ports and links: Associate the VSAN to the VLAN, allow the VSAN's VLAN on the trunk, and bind the vfc ports to their Ethernet ports. Finish this step by populating the VSAN database. Use the following commands for the first switch (also known as fabric A):

```
config t
vlan <<var_fabric_a_fcoe_vlan_id>>
name FCoE_Fabric_A
fcoe vsan <<var_vsan_a_id>>
exit
interface po11
switchport trunk allowed vlan add <<var_fabric_a_fcoe_vlan_id>>
exit
interface vfc11
```



```
switchport description <<var_indexer1_name>>:Port1
bind interface Eth1/1
switchport trunk allowed vsan <<var_vsan_a_id>>
no shutdown
exit
interface po12
switchport trunk allowed vlan add <<var_fabric_a_fcoe_vlan_id>>
exit
interface vfc12
switchport description <<var_indexer2_name>>:Port1
bind interface Eth1/2
switchport trunk allowed vsan <<var_vsan_a_id>>
no shutdown
exit
interface po13
switchport trunk allowed vlan add <<var_fabric_a_fcoe_vlan_id>>
exit
interface vfc13
switchport description <<var_indexer3_name>>:Port1
bind interface Eth1/3
switchport trunk allowed vsan <<var_vsan_a_id>>
no shutdown
exit
interface fc1/31
switchport description <<var_e2800_name>>-A:0c
no shutdown
exit
interface fc1/32
switchport description <<var_e2800_name>>-B:0c
no shutdown
exit
vsan database
vsan <<var_vsan_a_id>> name Fabric_A
vsan <<var_vsan_a_id>> interface vfc11
vsan <<var_vsan_a_id>> interface vfc12
vsan <<var_vsan_a_id>> interface vfc13
vsan <<var_vsan_a_id>> interface fc1/31
vsan <<var_vsan_a_id>> interface fc1/32
end
copy run start
```

2. Use the following commands for the second switch (fabric B):

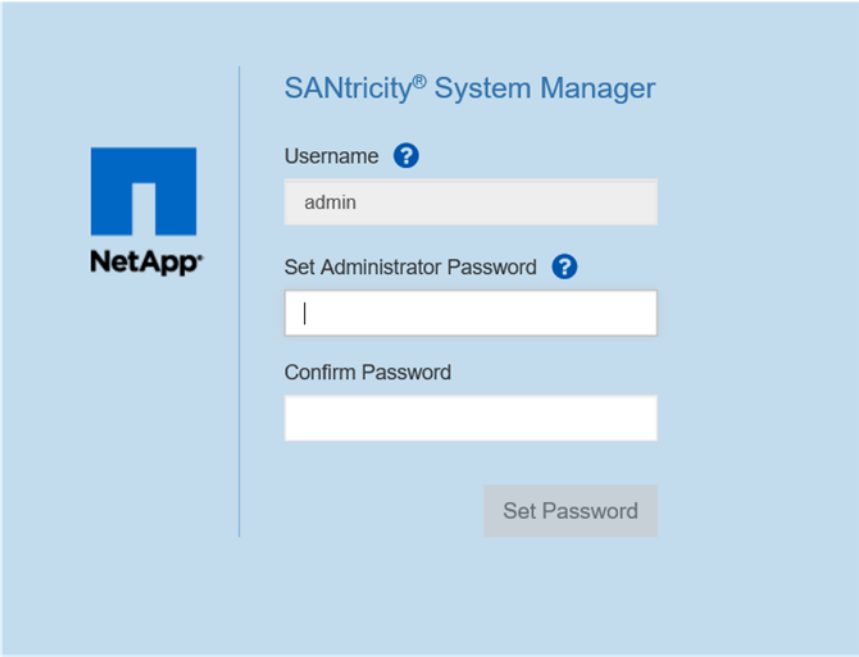
```
config t
vlan <<var_fabric_b_fcoe_vlan_id>>
name FCoE_Fabric_B
fcoe vsan <<var_vsan_b_id>>
exit
interface po11
switchport trunk allowed vlan add <<var_fabric_b_fcoe_vlan_id>>
exit
interface vfc11
switchport description <<var_indexer1_name>>:Port2
bind interface Eth1/1
switchport trunk allowed vsan <<var_vsan_b_id>>
no shutdown
exit
interface po12
switchport trunk allowed vlan add <<var_fabric_b_fcoe_vlan_id>>
exit
interface vfc12
switchport description <<var_indexer2_name>>:Port2
bind interface Eth1/2
switchport trunk allowed vsan <<var_vsan_b_id>>
no shutdown
exit
interface po13
switchport trunk allowed vlan add <<var_fabric_b_fcoe_vlan_id>>
exit
interface vfc13
switchport description <<var_indexer3_name>>:Port2
```



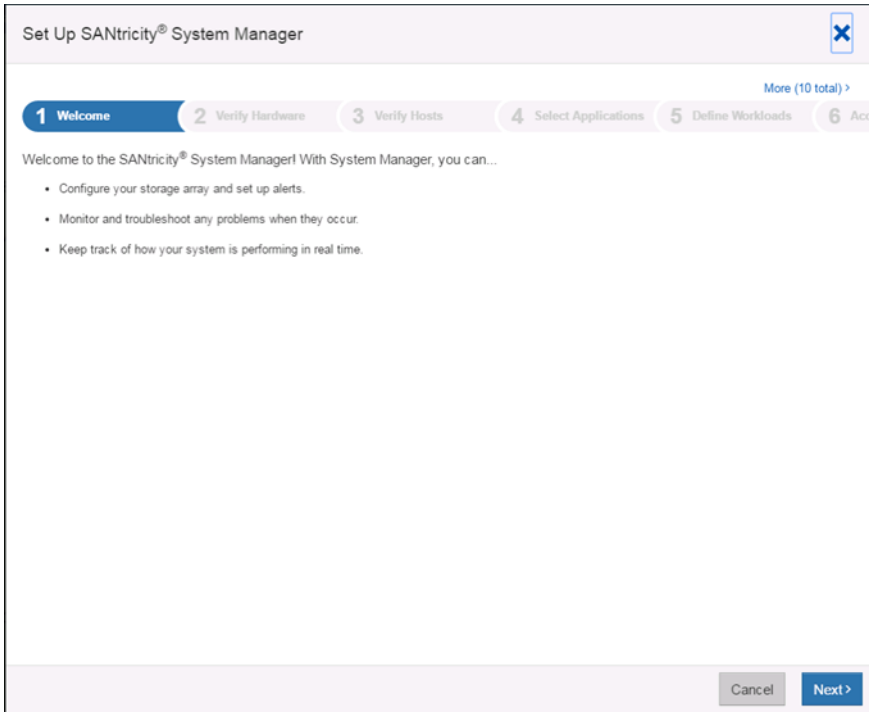
```
bind interface Eth1/3
switchport trunk allowed vsan <<var_vsan_b_id>>
no shutdown
exit
interface fc1/31
switchport description <<var_e2800_name>>-A:0d
no shutdown
exit
interface fc1/32
switchport description <<var_e2800_name>>-B:0d
no shutdown
exit
vsan database
vsan <<var_vsan_a_id>> name Fabric_B
vsan <<var_vsan_a_id>> interface vfc11
vsan <<var_vsan_a_id>> interface vfc12
vsan <<var_vsan_a_id>> interface vfc13
vsan <<var_vsan_a_id>> interface fc1/31
vsan <<var_vsan_a_id>> interface fc1/32
end
copy run start
```

6.9 NetApp E2824 Initial Configuration (All Topologies)

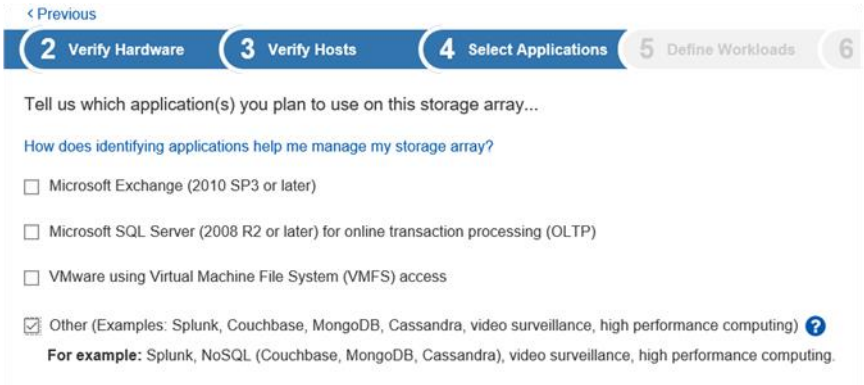
1. Connect the management port (port 1) of controller A to the reader workstation.
2. Set the IP address of the reader workstation to 192.168.128.100 with a subnet mask of 255.255.255.0. Leave the gateway and DNS servers blank.
3. Launch a browser and access the controller A using IP 192.168.128.101.
4. Set and confirm an administrator password <<var_admin_password>>.



5. The SANtricity System Manager setup wizard launches. Click Next.



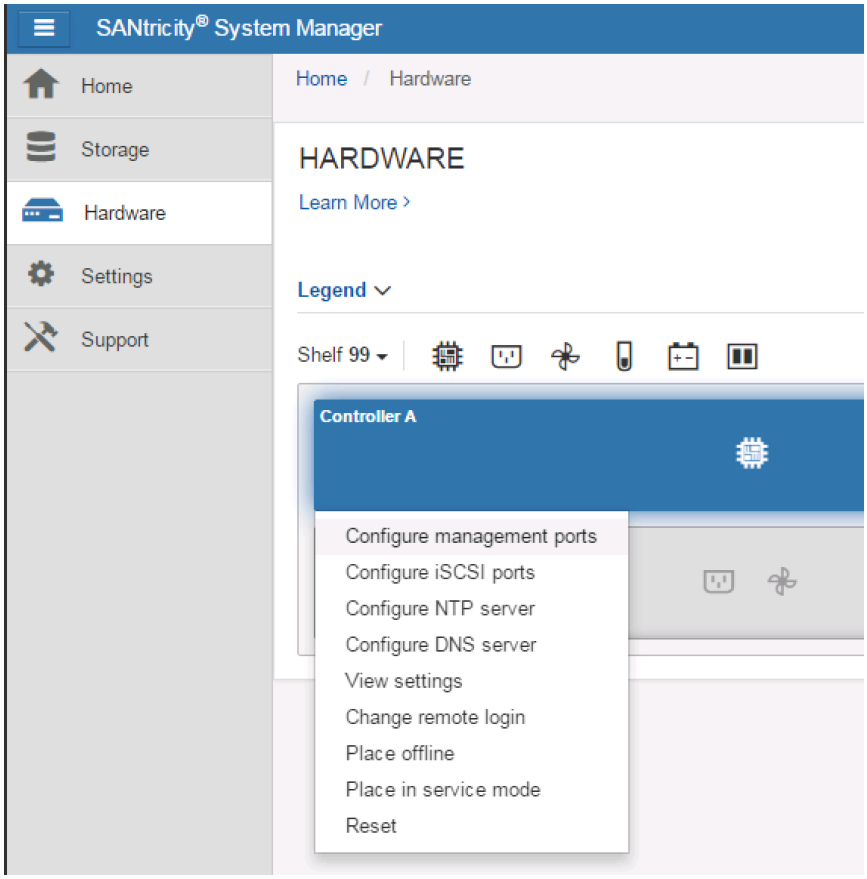
6. Enter a name for the storage array `<<var_e2800_name>>` and check if the hardware components are listed correctly. Click Next.
7. Click Next in the Verify Hosts section.
8. Under Select Applications, select Other. Click Next.



9. Enter a name for the workload `<<var_storage_workload_name>>` and click Next.
10. Select No to decline the recommended pool configuration (you create it later).
11. Click Next.
12. Choose No to decline the creation of hot spares. They are not needed with Dynamic Disk Pools.
13. Click Next.
14. Configure the Alerts section by providing the mail server details and recipient e-mail address.

    Note: Select Do this later if the necessary information is not available.
15. Enable AutoSupport® by selecting the checkbox. Click Next.
16. Review the configuration and click Finish.

17. Click Close. This closes the setup wizard and takes you to the SANtricity home page.
18. In the left pane, click Hardware. In the right pane, click Show back of shelf.
19. Click Controller A and select Configure management ports from the drop-down menu.



20. Select Port P1. Click Next.
21. Leave the speed and duplex mode set to Auto-negotiate.
22. Make sure the Enable IPv6 checkbox is not selected. Click Next.

    Note: This configuration uses IPv4.

23. If a DHCP source is being used, leave the selection set to Automatically obtain configuration from DHCP server. If no DHCP source is being used, select Manually specify static configuration and enter the details:

    ```
    <<var_e2800_A/B_mgmt_ip>>
    <<var_e2800_A/B_mgmt_netmask>>
    <<var_e2800_A/B_mgmt_gw>>
    ```

24. Click Finish.
25. Click Yes to confirm the network settings.

    Note: Connectivity to the storage array is lost during this process. Reconnect to the storage array using the newly configured management IP address.

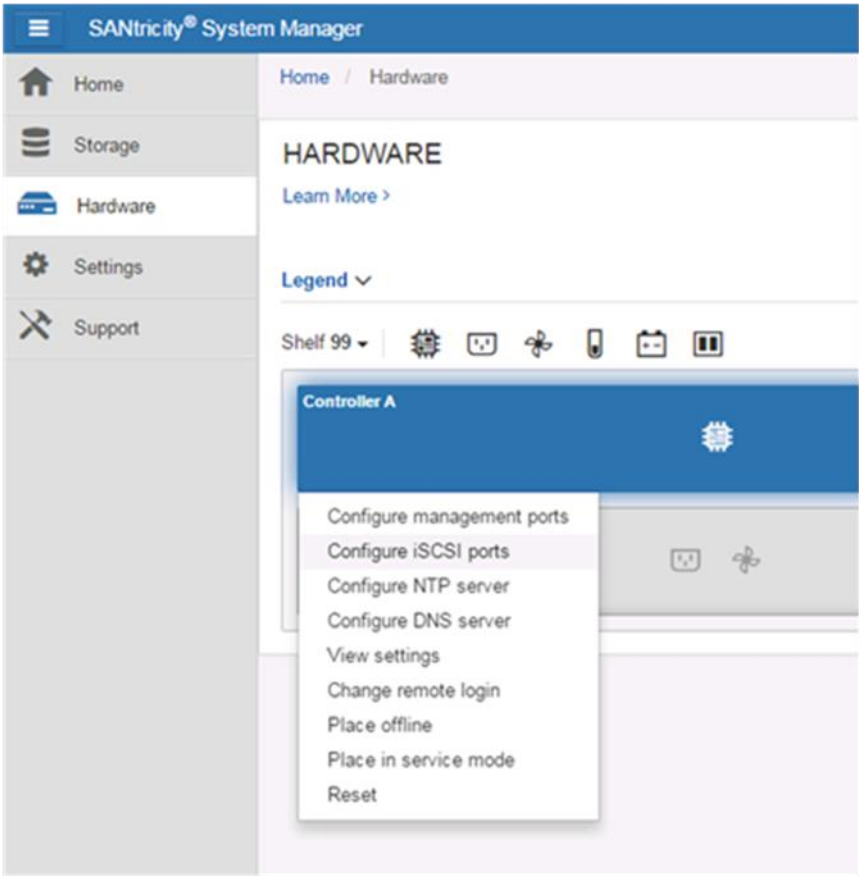
    Note: Change the IP address of the reader workstation to its original configuration.

26. Connect the management ports of controller A and controller B to the network infrastructure.
27. Repeat steps 18 through 25 for configuring the management IP address for controller B.

28. In the left pane, click Hardware. In the right pane, click Show back of shelf.
29. Click Controller A and select Configure NTP server from the drop-down menu.
30. Select the checkbox to enable NTP on controller A.
31. Select the second radio button to manually enter the NTP server address.
32. Enter the NTP server address `<<var_global_ntp_server_ip>>` and click Save.
33. Click Yes to apply the same NTP settings to controller B.
34. In the left pane, click Hardware. In the right pane, click Show back of shelf.
35. Click Controller A and select Configure DNS server from the drop-down menu.
36. Select the second radio button to manually specify the DNS server address.
37. Enter the primary and backup DNS server addresses <<var_dns1_ip>> and <<var_dns2_ip>> and click Save.
38. Click Yes to apply the same DNS settings to controller B.

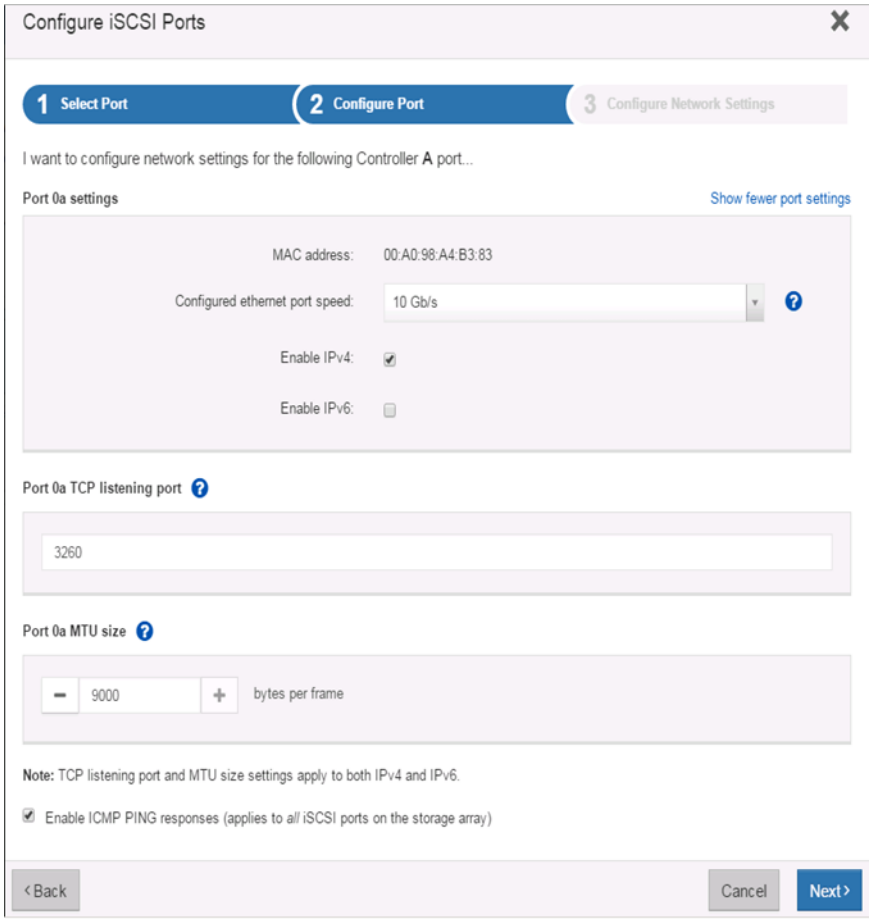
6.10 NetApp E2824 iSCSI Interface Configuration (iSCSI Topology Only)

1. In the left pane, click Hardware. In the right pane, click Show back of shelf.
2. Click Controller A and select Configure iSCSI ports from the drop-down menu.

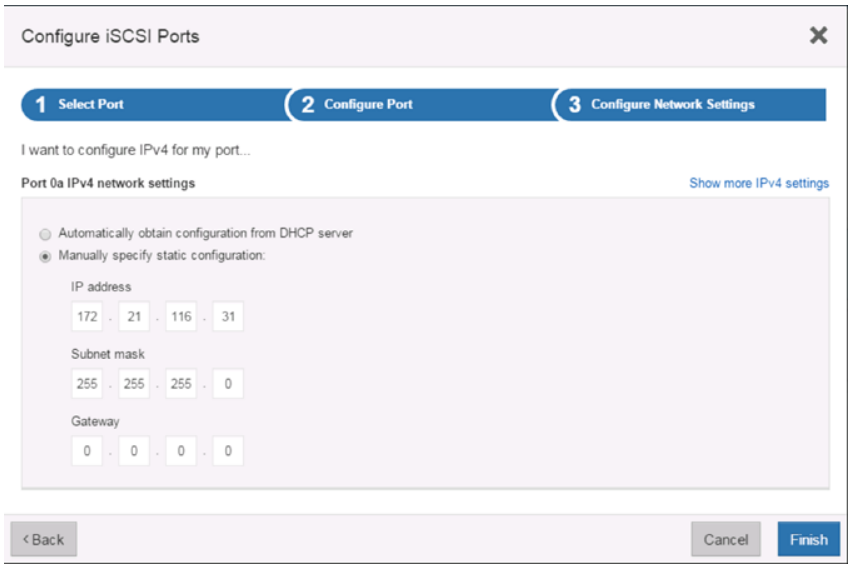


3. Select Port 0a from the drop-down menu and click Next.
4. Click Show more port settings.
5. Set the Ethernet port speed to 10Gbps from the drop-down menu.
6. Make sure Enable IPv6 is not selected.

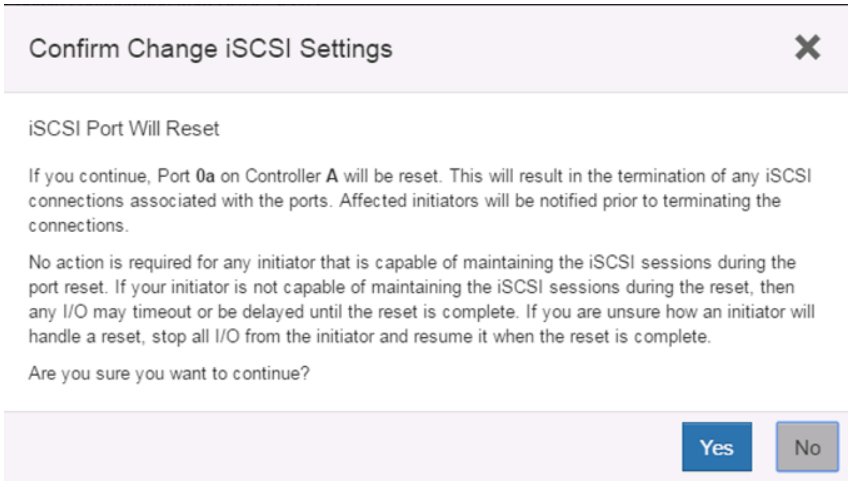
7. Leave the TCP listening port set to default.
8. Set the Port MTU size to 9000.



9. Click Next.
10. Select the Manually specify the static configuration radio button.
11. Enter the iSCSI A VLAN IP address for controller A: `<<var_e2800_A_iSCSI_A_ip>>`.
12. Enter the subnet mask for the iSCSI IP address: `<<var_e2800_A_iSCSI_A_netmask>>`.



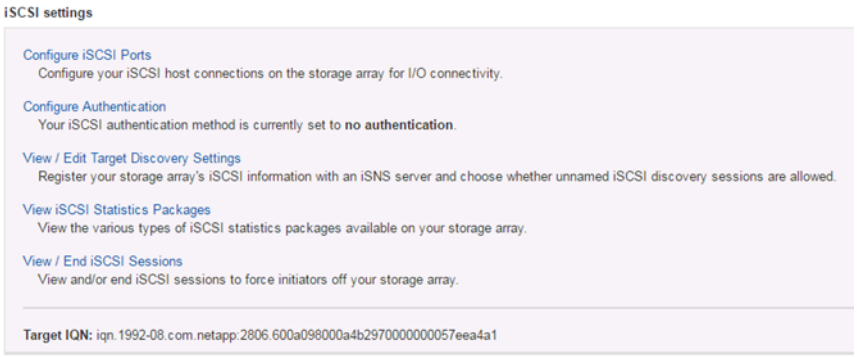
13. Click Finish.



14. Click Yes to confirm the iSCSI settings.

    Note: This process takes a couple of minutes to complete.

15. Repeat steps 1 through 14 for controller A port 0b, with an IP address from iSCSI VLAN B.
16. Repeat steps 1 through 15 for controller B port 0a and 0b, with IP addresses from iSCSI VLAN B (for port 0a) and iSCSI VLAN A (for port 0b).
17. Retrieve the iSCSI target IQN (this is necessary to configure the indexers) by clicking Settings in the left pane.
18. The iSCSI target IQN is listed at the bottom of the iSCSI settings section in the right pane.

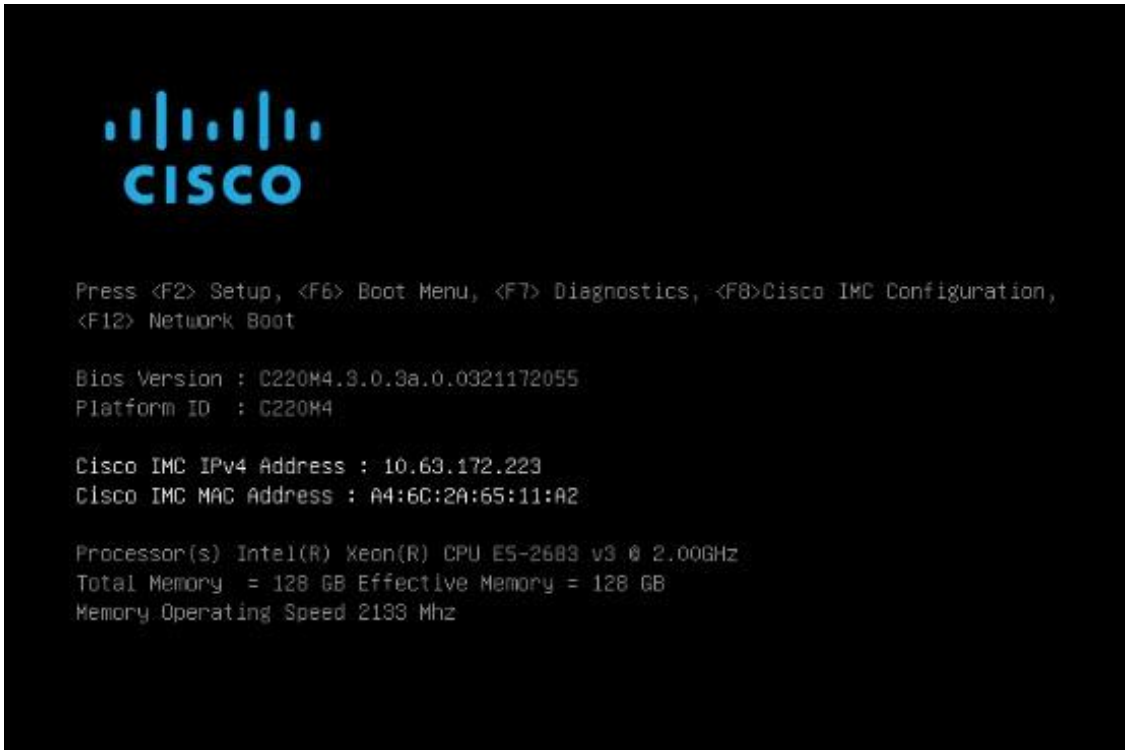


19. Record the iSCSI target IQN (<<var_e2800_iqn>>) for later use in configuring the indexers.

6.11 RHEL 7.3 Installation (All Topologies)

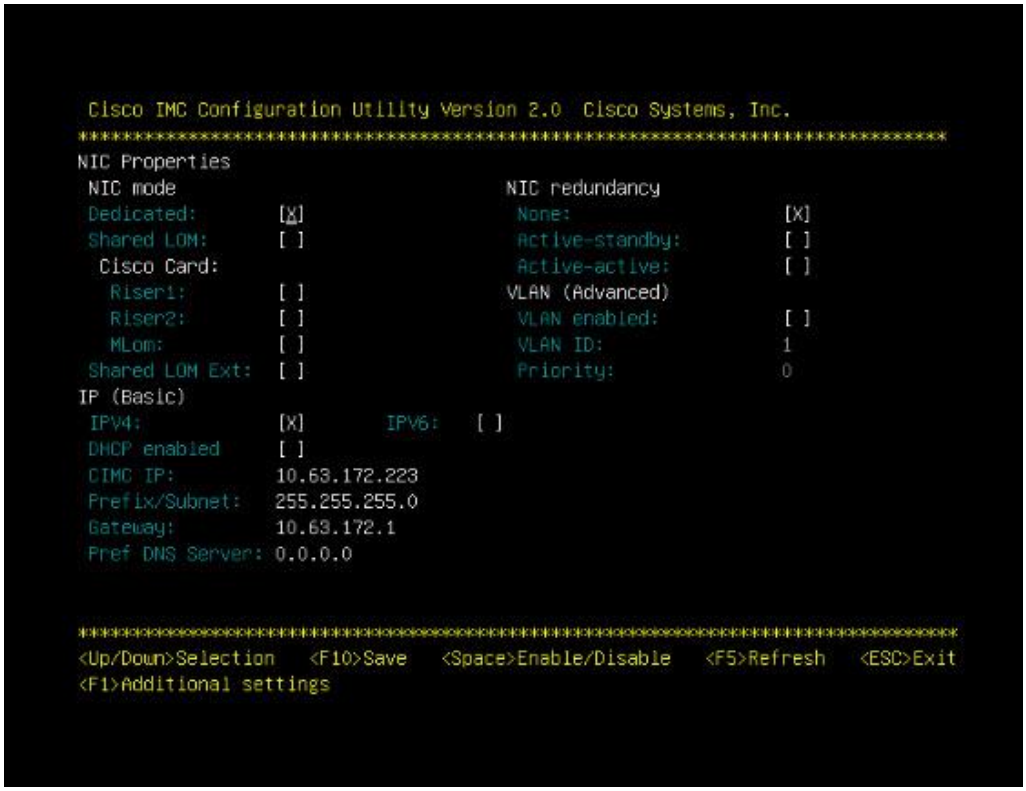
The Cisco UCS C-series servers have a remote management controller, called Cisco IMC (CIMC). One of the capabilities CIMC provides is a KVM, which is how you can install the Red Hat Enterprise Linux (RHEL) operating system. To access the CIMC, locally connect a physical keyboard and monitor to configure the CIMC's IP address. Then log in to the CIMC through a web browser, set the boot order to include a virtual DVD, map the virtual DVD to the RHEL ISO file, and install RHEL. Install RHEL on each server by following these steps:

1. Connect a USB keyboard and VGA monitor to the USB and VGA ports on the server.
2. Turn on the server and press F8 when prompted to enter the Cisco IMC configuration. The factory default CIMC password is "password."



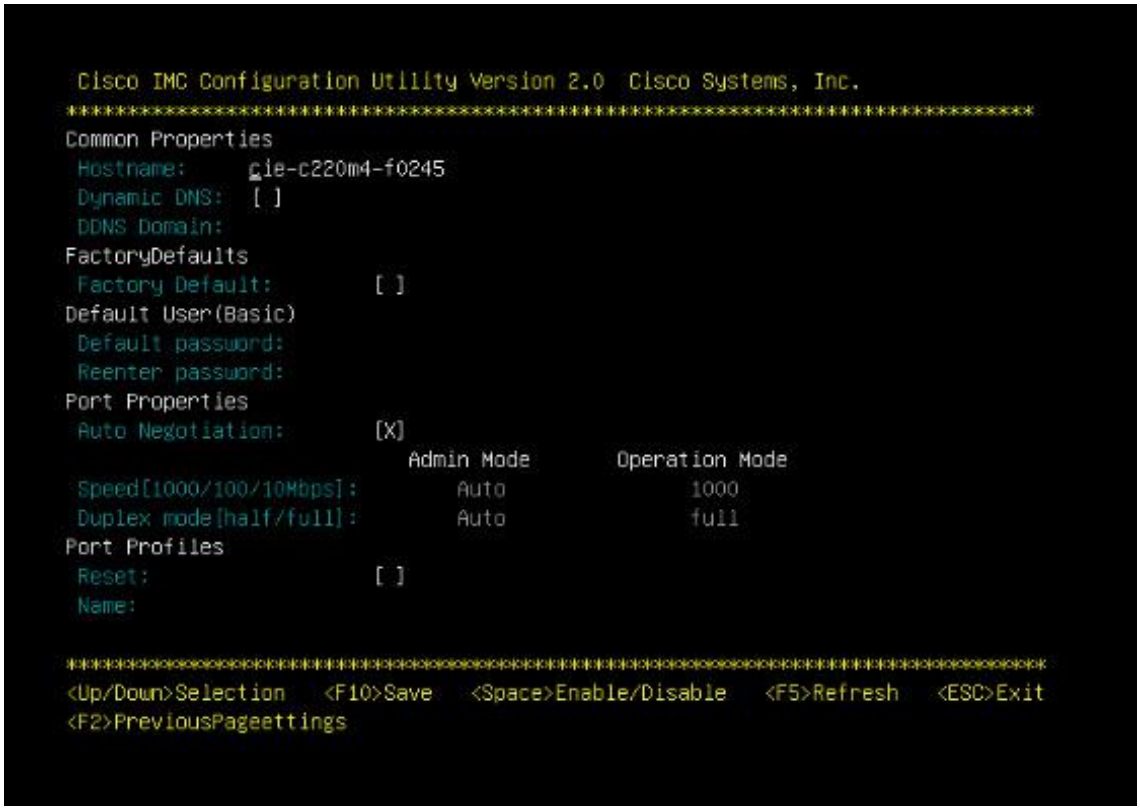
3. In the Cisco IMC configuration utility, set the following options:
   - Network Interface Card (NIC) Mode:

  - Dedicated [X]
- IP (Basic):
  - IPV4: [X]
  - DHCP enabled: Clear this option to disable DHCP.
  - CIMC IP: <<var_indexer1_cimc_ip>>
  - Prefix/Subnet: <<var_indexer1_cimc_netmask>>
  - Gateway: <<var_indexer1_cimc_gw>>
- VLAN (Advanced):
  - Clear this option to disable VLAN tagging.
- NIC Redundancy:
  - None: [X]

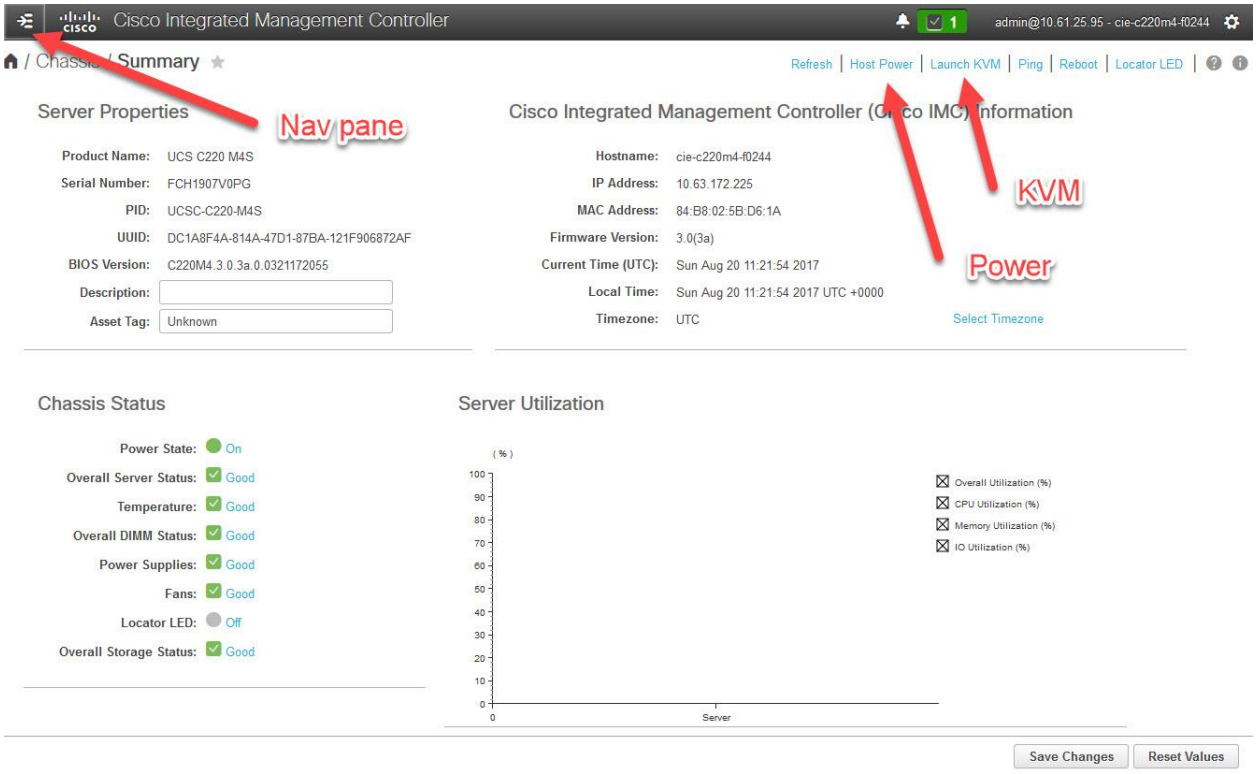


Press F1 to see additional settings:

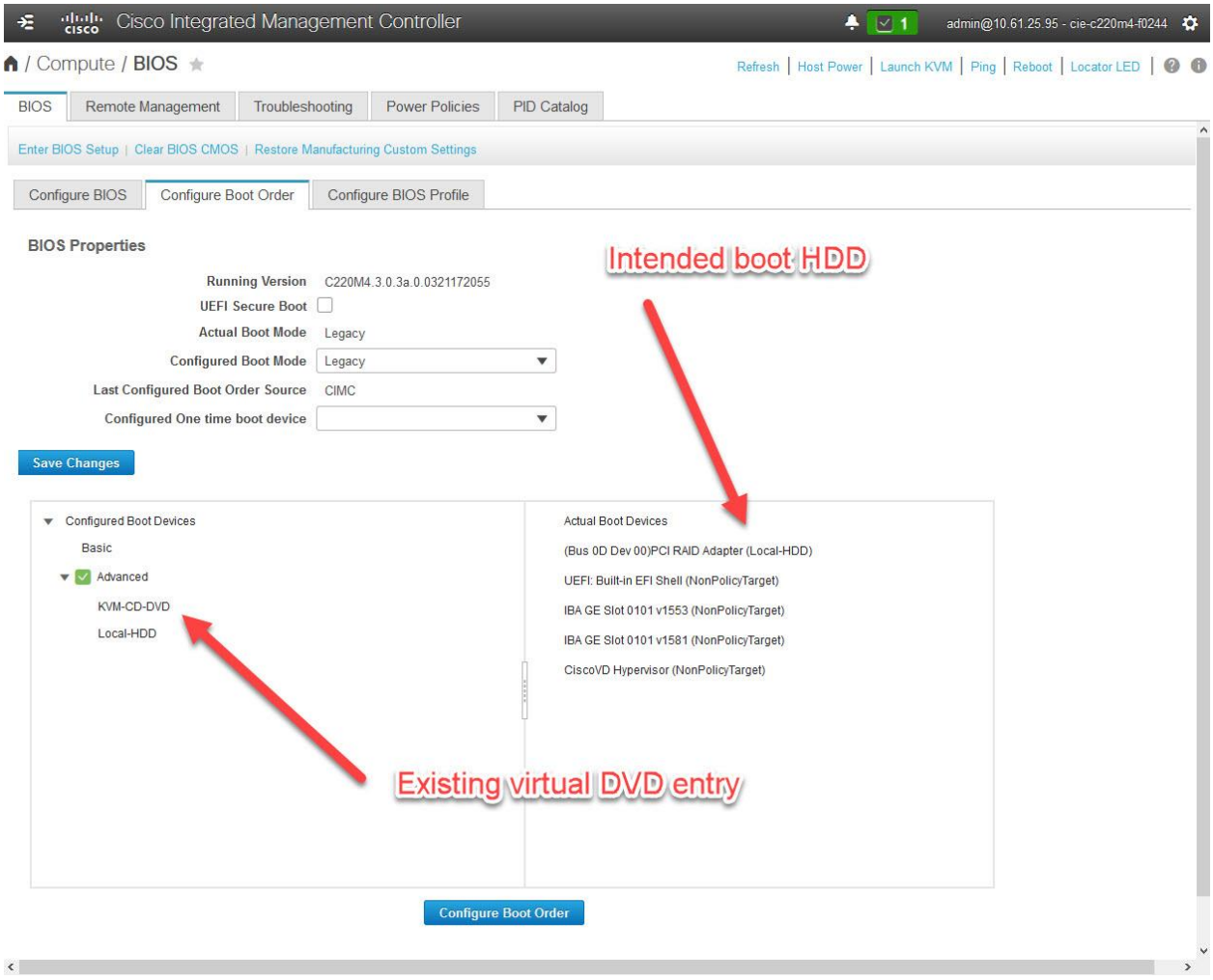
- Common Properties:
  - Host name: (optional)
  - Dynamic DNS: [ ]
- Factory Defaults: Make sure this is not checked.
- Default User (Basic):
  - Default password: <<var_password>>
  - Reenter password: <<var_password>>
- Port Properties: Use default values.
- Port Profiles: Make sure this is not checked.



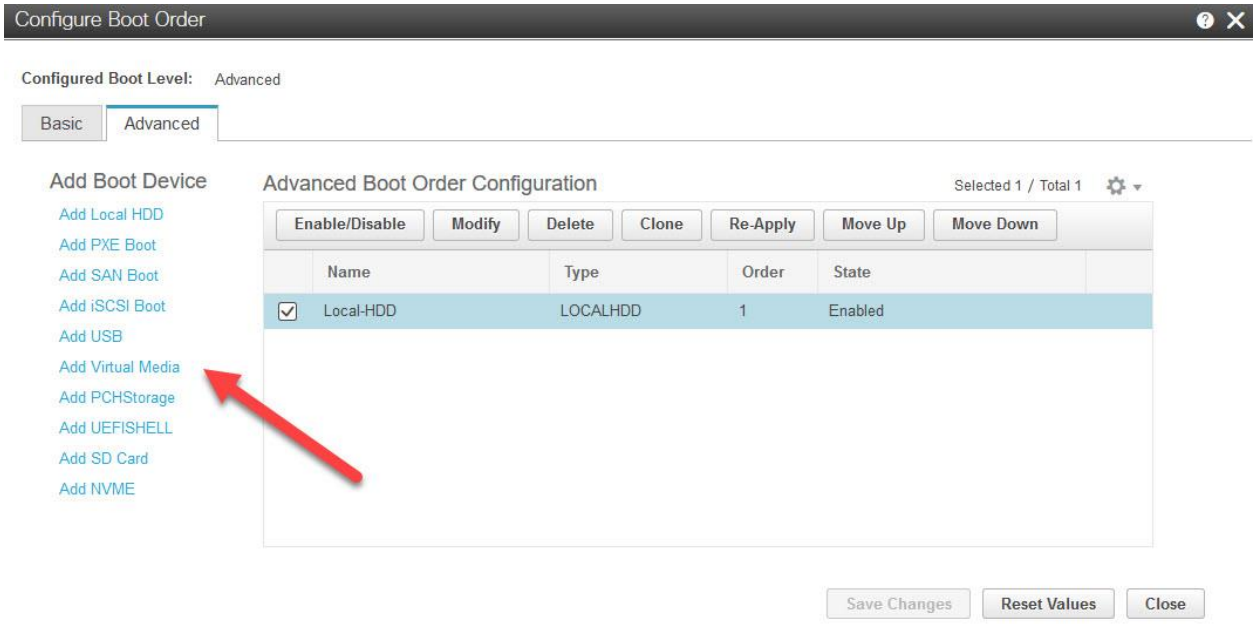
4. Press F10 to save the Cisco IMC interface configuration.
5. After the configuration is saved, press Esc to exit.
6. Disconnect the keyboard and monitor.
7. After the CIMC becomes responsive on its IP address, connect to it through a web browser and log in with the credentials set during the CIMC configuration.



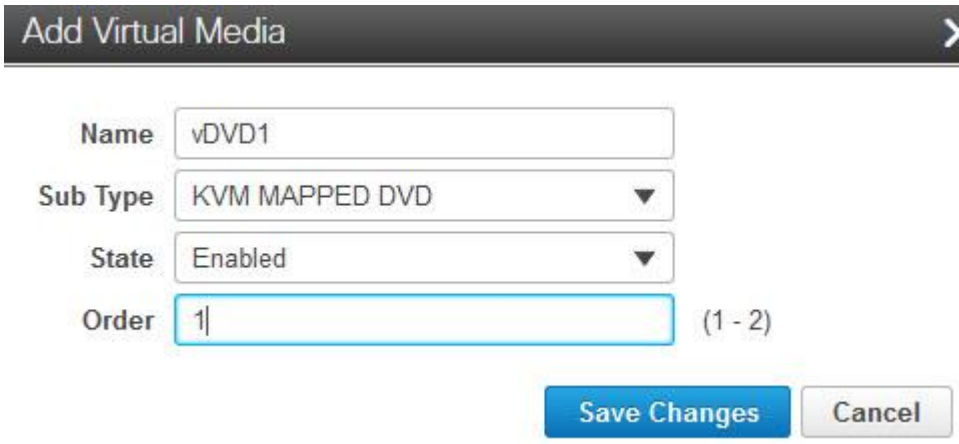
8. From the CIMC home page, expand the Navigation pane by clicking the icon in the upper-left corner.
9. In the Navigation pane, select Compute, then the BIOS main tab, and the Configure Boot Order subtab.



10. If an instance of a virtual DVD appears in the Configured Boot Devices list, then you can skip the step of adding one. (Don't decide based solely on the name of the device: That is just a string, which can be anything. Hover the mouse over it, and the properties are displayed. Look for type VMEDIA, subtype KVMDVD.) Also make sure that the storage device to which you are installing (in this example, "Local-HDD") displays in the Actual Boot Devices list.
11. If a virtual DVD is not in the list, then add one by clicking the Configure Boot Order button at the bottom of the screen. Click the Advanced tab and the "Add Virtual Media" link.



12. In the Add Virtual Media screen, add a name, set the Sub Type to “KVM MAPPED DVD,” set the state to Enabled, and set the order to 1. Click Save Changes.



13. Make sure the virtual DVD is now in the Configured Boot Devices list.
14. Launch a KVM session by clicking the “Launch KVM” link in the upper-right corner. There are two variants of the KVM: Java-based and HTML-based. The HTML-based is faster to open, with fewer mouse clicks. However, HTML-based KVM sessions can be knocked out by browser problems, such as a plug-in crashing in another browser tab (which also disconnects any virtual media sessions that are going through the KVM session). So for quick “check or set” tasks, the HTML-based KVM is a better choice; for installs, the Java-based KVM is a better choice. For this example, we are using the Java-based KVM to install the RHEL. Open a Java-based KVM session.
15. After the KVM window opens, select the Virtual Media menu, then the Activate Virtual Devices item.
16. When the Initializing Connection box disappears, select the Virtual Media menu again, then the Map CD/DVD item. Click Browse and navigate to the RHEL ISO file. Select the file and click Map Device.
17. Select the Boot Device menu, then select the menu item that corresponds to your virtual DVD.

18. Select the Power menu, then select the Power On System item (if the server is powered off) or the Reset System (warm boot) item (if it is already powered on). The server should boot into the RHEL installer.
19. Select the appropriate installation language and click Continue.
20. At the Installation Summary page, customize the installation as desired and resolve any errors. For our lab, our only nondefault selection was “Server with GUI” under Software Selection. The default settings were used for everything else. After you have completed installation selections and resolved any errors, click Begin Installation.
21. When the installation starts, the User Settings screen is displayed. Click ROOT PASSWORD to set the root password. RHEL installation should then complete with no more input until acknowledging the reboot when the installer is done.
22. After the RHEL installation completes, it unmounts/unmaps the install ISO and then reboots the server. After it reboots, accept the license agreement and click Finish Configuration. Do not configure the network at this point.
23. Go through the screens that set up language, time zone, and so on. When you are finished, log in as root.
24. The next step is to update the VIC 1227 drivers. Because the VIC 1227 is a CNA, there are two drivers to update: the enic driver for the Ethernet functionality, and the fnic driver for the FC HBA functionality. For our RHEL 7.3 operating system and Cisco UCS 3.0(3a) firmware, the appropriate drivers are on the version 3.0(3b) Linux drivers ISO, which is named “ucs-cxxx-drivers-linux.3.0.3b.iso.” It is available from Cisco’s website under the C-series software downloads. Mount the driver ISO using the Cisco UCS KVM virtual media, just as you did for the RHEL install ISO. A CDROM icon should appear on the RHEL desktop after the ISO is mounted.
25. Change to the enic driver directory and install the enic driver through RPM. Then change to the fnic directory and install the fnic driver through RPM. That process should look like this:

```
[root@localhost ~]# cd /run/media/root/CDROM/Network/Cisco/VIC/RHEL/RHEL7.3
[root@localhost RHEL7.3]# rpm -ivh kmod-enic-2.3.0.39-rhel7u3.el7.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:kmod-enic-2.3.0.39-rhel7u3.el7   ################################# [100%]
[root@localhost RHEL7.3]# cd /run/media/root/CDROM/Storage/Cisco/VIC/RHEL/RHEL7.3
[root@localhost RHEL7.3]# rpm -ivh kmod-fnic-1.6.0.34-rhel7u3.el7.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:kmod-fnic-1.6.0.34-rhel7u3.el7   ################################# [100%]
[root@localhost RHEL7.3]#
```

26. After the driver update completes, gracefully shut down the server, unmap the driver ISO, and power the server back on.
27. After the server comes back online, log in as root. Open a terminal session and disable the Network Manager tool with “systemctl disable NetworkManager.service.” (Network Manager does not configure some of the parameters needed for the bond interfaces. If it is left enabled, someone could use it and inadvertently break the network config.)
28. Stop the networking service with “systemctl stop network.service.”
29. Change directory to /etc/sysconfig/network-scripts. You should see ifcfg files that correspond to the four physical interfaces (two 1G Ethernet, two 10G Ethernet) in the server. Because you use link aggregation in converged infrastructure, combine these physical interfaces into two “bond” interfaces (the RHEL equivalent of a port channel). The two 1G physical interfaces (eno1 and eno2) go into “bond1g,” which you create now. Recall that the purpose of the 1G ports is management access, not wholesale data movement. The only reason to use two ports for management is redundancy. The appropriate bonding mode would be “active-backup,” as depicted in the following configuration. The corresponding switch port configuration would be access ports (that is, no VLAN header encapsulation) in the management VLAN. We’ll also be putting the management IP settings for this RHEL host on the bond1g interface. Because this interface is for management, it should have the

default gateway and DNS settings. Using your preferred text editor, create a new file "/etc/sysconfig/network-scripts/ifcfg-bond1g," with the following contents:

```
DEVICE=bond1g
NAME=bond1g
TYPE=bond
BONDING_MASTER=yes
BONDING_OPTS="mode=active-backup miimon=1 updelay=0 downdelay=0"
USERCTL=no
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
IPADDR=<<var_indexer1_mgmt_ip>>
NETMASK=<<var_indexer1_mgmt_netmask>>
GATEWAY=<<var_indexer1_mgmt_gw>>
DEFROUTE=yes
PEERDNS=yes
DNS1=<<var_dns1_ip>>
DNS2=<<var_dns2_ip>>
IPV4_FAILURE_FATAL=no
IPV6INIT=no
```

30. Edit the existing "/etc/sysconfig/network-scripts/ifcfg-eno1" file to slave it to bond1g. Edit the file so that it contains the following contents:

```
DEVICE=eno1
NAME=eno1
TYPE=Ethernet
SLAVE=yes
MASTER=bond1g
ONBOOT=yes
BOOTPROTO=none
NM_CONTROLLED=no
```

31. Repeat step 30 for file ifcfg-eno2.
32. Create the bond10g interface. The steps to create it are just like those for bond1g, except for the following, which are depicted in the following config:
    - Uses the 10G ports (enp6s0 and enp7s0), instead of the 1G ports.
    - Uses "802.3ad" mode, instead of "active-backup." (That corresponds to "active" mode port channels on the switches that use the LACP protocol, which was standardized in IEEE 802.3ad.)
    - IPv4 is disabled on the bond10g interface. (You are creating VLAN interfaces on top of bond10g; the IP addresses are applied to the VLAN interfaces.)

```
DEVICE=bond10g
NAME=bond10g
TYPE=bond
BONDING_MASTER=yes
BONDING_OPTS="mode=802.3ad miimon=10 lacp_rate=1"
USERCTL=no
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
IPV6INIT=no
```

33. Repeat steps 30 and 31 to slave ports enp6s0 and enp7s0 to bond10g.
34. Create the VLAN interfaces:
    - The VLAN interfaces are children of the bond10g interface. Their device name is "bond10g.<vlan_number>," and their "ifcfg-…" file name is "ifcfg-bond10g.<vlan_number>." For example, the file name for the VLAN 99 interface would be "ifcfg-bond10g.99."
    - Assign a VLAN number (in this example, VLAN 99 for the Splunk traffic VLAN).

- In the IPv4 settings for this example, you do not include either a gateway or DNS settings. That is because this is a private VLAN, solely for Splunk search and replication traffic. There is no need to route traffic on or off of this VLAN. That is the same for the two iSCSI VLANs you are creating.

```
DEVICE=bond10g.<<var_splunk_traffic_vlan >>
TYPE=Vlan
VLAN=yes
USERCTL=no
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPADDR=<<var_indexer1_splunk_ip>>
NETMASK=<<var_indexer1_splunk_mask>>
```

35. The search heads and forwarders only have one VLAN interface each, on the Splunk VLAN (VLAN 99). The indexers have three VLAN interfaces: the Splunk VLAN, the iSCSI-A VLAN, and the iSCSI-B VLAN. Repeat step 34 to create all of these VLAN interfaces.
36. After all of the interfaces are created and configured with IP, restart the networking service with "systemctl start network.service."
37. Run ping tests to make sure that all addresses are reachable.

6.12 iSCSI Session Creation (iSCSI Topology Only)

1. Determine the MAC address of the ports being used for iSCSI on the first indexer using "ifconfig -a," which outputs information similar to the following:

```
[root@localhost]# ifconfig -a
iSCSI-A: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet <<var_indexer1_iSCSI_A_ip>>  netmask <<var_indexer1_iSCSI_A_netmask>>
        ether <<var_indexer1_iSCSI_A_mac>>  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 2789 (2.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 3937 (3.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. Determine the initiator name for the first indexer using "cat /etc/iscsi/initiatorname.iscsi":

```
[root@localhost]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=<<var_indexer1_initiator>>
```

3. Create an iface for the first port using the following commands:

```
iscsiadm -m iface -I ifaceA -o new
iscsiadm -m iface -I ifaceA -o update -n iface.hwaddress -v <<var_indexer1_iSCSI_A_mac>>
iscsiadm -m iface -I ifaceA -o update -n iface.ipaddress -v <<var_indexer1_iSCSI_A_ip>>
iscsiadm -m iface -I ifaceA -o update -n iface.transport_name -v tcp
iscsiadm -m iface -I ifaceA -o update -n iface.initiatorname -v <<var_indexer1_initiator>>
```

Here "ifaceA" is the user-modifiable name of the iface file.

4. Create a separate iface file for the first port to handle the use of VLANs using the following commands:

```
iscsiadm -m iface -I ifaceA_vlan -o new
iscsiadm -m iface -I ifaceA_vlan -o update -n iface.net_ifacename -v ifaceA
iscsiadm -m iface -I ifaceA_vlan -o update -n iface.vlan_id -v <<var_iscsi_a_vlan>>
iscsiadm -m iface -I ifaceA_vlan -o update -n iface.vlan_state -v enable
```

Here "ifaceA_vlan" is the user-modifiable name of the iface VLAN file.

5. Repeat the preceding commands for the second port.
6. Use the following command to discover all of the connected iSCSI ports on the E2824.

```
iscsiadm -m discovery -t st -p <<var_e2800_A_iSCSI_A>> -I ifaceA
```

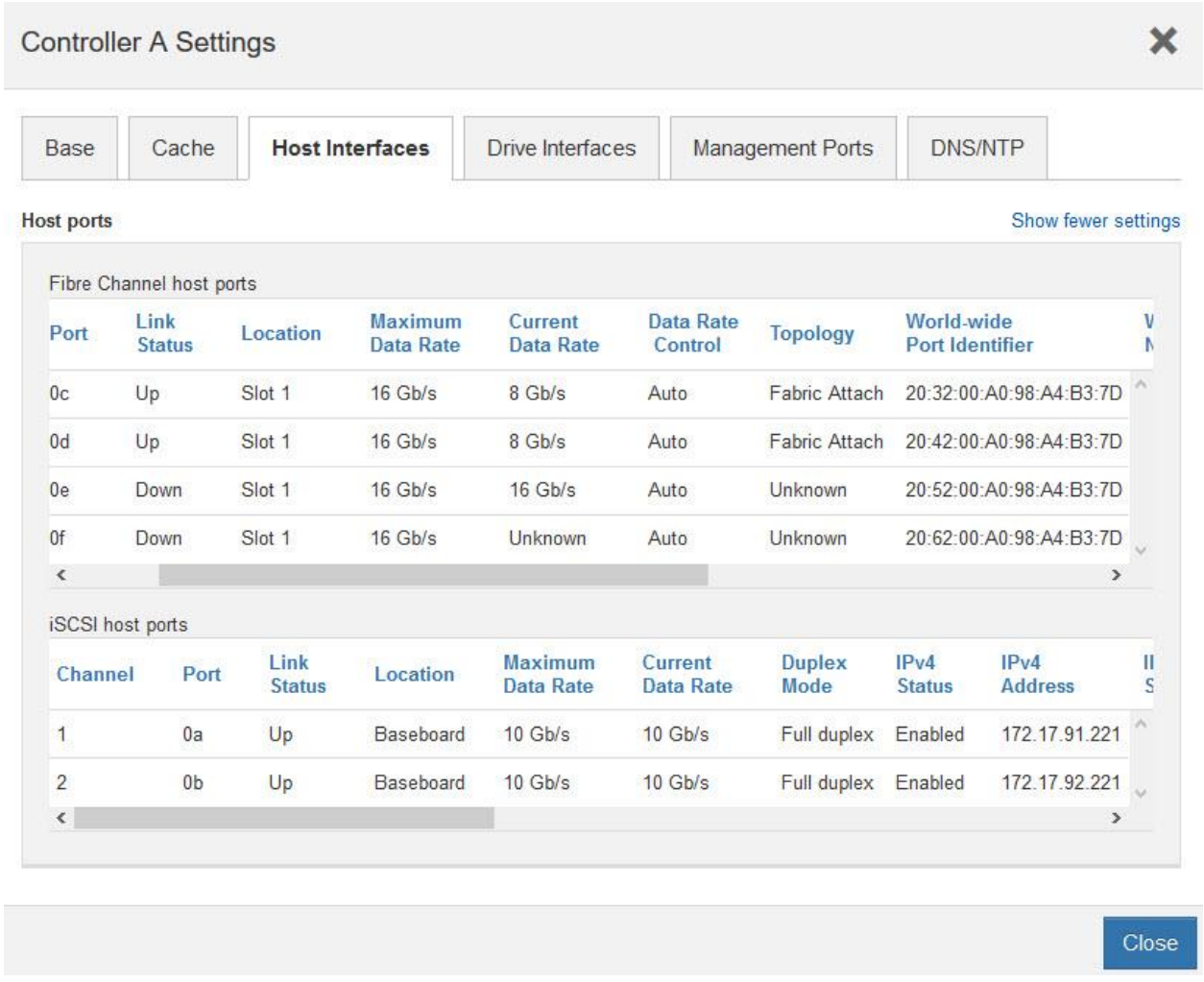

7. Log in to iSCSI sessions using the following commands:

```
iscsiadm -m node -T <var_e2800_iqn> -p <<var_e2800_A_iSCSI_A_ip>> -I ifaceA -l
iscsiadm -m node -T <var_e2800_iqn> -p <<var_e2800_B_iSCSI_A_ip>> -I ifaceA -l
iscsiadm -m node -T <var_e2800_iqn> -p <<var_e2800_A_iSCSI_B_ip>> -I ifaceB -l
iscsiadm -m node -T <var_e2800_iqn> -p <<var_e2800_B_iSCSI_B_ip>> -I ifaceB -l
```

6.13 FC Zoning Configuration (FC and FCoE Network Topologies Only)

Next, set up zoning on each fabric. Zoning works like an access list: The switch only forwards traffic between WWPNs (FC addresses) that are in the same zone. First, gather the WWPNs from both of the controllers in the E2824 and all three indexers. The WWPNs are set at the factory and are globally unique.

1. Open the home screen in SANtricity.
2. In the left pane, click Hardware. In the right pane, click Show back of shelf.
3. Click Controller A and select View settings from the drop-down menu.
4. Select the Host Interfaces tab in the Controller A Settings pop-up menu.
5. Click the “Show more settings” link.



6. The WWPNs are listed in the “World-wide Port Identifier” column. The rows correspond to the interfaces. For example, in the preceding graphic, the WWPN for port 0c is listed as

20:32:00:A0:98:A4:B3:7D. Note the WWPNs for all of the ports you have connected to the FC switches (in our lab 0c is connected to fabric A, and 0d fabric B, for both controllers).

7. Repeat steps 3 through 6 for controller B.
8. To retrieve the WWPNs from the indexers, SSH into the RHEL installation as root and run the following command to list the FC interfaces:

```
[root@localhost ~]$ ls /sys/class/fc_host
host1  host10  host2  host9
[root@localhost ~]$
```

9. This particular server (indexer1) has four FC interfaces: two that correspond to the vHBAs on the VIC 1227 and two on the Qlogic card. You can figure out which is which by retrieving the WWPNs for each port and checking which WWPNs are logged in on the MDS (for the native FC variant) or the Cisco Nexus 5000 series (for the FCoE variant):

```
[root@localhost ~]$ more /sys/class/fc_host/host1/port_name
0x2000cc167e1f1bcf
[root@localhost ~]$ more /sys/class/fc_host/host2/port_name
0x2100000e1e163180
[root@localhost ~]$ more /sys/class/fc_host/host9/port_name
0x2000cc167e1f1bd0
[root@localhost ~]$ more /sys/class/fc_host/host10/port_name
0x2100000e1e163181
[root@localhost ~]$
```

10. You can see which WWPNs are logged in to each port on the MDS (or Cisco Nexus 5000) using the "show flogi database" command. Here is the result of that command, on the fabric A MDS:

```
cie-c9148-f1525# show flogi database
--------------------------------------------------------------------------------
INTERFACE        VSAN    FCID           PORT NAME               NODE NAME
--------------------------------------------------------------------------------
fc1/1            101   0x960000  21:00:00:0e:1e:16:31:80 20:00:00:0e:1e:16:31:80
fc1/2            101   0x960200  21:00:00:0e:1e:15:db:10 20:00:00:0e:1e:15:db:10
fc1/3            101   0x960300  21:00:00:0e:1e:16:34:e0 20:00:00:0e:1e:16:34:e0
fc1/11           101   0x960100  20:32:00:a0:98:a4:b3:7d 20:02:00:a0:98:a4:b3:7d
fc1/12           101   0x960400  20:33:00:a0:98:a4:b3:7d 20:02:00:a0:98:a4:b3:7d

Total number of flogi = 5.

cie-c9148-f1525#
```

11. The WWPN (port name) logged in to the fabric A MDS port fc1/1 matches that for "host2" on indexer1. (WWPNs are typically expressed as 16-digit hexadecimal; the depiction from RHEL uses lowercase letters, while the depiction from SANtricity uses uppercase letters and inserts colons to help readability. The MDS uses lowercase letters, with colons. All of these depictions are valid WWPNs.) In our lab, port 1 from each indexer is connected to fabric A, and port 2 is connected to fabric B.
12. Repeat steps 8 through 11 to identify the remaining WWPNs.
13. You have three zones on each fabric, one for each indexer. Each of the six zones contains three WWPNs: one for the indexer (the initiator) and one for each of the two storage controllers (the targets). Because the WWPNs are tedious to work with, define aliases to help keep things straight. Finally, designate a zoneset, which makes it easier to activate the three zones as a group. Use the following commands to implement that on switch 1 (fabric A):

```
config t
device-alias database
device-alias name Indexer1_A pwwn <<var_indexer1_A_wwpn>>
device-alias name Indexer2_A pwwn <<var_indexer2_A_wwpn>>
device-alias name Indexer3_A pwwn <<var_indexer3_A_wwpn>>
device-alias name E2800-A_A pwwn <<var_e2800_A_A_wwpn>>
device-alias name E2800-B_A pwwn <<var_e2800_B_A_wwpn>>
exit
device-alias commit
zone name Indexer1_A vsan <<var_vsan_a_id>>
```



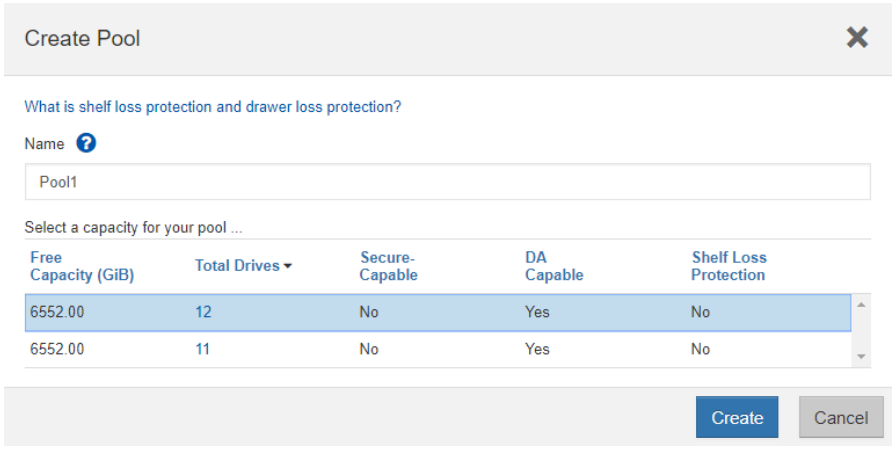
```
member device-alias Indexer1_A
member device-alias E2800-A_A
member device-alias E2800-B_A
exit
zone name Indexer2_A vsan <<var_vsan_a_id>>
member device-alias Indexer2_A
member device-alias E2800-A_A
member device-alias E2800-B_A
exit
zone name Indexer3_A vsan <<var_vsan_a_id>>
member device-alias Indexer3_A
member device-alias E2800-A_A
member device-alias E2800-B_A
exit
zoneset name Splunk vsan <<var_vsan_a_id>>
member Indexer1_A
member Indexer2_A
member Indexer3_A
exit
zoneset activate name Splunk vsan <<var_vsan_a_id>>
end
copy run start
```

14. Repeat step 13 for switch 2 (fabric B).
15. After the zoning is configured, you should see the indexers' WWPNs in the initiators list on the E2824. If not, you can check the fabric login status of each indexer (initiator) and storage controller (target) with the "show flogi database" command on each switch and the zoning with the "show zone active" command. Verify that the WWPNs logged in on each fabric are the same ones used to configure the zoning on that fabric.

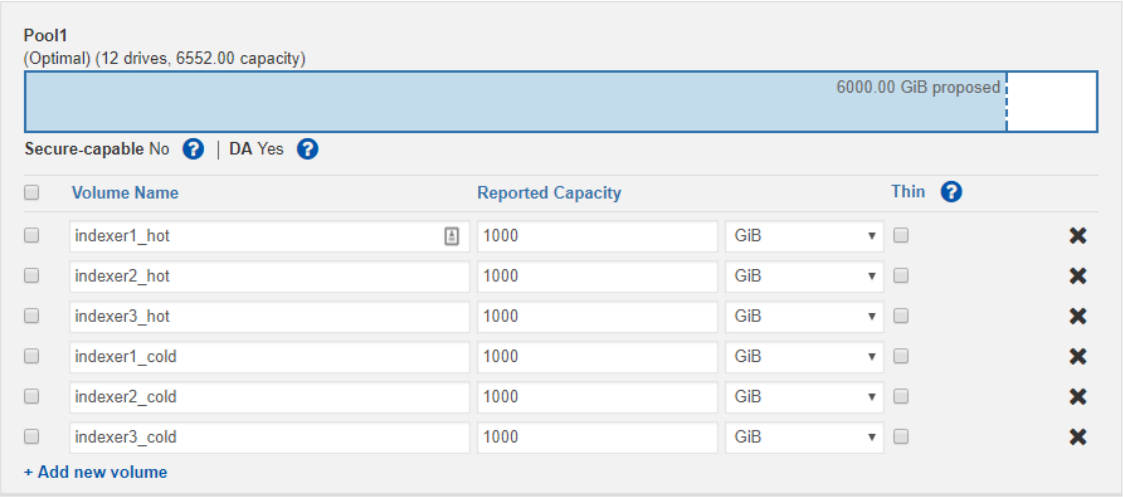
6.14 NetApp E2824 Storage Configuration (All Topologies)

You need to configure the E2824 with a hot/warm volume and a cold volume for each indexer. Then you can map each volume to its appropriate host for mounting.

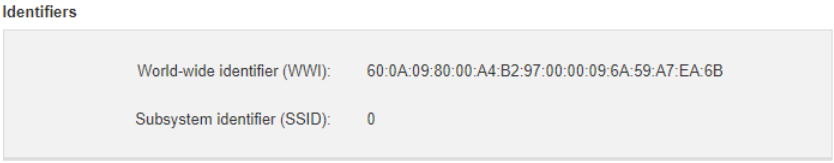
1. From the left pane, click Storage and then select the "POOLS & VOLUME GROUPS" tile.
2. From the Create drop-down menu, select Pool.
3. Type the desired name in the Name field and create the pool using all available drives.



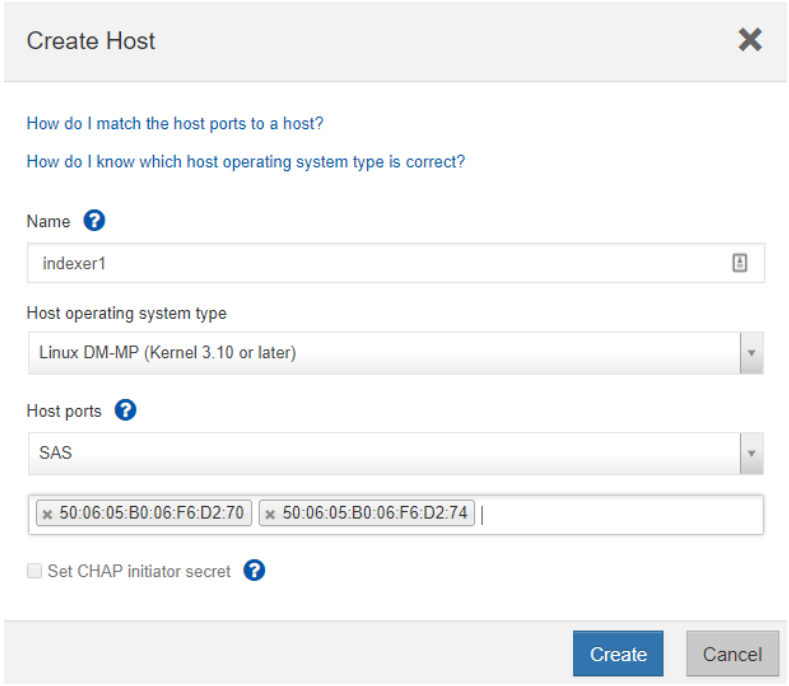
4. From the Create drop-down menu, select Volumes.
5. Select "Assign host later" and <<var_storage_workload_name>> for the workload.
6. Add six appropriately named volumes, each with approximately one-sixth of the available pool space.



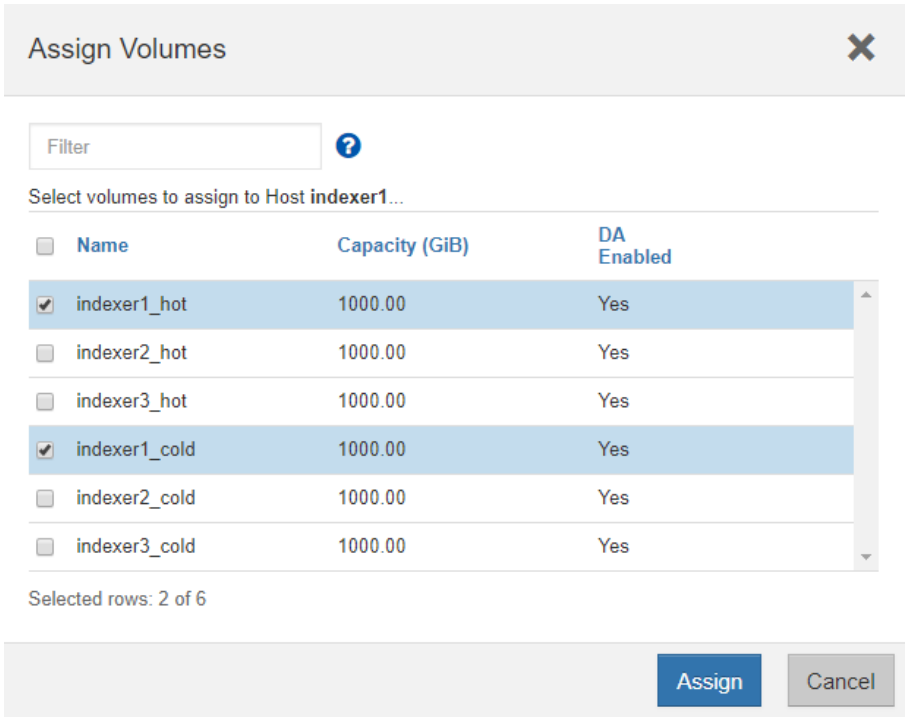
7. After adding the volumes, choose indexer 1's hot volume and click View/Edit Settings.
8. Record the "World-wide identifier (WWI)" as <<var_indexer1_hot_wwi>.



9. Repeat the previous step for the remaining volume.
10. From the left pane, click Storage and then select the "HOSTS" tile.
11. From the Create drop-down menu, select Host.
12. Type <<var_indexer1_name>> in the Name field and choose "Linux DM-MP (Kernel 3.10 or later)" for the "Host operating system type."
13. Depending on the topology you are using, select "iSCSI," "FC," or "SAS" for the I/O interface.
14. Enter the appropriate identifiers for indexer 1 according to the topology you are using. For network-attached topologies, this information is already recorded in Table 6. For FC direct-attached, identifiers are located in "/sys/class/fc_host/[number]/port_id." For SAS direct-attached, identifiers are located in "/sys/class/sas_host/[number]>>/device/scsi_host/[number]." In the following example, we use the two SAS WWPNs associated with Indexer 1.



15. Click the first indexer and click Assign Volumes.
16. Select the boxes next to <<var_indexer1_hot_volume>> and <<var_indexer1_cold_volume>> and click Assign.



17. Repeat the previous steps for the remaining indexers.

6.15 RHEL 7.3 Storage Mapping

You must now mount the volumes created on the E-Series storage array. Repeat the following steps for each indexer.

1. Reboot Indexer 1.
2. SSH into Indexer 1 and use “multipath -ll” to make sure that it sees the created volumes.

```
[root@inc10333952 ~]# multipath -ll
3600a098000a4b37d00000c6459a7eb0b dm-2 NETAPP  ,INF-01-00
size=1000G 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
| `- 0:0:0:1 sdb 8:16 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  `- 0:0:1:1 sde 8:64 active ready running
3600a098000a4b2970000096a59a7ea6b dm-3 NETAPP  ,INF-01-00
size=1000G 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
| `- 0:0:1:2 sdf 8:80 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  `- 0:0:0:2 sdc 8:32 active ready running
```

3. RHEL multipathing identifies the volumes according to the following convention:

   Hot/warm volume: /dev/mapper/3<<var_indexer1_hot_wwi>>

   Cold volume: /dev/mapper/3<<var_indexer1_cold_wwi>

4. Use “fdisk” to create a partition on the first hot volume using default settings.
5. Use “partprobe” to reread the partition table.
6. Use “fdisk -l” to determine the name of the created partition : <<var_indexer1_hot_partition>>.
7. Use “mkfs.ext4” to create an ext4 file system on the created partition.

```
[root@inc10333952 ~]# fdisk /dev/mapper/3<<var_indexer1_hot_wwi>
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-2097151999, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-2097151999, default 2097151999):
Using default value 2097151999
Partition 1 of type Linux and of size 1000 GiB is set
Command (m for help): w

[root@inc10333952 ~]# partprobe
[root@inc10333952 ~]# fdisk /dev/mapper/3<<var_indexer1_hot_wwi> -l

Disk /dev/mapper/3<<var_indexer1_hot_wwi>>: 1073.7 GB, 1073741824000 bytes, 2097152000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xdbc79101

                                      Device Boot      Start         End      Blocks   Id
System
<<var_indexer1_hot_partition>>            2048  2097151999  1048574976   83

[root@inc10333952 ~]# mkfs.ext4 <<var_indexer1_hot_partition>>
```

8. Choose or make a mount point for the hot/warm volume: <<var_hot_mount>>. In this example, mount the hot/warm volume to “/indexes/hot.”

```
[root@inc10333952 /]# mkdir <<var_hot_mount>>
```

9. Modify “/etc/fstab” to include mount information for the partition:

```
<<var_indexer1_hot_partition>>   <<var_hot_mount>>   ext4   defaults   0 0
```

10. Mount the partition.

```
[root@inc10333952 /]# mount <<var_indexer1_hot_partition>>
```

11. Repeat the previous steps for the cold volume.

6.16 Splunk Installation and Configuration

All referenced Splunk documentation can be found at Splunk>docs.

1. Perform a full install of Splunk on the forwarder/cluster master, search head, and indexers according to the Splunk Installation Manual documentation.
2. Configure the forwarder to forward data to the indexers according to the instructions in the Forwarding Data documentation.
3. Configure the forwarder as cluster master with the three indexers as peer nodes according to the Managing Indexers and Clusters of Indexers documentation.
4. Add the search head to the cluster according to the Managing Indexers and Clusters of Indexers documentation.
5. Create a master indexes.conf file to distribute to the indexers according to the Managing Indexers and Clusters of Indexers documentation. Be sure to define the “homepath” and “coldPath” of any index in the indexes.conf file as <<var_hot_mount>> and <<var_cold_mount>>, respectively, so that data is indexed to the volumes on the NetApp E2824. In this deployment, create an eseries index for the SANtricity Performance App for Splunk Enterprise to use to index data on the NetApp E2824 array.
6. Distribute the configuration bundle according to the Managing Indexers and Clusters of Indexers documentation.
7. Modify the inputs.conf file on the forwarder to collect data to send to the indexers according to the Forwarding Data documentation. In this deployment, use the Technology Add-On for NetApp SANtricity to automatically configure the inputs.conf file to collect data on the NetApp E2824.
8. Verify that the data in the indexer cluster is searchable from the search head. In this deployment, use the SANtricity Performance App for Splunk Enterprise to generate dashboards useful for viewing data on the E2824 in our deployment.

6.17 SANtricity Performance App for Splunk Enterprise Setup

In this deployment, we use the Technology Add-On for NetApp SANtricity and the SANtricity Performance App for Splunk Enterprise to verify that our clustered Splunk environment is operating correctly. NetApp recommends installing these tools in any Splunk environment that involves E-Series hardware. Both tools can be found and downloaded from splunkbase. Complete documentation for the install can be found in the Technology Add-On for NetApp SANtricity.

The Technology Add-On for NetApp SANtricity is installed on the forwarder and configured to collect data about deployed arrays monitored by an instance of NetApp Web Services, which must also be installed somewhere in the Splunk deployment. In this deployment, install NetApp Web Services on the same forwarder that you use for data collection.

1. Download NetApp Web Services and install it on the forwarder according to the NetApp SANtricity Web Services Proxy Installation Guide.
2. SSH into the forwarder and navigate to the install directory of NetApp Web Services (/opt/netapp/santricity_web_services_proxy by default).
3. Modify the “wsconfig.xml” file with the following values to enable performance data collection:

```
<env key="stats.poll.interval">30</env>
<env key="stats.poll.save.history">1</env>
```


4. Install the Technology Add-On for NetApp SANtricity using the Splunk web interface on the forwarder.
5. SSH into the forwarder and navigate to the binary files in the install directory for the Technology Add-On for NetApp SANtricity (/opt/splunk/etc/apps/TA-netapp_eseries/bin by default).
6. Execute the “Add_Array.sh” script with the following parameters:

```
[root@localhost TA-netapp_eseries]# ./Add_Array.sh <<var_forwarder1_mgmt_ip>>:8443
<<var_e2800_A_mgmt_ip>> <<var_e2800_B_mgmt_ip>> rw rw
```

   Here `rw` and `rw` are the default user name and password, and 8443 is the default management port of NetApp Web Services.
7. Create an eseries index in the master indexes.conf file and distribute the modified configuration bundle to the cluster using the Splunk web interface on the cluster master (see the Managing Indexers and Clusters of Indexers documentation referenced earlier for more information).
8. Install the SANtricity Performance App for Splunk Enterprise using the Splunk web interface on the search head.
9. Restart Splunk on the search head.
10. Navigate to the SANtricity Performance App for Splunk Enterprise on the search head. Configuration data from the NetApp E2824 should already populate the app. It might take up to an hour for performance data to appear, but the data continues to populate in real time after it does.

7 Solution Verification

For each architecture, Splunk should successfully collect and forward data from the forwarder to the clustered indexers. Each indexer should index the received data on the appropriate hot/warm volume on the E-Series storage array. After the configured amount of time, Splunk should move data buckets from each hot/warm volume to the appropriate cold volume on the E-Series storage array. The indexer cluster should replicate the indexes to maintain its search and replication factors at all times. The data indexed on the E-series storage array in both the hot/warm and cold tiers should be searchable from the search head.

7.1 Volume Mapping

Table 8) Data forwarding and indexing.

Test Case	Details
Test number	Volume Mapping-1
Date	Multiple architectures: 8/17/17, 8/18/17, 8/24/17
Test prerequisites	We connected each peer node indexer to the E-Series storage array and configured each connection as described earlier. We created a hot/warm volume and a cold volume for each peer node indexer using SANtricity System Manager. We mapped each volume to the appropriate peer node indexer using SANtricity System Manager.
Expected outcome	We expected to see a hot/warm volume and a cold volume using “multipath -ll” on each peer node indexer. We expected to be able to create a partition and file system on each volume using “fdisk” and “mkfs.” We expected to be able to mount each volume to its associated indexer by modifying “/etc/fstab.” We expected to be able to open the mount point for each volume and create an eseries directory for the Splunk eseries index.
Test results	Passed
Comments	

7.2 Data Forwarding and Indexing

Table 9) Data forwarding and indexing.

Test Case	Details
Test number	Data Forwarding and Indexing-1
Date	Multiple architectures: 8/17/17, 8/18/17, 8/24/17
Test prerequisites	We completed the Volume Mapping test case and observed that one hot/warm and one cold volume from the E-Series storage array were mapped to each peer node indexer. We installed and configured Splunk according to the Splunk Enterprise 6.6.2 Installation Manual on the forwarder/cluster master, peer node indexers, and search head. We configured Splunk to index eseries data on the appropriate volumes. We installed the Technology Add-On for NetApp SANtricity on the forwarder and configured it to collect data from the E-Series storage array through NetApp Web Services Proxy.
Expected outcome	We expected the eseries index on each peer node indexer's hot/warm volume to fill with one hot bucket following the "hot_v1_<localid>" naming convention and multiple warm buckets following the "db_<newest_time>_<oldest_time>_<localid>_<guid>" naming convention. We expected the cluster master to report an increasing bucket count on the cluster's eseries index.
Test results	Passed
Comments	In our deployment, the cluster master is on the same host as the forwarder.

7.3 Index Replication

Table 10) Index replication.

Test Case	Details
Test number	Index Replication-1
Date	Multiple architectures: 8/17/17, 8/18/17, 8/24/17
Test prerequisites	We completed the Data Forwarding and Indexing test case and observed that E-Series array data was being forwarded and indexed correctly.
Expected outcome	We expected the eseries index on each peer node indexer's hot/warm volume to fill with multiple replicated buckets following the "rb_<newest_time>_<oldest_time>_<localid>_<guid>" naming convention. We expected the cluster master to report that the search and replication factors for the cluster's "eseries" index were met.
Test results	Passed
Comments	In our deployment, the cluster master is on the same host as the forwarder.

7.4 Warm to Cold Rollover

Table 11) Warm to cold rollover.

Test Case	Details
Test number	Warm to Cold Rollover-1
Date	Multiple architectures: 8/17/17, 8/18/17, 8/24/17
Test prerequisites	We completed the Data Forwarding and Indexing test case and observed that E-Series array data was being forwarded and indexed correctly. We configured the “indexes.conf” file shared among the cluster peers so that warm buckets would quickly roll over into the cold tier.
Expected outcome	We expected the eseries index on each peer indexer’s cold volume to fill with multiple buckets following the “db_<newest_time>_<oldest_time>_<localid>_<guid>” naming convention over time. We expected the eseries index on each indexer’s cold volume to fill with multiple replicated buckets following the “rb_<newest_time>_<oldest_time>_<localid>_<guid>” naming convention over time.
Test results	Passed
Comments	

7.5 Ability to Search

Table 12) Ability to search.

Test Case	Details
Test number	Ability to Search-1
Date	Multiple architectures: 8/17/17, 8/18/17, 8/24/17
Test prerequisites	We completed the Data Forwarding and Indexing test case and observed that E-Series array data was being forwarded and indexed correctly. We installed the NetApp Performance App for Splunk Enterprise on the search head.
Expected outcome	We expected configuration, performance, and event data collected from the E-Series storage array and indexed on the indexer cluster to populate in the NetApp Performance App for Splunk Enterprise. We expected to be able to search the cluster’s eseries index for all indexed events using a basic “index = eseries” query in the built-in Splunk Search and Reporting app.
Test results	Passed
Comments	

8 Conclusion

The NetApp E-Series 2800 provides a number of significant advantages over internal direct-attached storage (DAS) for Splunk deployments. These advantages include exceptional storage management capabilities, dramatically improved reliability, high availability, and limited performance degradation

because of failure conditions such as disk failures. By decoupling storage from compute, you gain the ability to scale capacity and compute separately, saving the cost of overprovisioning one or the other.

The advantages also include excellent performance handling ingest of machine log data and excellent search capabilities at very low latency with an E2800 configuration of all-flash SSDs for hot and warm Splunk buckets or a hybrid configuration employing HDDs, which utilizes SSDs as a read cache. The E-Series E2860 provides excellent performance and reliability for the Splunk cold data bucket tiers as well and can make use of an SSD read cache to improve query performance as needed.

Organizations that use Splunk often use traditional server-based storage with inefficient, hard-to-scale internal DAS. The NetApp reference design employs the managed DAS model, with higher scalability and performance. The reduction of the Splunk cluster replication factor available when deploying E-Series storage reduces the amount of indexed data stored. This reduction prevents unnecessary purchases of compute nodes for storage-intensive workloads for Splunk environments that need to grow to meet organizational requirements.

The NetApp Verified Architecture for Splunk is optimized for node storage balance, reliability, performance, storage capacity, and density. From an administrative standpoint, E-Series offers simplified storage management with a browser-based UI. This solution enables new volumes and Dynamic Disk Pools to be created easily and provisioned immediately for use by Splunk cluster servers. In addition, existing volumes and Dynamic Disk Pools can all be increased in size dynamically to provide additional capacity and/or performance as required for the Splunk indexer cluster environment.

This NVA describes using an E2824 with 12 SSDs for both the hot/warm tier and the cold tier as an example of what can be done. A different option would be to use an E2860 with SSDs for the hot/warm tier and large capacity NL-SAS drives for the cold tier.

Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and/or websites:

NetApp Documentation

- E-Series SANtricity Management Software Documentation
  https://mysupport.netapp.com/documentation/productlibrary/index.html?productID=61197
- Installing and Configuring for Linux Power Guide for Advanced Users
  https://library.netapp.com/ecm/ecm_download_file/ECMLP2439710
- NetApp SANtricity Web Services Proxy
  https://mysupport.netapp.com/NOW/cgi-bin/software?product=E-Series+SANtricity+Web+Services+%28REST+API%29&platform=WebServices
- Performance Characteristics of the Converged Infrastructure Solution with NetApp E-Series and Splunk – This is a NetApp Technical Report that will be published in the near future.

Splunk Documentation

- Splunk>docs
  http://docs.splunk.com/Documentation
- Splunk Enterprise Installation Manual
  http://docs.splunk.com/Documentation/Splunk/latest/Installation/Whatsinthismanual
- NetApp SANtricity Performance App for Splunk Enterprise
  https://splunkbase.splunk.com/app/1932/
- Technology Add-On for NetApp SANtricity
  https://splunkbase.splunk.com/app/1933/

Version History

Version	Date	Document Version History
Version 1.0	September 2017	Initial release.

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

Copyright Information

Trademark Information

NVA-1115-DESIGN-0917