Technical Report

NetApp HCI and Splunk Enterprise Solution with Arrow

Bobby Oommen, Stephen Carl, NetApp
Ronald Kuffer, Arrow

May 2019 | TR-4778

Abstract

This document presents performance and reliability data for NetApp® HCI in a Splunk Enterprise environment. It also presents test results for verifying healthy Splunk responses to restoring deleted data from snapshots and for recovering from a controller and disk failure.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Studies predict that, by the year 2025, the amount of data in existence will have grown by a factor of 10 times the amount present in 2017. This data comes in many forms, ranging from semi-structured to completely unstructured. Sources include the Internet of Things (IoT), business applications, social media, customer behavior, and machine sensors, to name just a few.

In this age of digital transformation, machine data is one of the key drivers fueling that growth. Machine data is generated by technology infrastructures, security systems, and business applications. It's one of the fastest growing types of data, and it’s also one of the most complex categories of big data. With that complexity comes high value in the form of information about customer behavior, business and personal transactions, sensor readings, machine behavior, security threats, and fraudulent activity. Splunk Enterprise software allows you to extract value from machine data in real time.

Enterprise organizations depend on big data to help run their businesses, to meet competitive demands in the marketplace, and to avoid costly infrastructure downtime. The ability to collect and analyze this data is key to transportation safety, machine reliability, fraud detection, and security. Enterprise data protection is an absolute necessity in meeting these demands.

NetApp in partnership with Arrow validated this solution to show how to create a Splunk setup on a NetApp HCI architecture. Arrow provides proven IT solutions that address business challenges and drive revenue, reduce costs, and reduce risks for enterprise organizations that are undergoing digital transformation. They also provide professional services, enterprise consulting, and training services, with Splunk as one of their core competencies.

2 NetApp HCI Private Cloud - Running Demanding Workloads in a Multitenant Environment

Moving to the private cloud is a journey. Most organizations start small and grow the environment over time. NetApp, in collaboration with VMware, has developed a private cloud solution that can start with a small footprint and grow compute and storage independently as workloads and performance expectations change.

We have the following goals and objectives for Splunk deployments in a private cloud:

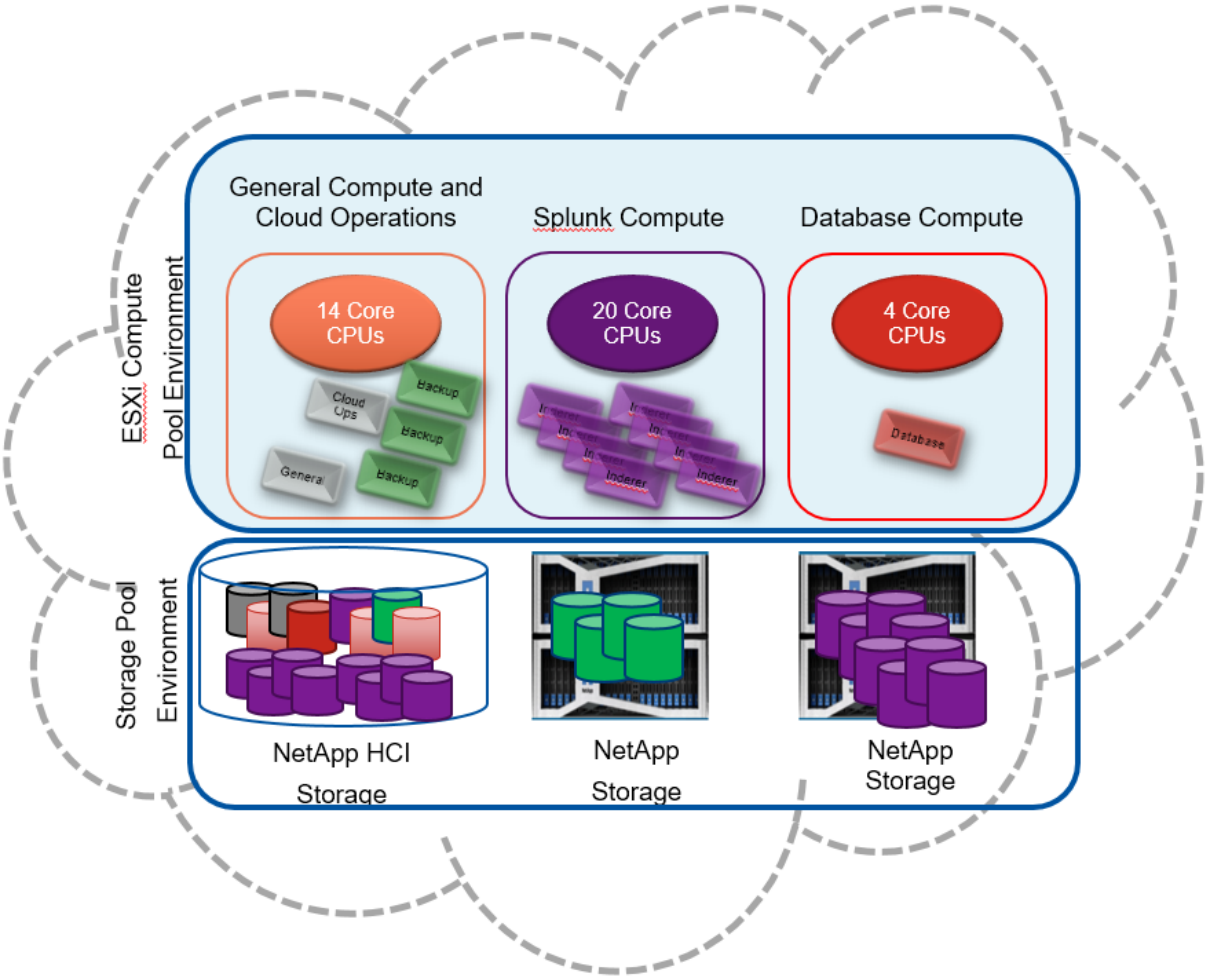
- **Self service.** Automate the deployment of additional services and support DevOps-type operations.
- **A flexible environment.** Grow, move resources to meet demand, and share resources across applications in a manner similar to hyper-scalers such as Amazon Web Services (AWS), Azure, and Google Cloud.
- **Meet application expectations in a multitenant and multiworkload environment.** Provide highly reliable performance, support, and capacity in a single managed environment.

NetApp has created two documents describing how to deploy VMware Private Cloud on NetApp HCI: a VMware Validated Design (VVD) and a NetApp Verified Architecture (NVA) for VMware Private Cloud on NetApp HCI. These documents provide guidelines, best practices, and deployment documentation for a VMware private cloud using the VMware vRealize Suite of products and NSX with NetApp HCI.

With this infrastructure, you can deploy applications and workloads on the desired CPUs and move applications as your needs evolve. The storage holding the data does not have to move and can be grown independent of the compute environment. The NetApp Data Fabric also enables you to connect your private cloud ecosystem on NetApp HCI to the hyper-scalers to deliver a hybrid multicloud experience.

For more information about the NetApp Data Fabric, see What Is Hybrid Multicloud Experience?

Figure 1) NetApp HCI Private Cloud with Splunk.



3 Solution Overview

Traditionally, Splunk was configured to use commodity servers with internal storage media (disks and solid-state drives [SSDs]) referred as direct-attached storage (DAS). Splunk best practices recommend configuring those internal disks by using RAID 10, also known as RAID 1+0. In a RAID 10 configuration, data is striped across pairs of mirrored disks to improve I/O performance and to protect against data loss in the event of disk failure. Because of RAID 10 mirroring, only half of the total raw disk capacity is available. Also, after all the disk bays in the servers are full, more servers must be added to accommodate data growth. This type of configuration leads to poor storage efficiency and a perpetual imbalance between compute and storage, because compute and storage cannot be scaled independently.

Storing Splunk data on NetApp HCI allows enterprises to create efficient configurations that meet their needs today and tomorrow. And with NetApp HCI both storage and compute resources can be added independently to meet the dynamic requirement. In addition to independent scaling of resources, NetApp HCI provides the data protection, data governance, storage efficiency, and copy management features needed to meet the requirements of enterprise organizations that use Splunk. Those features are discussed later in section 5.2.

4 Technology Overview

4.1 Splunk

Machine data is the fastest-growing type of big data. The format is unpredictable, coming from so many different sources, at such a high rate, and in such great volumes, that it’s often referred to as digital exhaust. It’s constantly being generated by servers, server infrastructures, applications, sensors, electronics, buildings, security systems, and all the elements that make up the Internet of Things (IoT). Machine data is tremendously valuable, in that it contains records of customer behavior, transactions, diagnostics from critical mechanical systems, message queues, change events — and the list goes on. It's difficult to unlock the value in this data due to its high volume and lack of structure. Splunk Enterprise provides the platform for collecting, indexing, and analyzing machine data from any source to deliver operational intelligence, which can be used to optimize IT, security, and business performance. NetApp HCI can fulfill these requirements and meet the SLA of an enterprise.

4.2 NetApp HCI

NetApp HCI is an enterprise-scale, hybrid-cloud-infrastructure solution that is ideally suited for customers who want to break free from the limitations of first-generation HCI systems.

With NetApp HCI, you can run multiple applications with guaranteed performance and confidently deploy resources across your entire data center. The architecture allows you to simplify management and independently scale both compute and storage resources. NetApp HCI is Data Fabric ready, so that you can access all your data across any public, private, or hybrid cloud. By moving to NetApp HCI, IT organizations can transform their data center by driving operational efficiency and reduce costs.

4.2.1 Performance Guarantee

One of the biggest challenges in any data center is delivering predictable performance, especially when multiple applications share the same infrastructure. One application might also interfere with another causing performance issues and forcing administrators to spend time troubleshooting. Mainstream applications, such as virtual desktop infrastructure (VDI) and database applications, have very different I/O patterns that tend to affect one another when deployed in a shared infrastructure. Storage nodes come with the Quality of Service (QoS) feature which enables finely grained control of every application, eliminating noisy neighbors, meeting unique performance needs, and satisfying performance SLAs. The NetApp HCI storage architecture also eliminates any performance variation resulting from data locality because the data is distributed across all the nodes in the cluster.

4.2.2 Enterprise Scale

NetApp HCI scales compute and storage resources independently, avoiding costly and inefficient overprovisioning and simplifying capacity and performance planning. Running on innovative NetApp SolidFire® technology and delivered on an architecture designed by NetApp, NetApp HCI is a true enterprise-scale, hybrid cloud infrastructure solution. NetApp HCI comes in 2RU by four-node building blocks (chassis) with compute and storage nodes available in small, medium, and large sizes.

4.2.3 Streamline Operation

A primary goal of all IT departments is to automate all routine tasks, eliminate the user error associated with manual operations, and focus on higher value assignments that drive business. NetApp HCI enables IT departments to become more agile and responsive by simplifying day 0 deployment and ongoing management from day one forward. The NetApp Deployment Engine (NDE) eliminates most manual steps needed to deploy infrastructure, although the vCenter plug-in makes management in the VMware environment simple and intuitive. Finally, a robust suite of APIs enables seamless integration into higher-level management, orchestration, backup, and disaster recovery tools.

4.2.4 Configuration

NetApp HCI is available with multiple configuration options—small, medium, large for both compute and storage—that are essentially a small blade that sits inside a chassis. Table 2 lists the configuration specifications.

From the configurations described in Table 2, the NetApp HCI solution supports 5TB to 44TB of effective capacity. For compute resources, NetApp HCI supports 16 to 36 cores and 256G to 768G of RAM with the flexibility of mixing and matching compute or storage nodes independently. For example, if your setup has two compute nodes and four storage nodes and you must add more compute resources, you can add a compute blade into the chassis without adding storage. This scalability and flexibility enables you to build an efficient and agile cloud in your data center.

Table 1) NetApp HCI configuration.

Storage Nodes				Compute Nodes			
	Small	Medium	Large		Small	Medium	Large
Rack unit	1 rack unit, half width			Rack unit	1 rack unit, half width		
SSD size	6 x 480GB	6 x 960GB	6 x 1.92TB	Cores	8 to 16	16 to 28	28 to 40
Effective capacity	5.5TB to 11TB	11TB to 22TB	22TB to 44TB	RAM	256BG	512GB	768GB to 1TB

5 NetApp HCI Use Cases

NetApp HCI can support a wide range of database application use cases. This section shows how to identify when application use cases are a good fit for the NetApp HCI solution.

5.1 Consolidation

The per-volume QoS controls for storage nodes help individual databases get required I/O throughput without being affected by other databases that run in parallel on the same NetApp HCI setup. With QoS and data reduction efficiencies, you can achieve higher database density with the shared storage infrastructure by having several database instances. By adding compute nodes, the environment can be used as a multipurpose multi-workload environment using shared storage without the risk of performance contention among applications.

Moreover, database or VMware administrators have full control of each storage volume on which the database resides and can perform all maintenance operations. They can set the QoS for each database copy through the vCenter SolidFire plug-in, use REST APIs to achieve full automation, and make storage management simpler and easier. Each index server in the Splunk architecture is configured with separate volumes on the storage nodes. In addition, the global data reduction feature provides efficient space usage regardless of the number of servers in the cluster.

5.2 Development and Testing

Storage snapshots provide a point-in-time view of the contents of an active file system or storage volume. You can use these snapshots for rapid recovery of corrupted datasets and to create space-efficient copies of datasets for development and testing use cases. The cloning process can be coupled with SolidFire QoS control so that database clones can coexist with the production copies without any performance effects on the upstream applications. The CopyVolume feature of storage nodes allows you to refresh an existing cloned copy of a database without performing any file system remount operations. In this use case, you frequently refresh a copy of the database by only taking changes from the production copy.

6 Test Configuration Details

This section describes our tested configuration, including the network infrastructure, Splunk server functionality, storage provisioning details, and Linux host-side storage configuration. The configuration is based on the NetApp HCI and Splunk Enterprise Solution with Arrow.

6.1 NetApp HCI

A configuration consisting of four storage nodes (H300s) and two compute nodes (H300e) was used for this setup. The storage nodes form a storage cluster with a guaranteed I/O performance of 200,000 4K IOPS and a raw capacity of 9.6Tib. Each H300e contains 2 x Intel E5-2620 V4 with 8 cores CPU's rated at 2.1GHz and 384GB of memory. iSCSI protocol was used to present storage resources as a VMware Virtual Machine File System (VMFS) datastore or as a VVol for the connected vSphere instance.

Figure 2) NetApp HCI chassis.

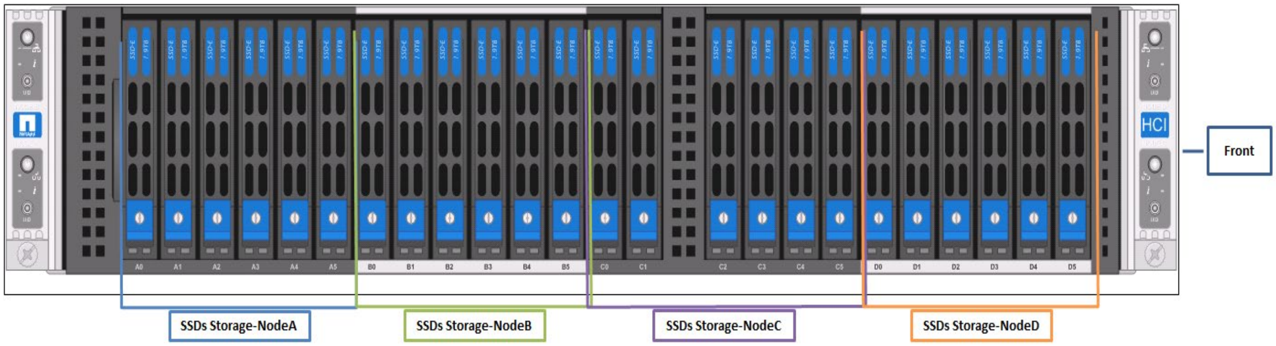
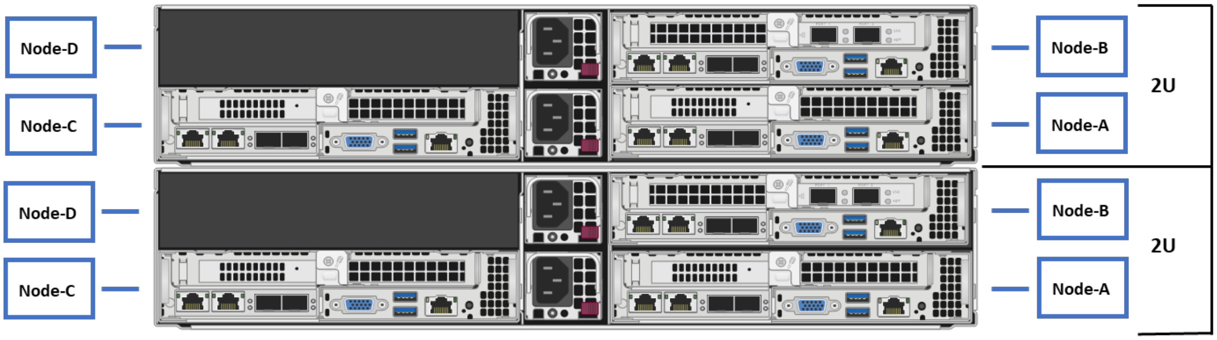


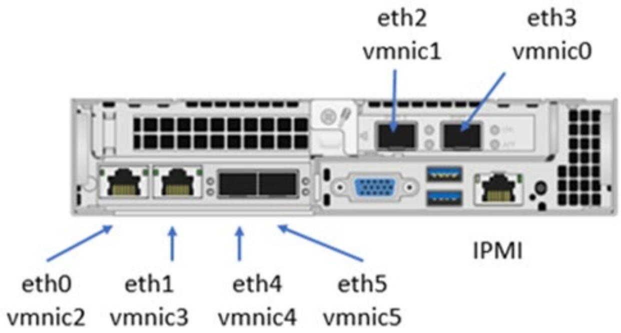
Figure 3) NetApp HCI rear view.

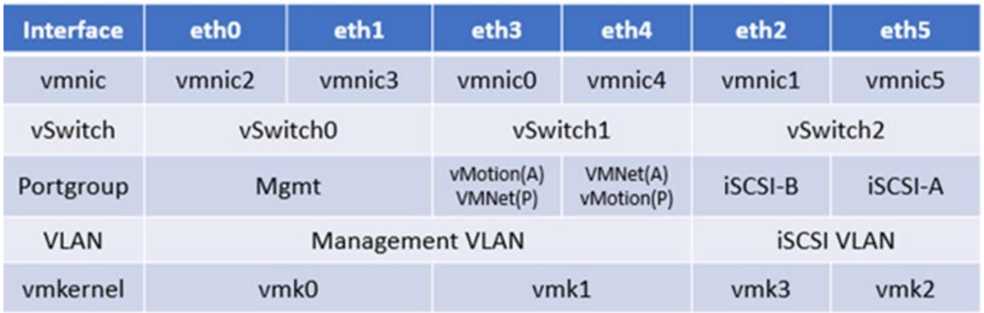


6.2 Network

A Juniper EX4500 10GbE switch with two uplink modules was used for redundancy. The two 10/25GbE ports for the NetApp HCI nodes were not configured with the Link Aggregation Control Protocol as per NetApp best practice. Instead, they were implemented in an active-passive mode. For the ESXi hosts, two 10/25GbE Ethernet ports were configured with individual VMkernels and port binding to the iSCSI virtual adapter for the switch. The following three virtual LANs (VLANs) were defined (management, vMotion, and iSCSI). Therefore, for each datastore, there were two active iSCSI sessions available. The default settings were used for vSphere.

Figure 4) NetApp HCI compute node ports.



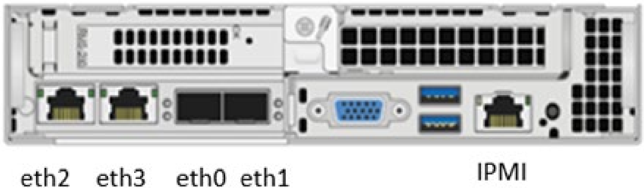


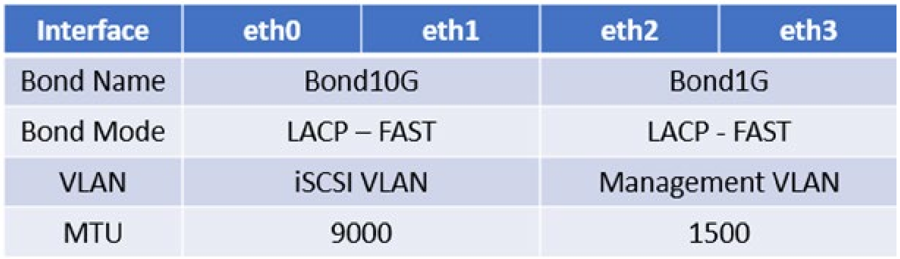
Interface	eth0	eth1	eth3	eth4	eth2	eth5
vmnic	vmnic2	vmnic3	vmnic0	vmnic4	vmnic1	vmnic5
vSwitch	vSwitch0		vSwitch1		vSwitch2	

Portgroup	Mgmt		vMotion(A)
			VMNet(P)
				VMNet(A)
				vMotion(P)
					iSCSI-B	iSCSI-A

VLAN	Management VLAN				iSCSI VLAN	
vmkernel	vmk0		vmk1		vmk3	vmk2

Figure 5) NetApp HCI storage node ports.

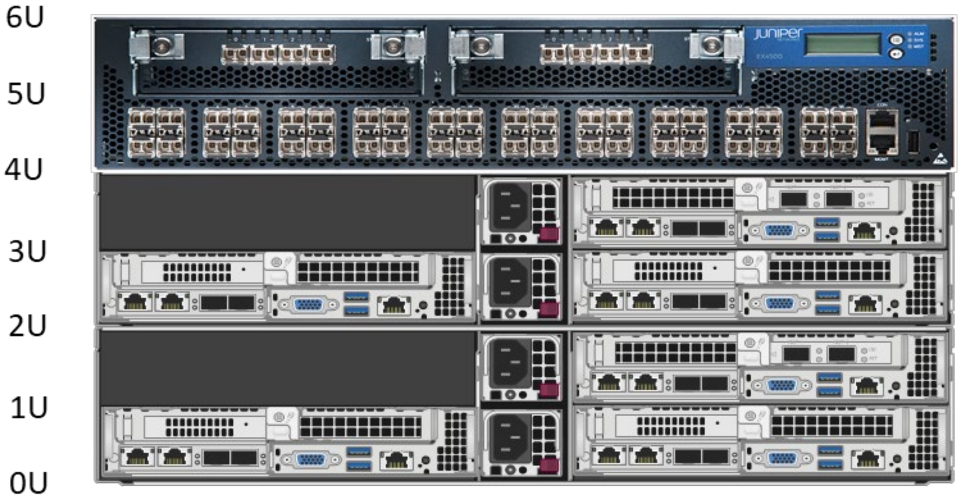




Interface	eth0	eth1	eth2	eth3
Bond Name	Bond10G		Bond1G	
Bond Mode	LACP – FAST		LACP - FAST	
VLAN	iSCSI VLAN		Management VLAN	
MTU	9000		1500	

Test Configuration

Figure 6) Tested solution configuration.



Components

- 4-6 U Juniper EX4500
  10Gbit switch SFP+
- 0-4 U NetApp HCI
  4x H300s, 2x H300e

Network Information: 6-8 U Juniper EX4500

A Juniper Networks EX4500 Ethernet switch provide high-density 10-gigabit ports for aggregation layer and data center top-of-rack deployments. Passive Twinax direct-attached copper cabling was used (10G SFP+ DAC, 2m) for 10Gbit connections. For 1Gbit connections, SFP+ to RJ45 adapters (10GBASE-T copper SFP+) provide adequate port adaptions.

2-6 U NetApp HCI

The smallest valid configuration for NetApp H-Series systems consists of two 2U chassis bearing four small storage nodes (H300s) and two small compute nodes (H300e). The storage nodes were an all-flash storage cluster with a raw capacity of 9.6TiB and 200k IOPS performance. LUNs or VVols were presented to hosts over iSCSI with storage QoS to eliminate any noisy neighbor effects. Both compute nodes had two Intel E5-2620v4 CPUs, each, providing a system total of 64 v-cores and 768GB RAM as resources to the VMware vSphere cluster. Two unoccupied bays could be upgraded with one or two more nodes, either storage or compute.

NetApp HCI Hardware

Figure 7) Tested NetApp HCI configuration.

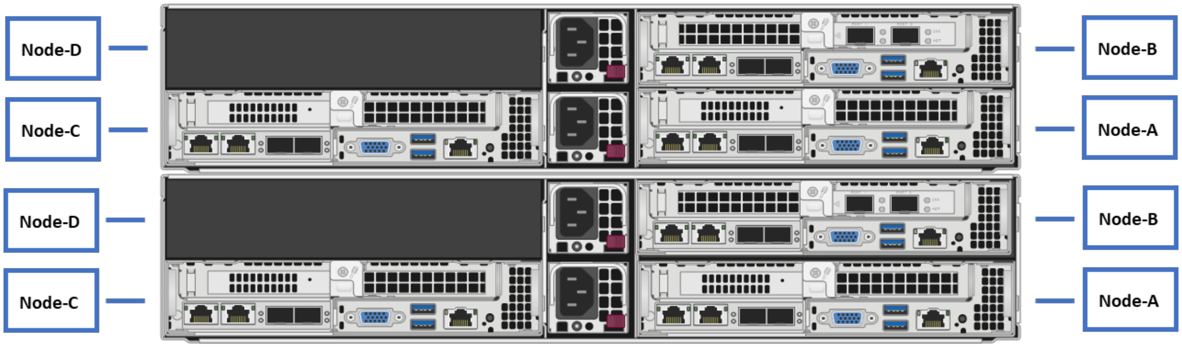


Table 2) NetApp HCI hardware compute nodes.

Node Qty	Type	Small	Medium	Large
2	RU	1RU, half-width	1RU, half-width	1RU, half-width
2	Cores for VMs	16	24	36
2	CPU	E5-2620 v4 – 8C @ 2.1GHz	E5-2650 v4 – 12C @ 2.2GHz	E5-2695 v4 – 18C @ 2.1GHz
2	Memory	384GB	512GB	768GB
2	Boot device	(1) 240GB MLC	(1) 240GB MLC	(1) 240GB MLC
2	Base networking	(4) 25/10GbE SFP28 /SFP+ (2) 1GbE RJ45	(4) 25/10GbE SFP28 /SFP+ (2) 1GbE RJ45	(4) 25/10GbE SFP28 /SFP+ (2) 1GbE RJ45

Other Hardware

Network

- **Switch.** Juniper EX4500 10Gb SFP+

Additional Storage

- **NetApp E-series - E2724.** Not used in testing, part of the Arrow, Splunk, and NetApp solution tiered storage for warm and cold data. For warm and cold data.

Software

NetApp HCI 1.1

- Bootstrap OS 1.1.0.8
- NetApp Element® software 10.1.0.83
- VMware vCenter Plug-in 3.0.1
- NetApp Monitoring Agent 1.1.0

VMware vSphere 6.x

- VMware vCenter 6.0 U3a build 5183549
- VMware vCenter 6.5 U1 build 5973321

Operating System

- Cent OS 7

CentOS 7 Splunk Data Ingestion VMs

- Splunk clients (VM)
- Syslog server (VM)

6.3 Solution Architecture

The Splunk solution discussed in this document was built and tested by using the following components, as illustrated in Figure 6.

Splunk

- Splunk Enterprise Server 7.0.2
- Distributed architecture, REST API only
- Data collection: ingest data directly from data source or through forwarder
- Index data and search data, distributed search
- Indexer: 128kB chunks: journal.gz
- Time serialized data index TSIDX (uncompressed binary)
- Buckets: journal.gz + TSIDX, auto: 750MB, high volume: 10GB per bucket, customizable
- Rule of thumb: 15% of incoming data stored raw, 35% stored in TSIDX (sums up to 50%)
- Index writes uses max four cores per index server

6.4 Recommended Configuration on NetApp HCI: Small Two Compute Nodes and Four Storage Nodes

Table 3 presents the recommended configuration for the solution with two compute nodes in the small configuration.

Table 3) Recommended Splunk configuration first compute node.

Splunk Component	Task	Qty	Cores	Memory
Indexer	Manages the user data	1	16 vCores	32Gb RAM
Search head	User front end, searches data in indexers	1	16 vCores	32Gb RAM

Table 4 presents the recommended Splunk configuration for the second compute node.

Table 4) Recommended Splunk configuration second compute node.

Splunk Component	Task	Qty	Cores	Memory
Indexer	Manages the user data	1	16 vCores	32Gb RAM

Splunk Component	Task	Qty	Cores	Memory
Cluster Master	Manages the Splunk installation and indexers	1	8 vCores	16Gb RAM
vCenter		1	2 to 8 vCores	8Gb to 16Gb RAM

This Splunk configuration on NetApp HCI supports a 750Gb daily ingest rate for 12 users.

Splunk Deployment Optimizations

Cent OS 7 Optimization

Set the following ulimits in the operating system to these recommended values:

```
ulimit -c unlimited
ulimit -d unlimited
ulimit -n 65536
ulimit -u 258048
```

To disable the following values, enter these commands:

```
echo never > /sys/kernel/mm/transparent_hugepage/enable
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

Splunk Optimization

In the server.conf file on an indexer in a cluster, add the following entry:

```
[general]
parallelIngestionPipelines = 2
```

6.5 Overview of Splunk Components

A Splunk installation always consists of the following components.

Table 5) Splunk hardware components for scale-out configuration.

Splunk Component	Task
Forwarder	Client ends data to the indexer
Indexer	Manages the user data
Search head	User front end, searches data in indexers
Cluster master	Manages the Splunk installation and indexers

These components can be on a server (combined), or they can be distributed in larger installations (distributed or scale-out). The following test scenarios depict useful configurations for production environments. Using the scale-out variant, the Splunk installation can be grown in a variety of ways.

Indexer

The Indexer manages the payload and is thus the workhorse of a Splunk installation. This component indexes the data from the various data sources (for example, forwarders) and respond to the search queries from the search heads. In the default scenario, each indexer uses up to four CPU cores for data adoption and indexing. The remaining cores are available for search queries for optimal performance. NetApp recommends a user-to-CPU ratio of 1:1. However, the server should still have cores available. The performance of a virtual core (hyperthreading or HT) is not equivalent to a physical core. The load is dependent on the search queries and can lead to careless searches in which the entire compressed data stock must be searched. An index can also be replicated to other indexers.

Search Head

The search head forms the front end to the user and continues to distribute search queries to the indexers. In the configurations listed in this document, there is enough reserve available that the environment can scale and grow to meet search query demand.

Cluster Master

The cluster master manages the entire solution. Responsibilities include the administration of apps and indexers and the replication of indexes.

7 Test Information

The testing setup is designed in accordance with Splunk reference architecture information. A basic Splunk cluster consisting of two indexers, one search head, and cluster management can be installed on a minimum NetApp HCI solution. The minimum solution consists of two H300 compute nodes and four H300 storage nodes. This is the basic configuration for testing.

All tests were performed on a single instance VM according to the Splunk 7.0.2 Capacity Reference hardware guide and Splunk 7.0.2.Capacity Summary of Performance Recommendations.

The component server uses 12 vCores @ 2GHz+, 12GB of RAM, and the CentOS 7 (64 bit) as shown in Table 6.

7.1 Base OS Installation

Table 6) Base OS test installation.

Host/VM	Server Role	CPU Cores	Memory	Daily ingest rate	Users
CentOS 7 VMware tools Bonnie++	Search head / indexer / cluster master	12	12GB	Up to 500GB	Up to 4

NetApp recommends the combined/single instance installation for all customers with a daily ingest rate up to 500GB as is indicated in Table 6. Storage sizing is based on how high the daily data rate is and on the duration and retention of this data in the hot/warm bucket.

The data for Splunk single-instance data sources is as follows:

- Firewall > syslog
- vSphere cluster > syslog
- Eventgen > direct files

The single instance in this test required one indexer. When tuning the setup, we only optimized the Linux OS for core file size, data segment size, max open files, max user processes, and database HugePage handling. We also implemented a second ingestion pipeline in Splunk. Only a single VM instance was needed to verify the performance of Splunk on the hardware. All cluster configuration information was based on reference material and the single instance VM tested.

The use of forwarders was not needed in this setup, because our lab environment could not generate data fast enough. We pre-generated data like syslog files and artificial data from Eventgen and stored them on separate iSCSI LUNs on the NetApp HCI system. See the Appendix section 0 for the method we used to create the test data.

For more general information about Eventgen data, see the Splunkbase page for Eventgen.

Our primary goal for these test sessions was determining the pure indexing performance using Bonnie++ and demonstrating that shared storage is a highly capable and fast solution for Splunk. Therefore, only hot data was used for testing on this single instance.

7.2 Splunk Components Common Purpose

The components in Figure 3 provide the following functions:

- The indexer servers receive and index incoming data, send or receive replicated data in the cluster, and search across indexed data for search requests from the search head.
- The forwarders consume data from external sources and forward it to indexer servers.
- The search head manages searches across the cluster of indexer servers. It distributes the search queries to the indexer servers and consolidates the results.

  Note: All searches are run from the search head. Each cluster must have at least one search head.
- The cluster master (controller monitor) manages the Splunk cluster. It coordinates replicating activities of the indexer servers and communicates with the search head for information about where to locate data for searches. It also remediates activities if an indexer server goes offline.

  Note: Each cluster has only one controller monitor.

The machine log data from the Splunk forwarders sent to the indexer peer nodes uses the recommended data replication factor of three, which makes three copies of data available. The ingested data is compressed and indexed as raw data files and metadata, which are then distributed among the indexer peer nodes for redundancy. Reducing the replication factor to two copies can potentially boost performance and reduce the storage capacity needed for indexes.

8 Test Results Summary

We performed testing with the ingest of several data sources. To get data fast enough, we performed the performance testing with prerecorded data from syslog and Eventgen sources. The testing environment did not allow extensive searches. Therefore, we did not pursue search measurements other than monitoring the appliance. By using a single instance, we did not focus the effect of Quality of Service (QoS) on multiple workloads. Tests were all on small, subterabyte volumes.

We used the following Splunk data ingest scenarios and rates for our testing:

- 10,995 Kbps; 950GB/day set up and go
  - Nonoptimized configuration
  - single/combined instance (12 vCores, 12GB RAM)
- 13,836 KBps; 1,195 GB/day enhanced Linux performance
- Performance kernel settings of Linux guests
  - Distribution of VMs, raw data, and indexed data to three independent datastores
- 16,995 KB/s; 1,467 GB/day increased cores, additional second pipeline
  - Increasing use of cores from 4 to 6 in indexer configuration
  - Still same single/combined instance (12 vCores, 12 GB RAM)

Splunk Deployment Optimizations

To optimize performance, the following commands were run on the Linux hosts:

```
ulimit -c unlimited
ulimit -d unlimited
ulimit -n 65536
ulimit -u 258048
```

Disable these values by entering the following commands:

```
echo never > /sys/kernel/mm/transparent_hugepage/enable
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

Splunk-Specific Optimizations

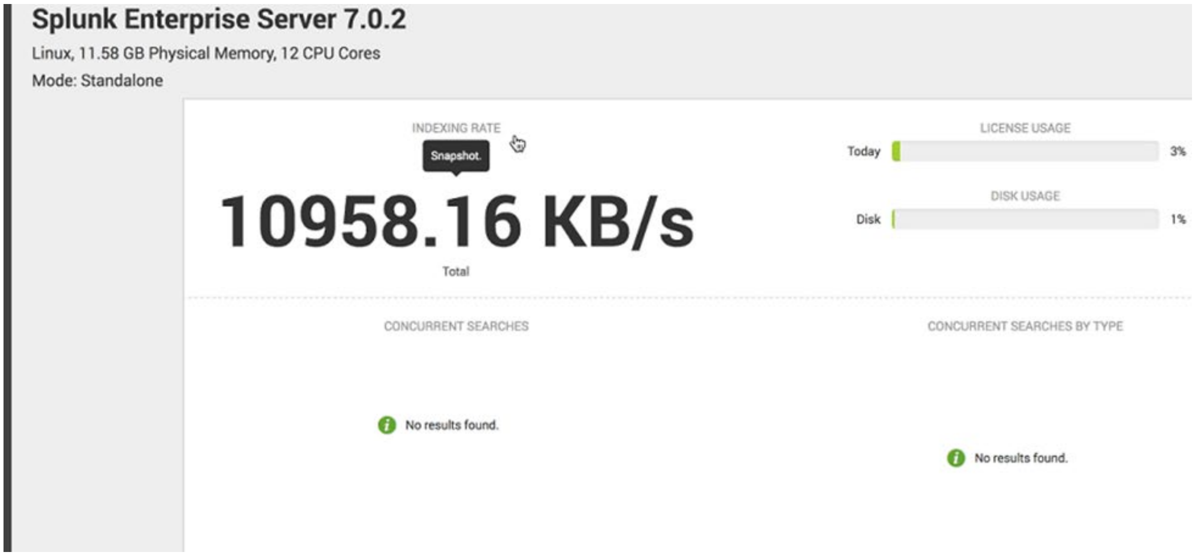
You should perform Splunk-specific optimization to increase the ingestion rate by using two pipelines with CPU vCores.

In the `server.conf` file on an indexer in a cluster, add the following entry:

```
[general]
parallelIngestionPipelines = 2
```

In the first test run, the default settings were used by the operating system and the Splunk installation. We achieved an index rate of approximately 10,000Kbps under a full load of the four CPUs that were available for the index process (Figure 8).

Figure 8) Splunk index rate with default settings.



Splunk Enterprise Server 7.0.2

Linux, 11.58 GB Physical Memory, 12 CPU Cores

Mode: Standalone

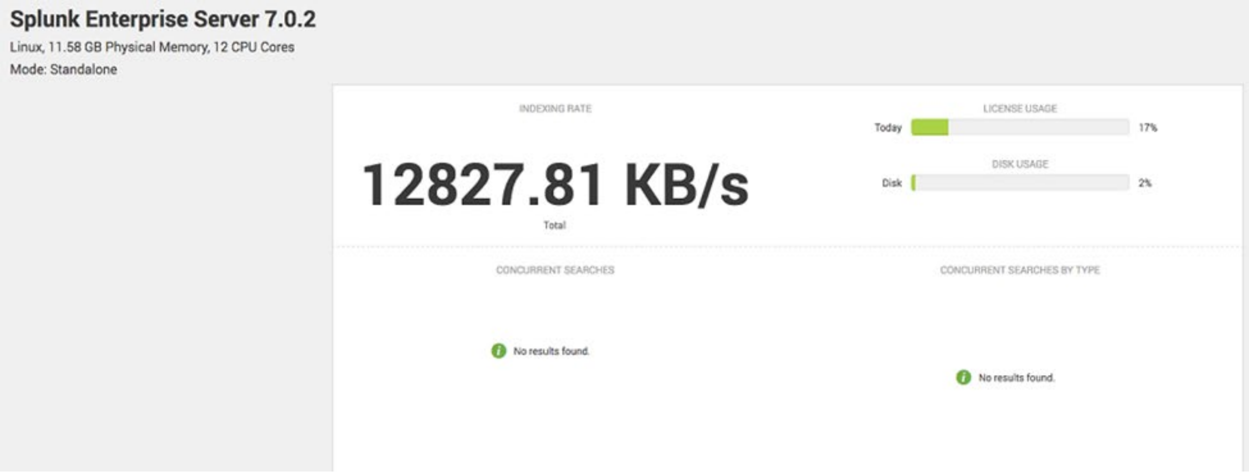
No results found.

No results found.

We ran the second test run with optimized operating system settings. We changed the CentOS kernel parameters according to Splunk architecture recommendations. For more information, see the Splunk Deployment .

With the Cent OS 7 optimizations, we achieved 12827KBps (1TiB/day; Figure 9).

Figure 9) Tested solution for basic system configuration with Cent OS optimizations.



Splunk Enterprise Server 7.0.2

Linux, 11.58 GB Physical Memory, 12 CPU Cores

Mode: Standalone

INDEXING RATE

12827.81 KB/s

Total

LICENSE USAGE

Today 17%

DISK USAGE

Disk 2%

CONCURRENT SEARCHES

No results found.

CONCURRENT SEARCHES BY TYPE

No results found.

Test 3 - Optimized Basic System Plus Adjusted Splunk Parameters

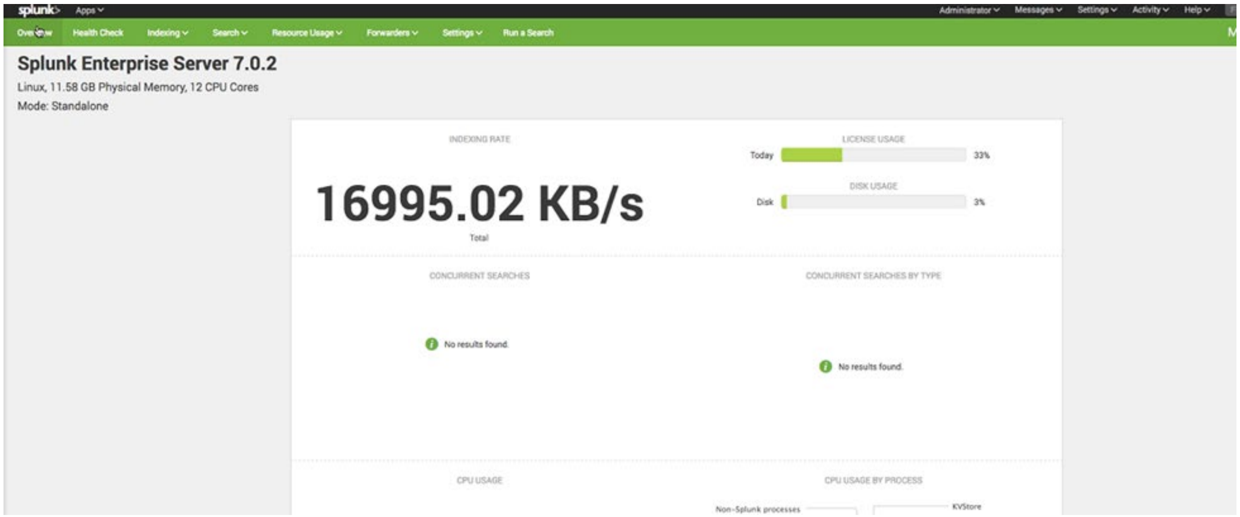
In the final test, the Splunk options were adjusted as follows:

- vCPUs were increased to 6.
- A second pipeline was added with parallelIngestionPipelines = 2.

This allows the indexer to use up to eight CPUs for indexing and achieve an index of 15,696 Kbps (1.4TiB/day).

In Figure 10, the optimizations include Splunk parameter changes that increase the pipeline bandwidth, which adds additional CPUs in the indexer to ingest data from the forwarders. This configuration enables the indexer to use up to eight CPUs for indexing and achieve an index rate of 16995 Kbps (approximately 1.4TB/day).

Figure 10) Splunk index rate with optimizations and increased pipelines.



Splunk Enterprise Server 7.0.2

Linux, 11.58 GB Physical Memory, 12 CPU Cores

Mode: Standalone

INDEXING RATE

16995.02 KB/s

Total

LICENSE USAGE

Today 33%

DISK USAGE

Disk 3%

CONCURRENT SEARCHES

No results found.

CONCURRENT SEARCHES BY TYPE

No results found.

CPU USAGE

CPU USAGE BY PROCESS

Non-Splunk processes

KVStore

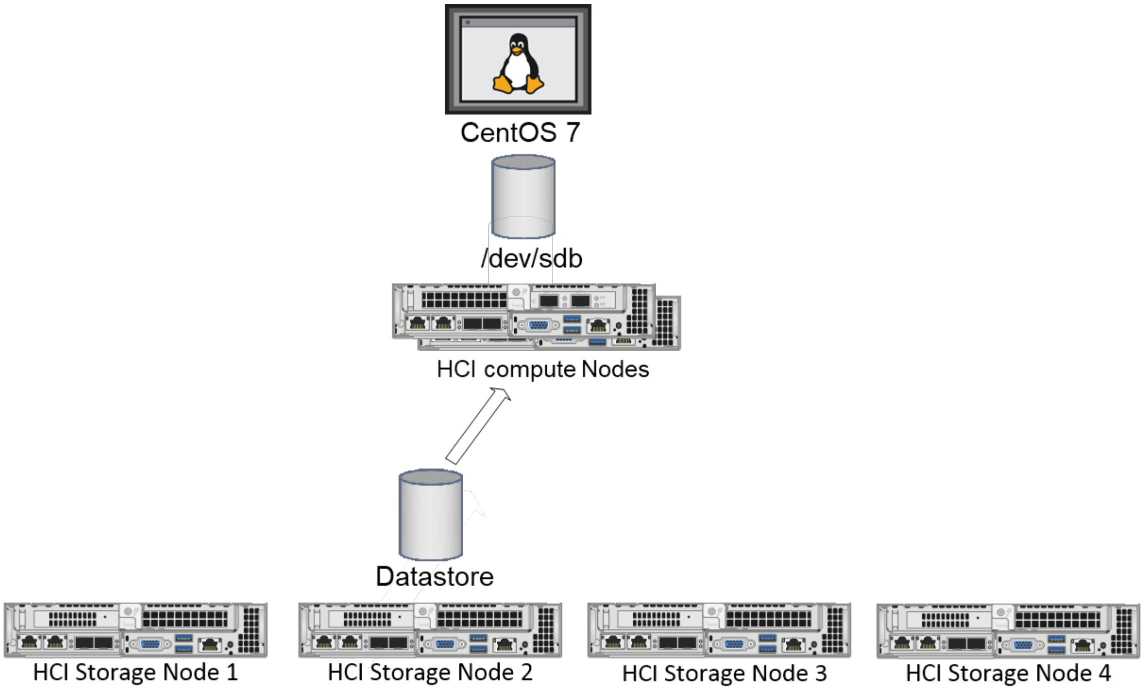
8.1 Bonnie ++ Storage Tests

Two scenarios have been tested using Bonnie ++. The first is the minimum configuration with a Virtual Machine Disk (VMDK) in a separate VMFS datastore. The second is a performance configuration in which four VMDKs configured in VVols were created in the base VM using LVM with a 64k segment size. In both tests, eight vCores and 8GB of RAM were available to the VM.

Bonnie++ Minimum Configuration

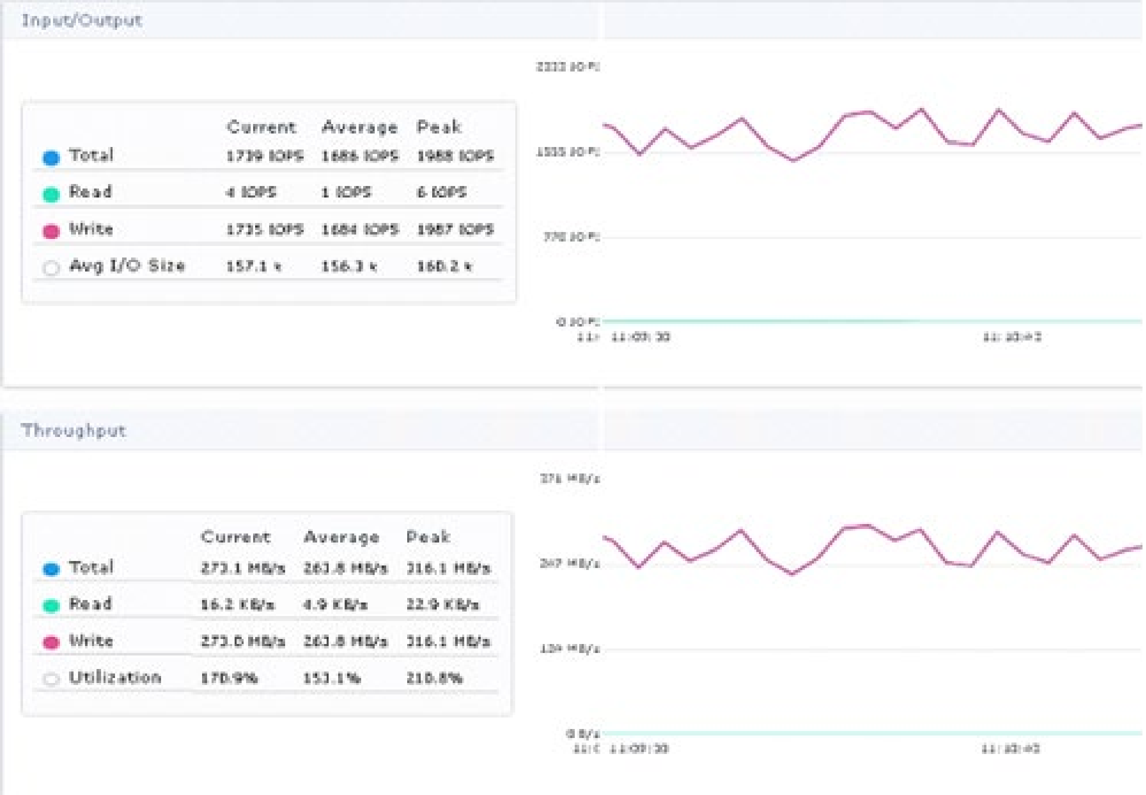
In the minimum configuration, a VMDK file was placed in a separate VMFS datastore and mounted without the use of LVM (Figure 2). On each datastore, we defined QoS settings of a minimum of 10,000 IOPS, a maximum of 30,000 IOPS, and a burst of 35,000 IOPS.

Figure 11) Minimum Bonnie ++ configuration.



In this configuration, we measured a sequential write with an average rate of 256 MBps (Figure12).

Figure12) IOPS and throughput for minimum Bonnie++ configuration.



Input/Output

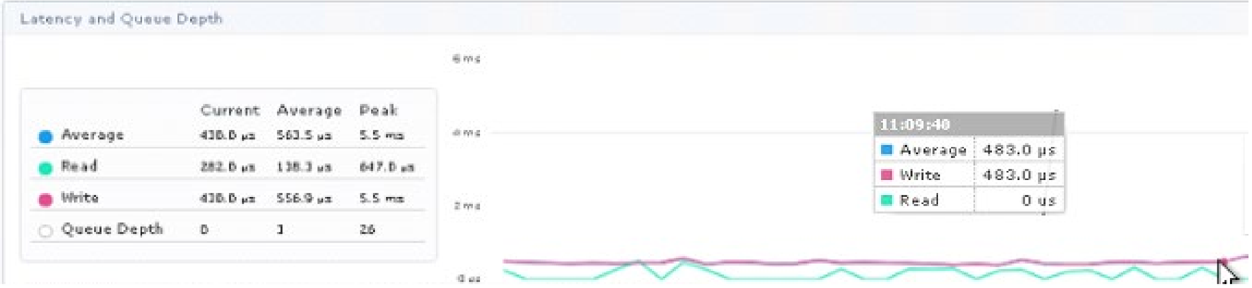
	Current	Average	Peak
● Total	1739 IOPS	1686 IOPS	1988 IOPS
● Read	4 IOPS	1 IOPS	6 IOPS
● Write	1735 IOPS	1684 IOPS	1987 IOPS
○ Avg I/O Size	157.1 k	156.3 k	160.2 k

Throughput

	Current	Average	Peak
● Total	273.1 MB/s	263.8 MB/s	316.1 MB/s
● Read	16.2 KB/s	4.9 KB/s	22.9 KB/s
● Write	273.0 MB/s	263.8 MB/s	316.1 MB/s
○ Utilization	170.9%	153.1%	210.8%

We observed less than sub-millisecond latency during the Bonnie++ tests (Figure13).

Figure13) Latency for minimum configuration.



Latency and Queue Depth

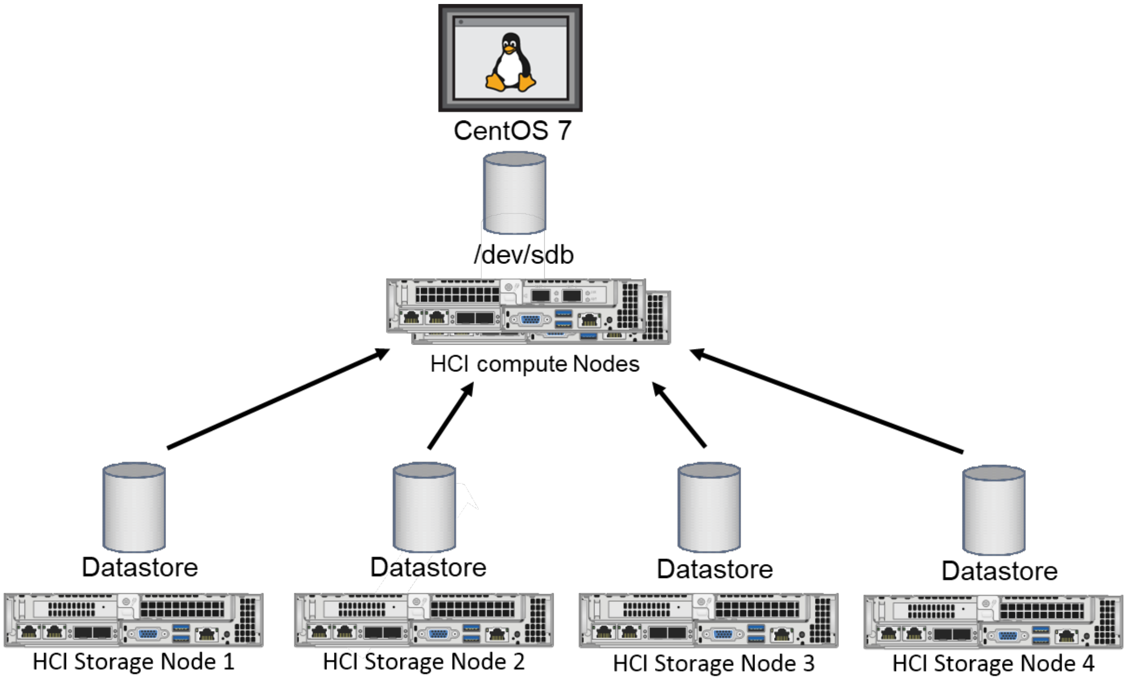
	Current	Average	Peak
● Average	430.0 µs	563.5 µs	5.5 ms
● Read	282.0 µs	138.3 µs	647.0 µs
● Write	430.0 µs	556.9 µs	5.5 ms
○ Queue Depth	0	3	26

11:09:40	
■ Average	483.0 µs
■ Write	483.0 µs
■ Read	0 us

Bonnie++ Performance Configuration

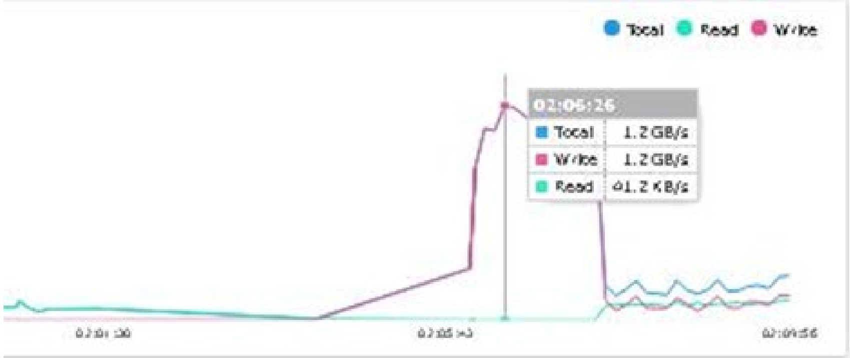
To show what performance is expected from a four-storage-node configuration, four virtual disks were mounted in VVols to the CentOS, and LVM created a virtual volume with a strip size of 64k. On each VVol, we defined QoS settings of a minimum of 10,000 IOPS, a maximum of 30,000 IOPS, and a burst of 35,000 IOPS.

Figure 14) Bonnie ++ performance configuration.



A quick performance verification with the Bonnie++ performance configuration delivers more than 1 GBps of writes at a 64k block size using eight writers (threads). This performance is more than enough to deliver ingest rates of 500 to 2,000GB/day.

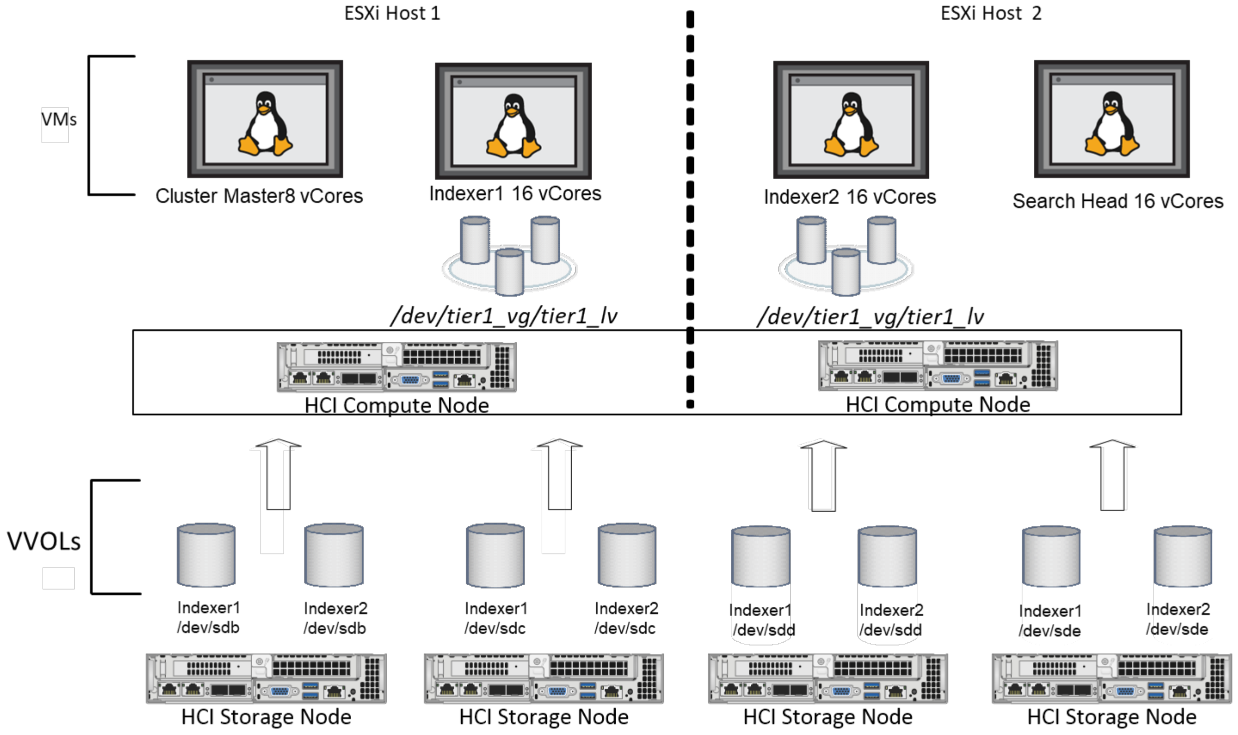
Figure 15) Bonnie++ performance configuration writes.



9 Scale-Out Installation

A scaled-out installation of the solution is recommended for anyone who has a higher index rate (a per indexer maximum of 500GBps), higher reliability from replication, or simply a requirement for more concurrent users. The cluster master is designed for additional tasks. Proper sizing of the search head and the indexer are required to scaling the system correctly. When using replication, the data rate of indexer performance should be approximately halved in case there is a failure. As a result, an indexer should still have enough resources to index the data.

Figure 16) Scale-out installation.



As shown in Figure 16, there are 32 vCores per H300e compute node. Splunk recommends sharing no cores so that you only have to decide if running an ESXi host in maintenance or failure mode conditions provides sufficient performance. Another possible consideration is adding a third H300e to satisfy the VMware Best practices of n + 1. This includes handling the requirement that the vCenter or the NetApp HCI management node (mnode) still needs a total of five vCores.

Table 7) Scale-out configuration.

Server Role	CPU Cores	Memory	Daily Ingest Rate	Users
Search Head	16	32GB	Up to 700GB	Up to 8
Indexer 1	16	32GB	Up to 700GB	Up to 8
Indexer 2	16	32GB	Up to 700GB	Up to 8
Cluster master	8	32Gb	Up to 700GB	Up to 8

The testing was performed with VVols configured for the host volumes.

10 NetApp HCI, Splunk, and Arrow Solution

The NetApp HCI system provides both storage and compute resources, combining them to build a VMware vSphere environment backed by the capabilities of NetApp Element software. You can quickly deploy and configure your fully racked and powered NetApp HCI system using the NetApp Deployment Engine web interface in a web browser.

This document describes the evaluation setup and test results of Arrow ECS Germany. These results were used to provide rudimentary sizing and performance information for setting up Splunk Enterprise virtualized on a hybrid cloud infrastructure solution.

10.1 Benefits

NetApp HCI QoS guarantees performance and predictability for Splunk workloads.

Figure 17) Benefits of NetApp HCI QoS.

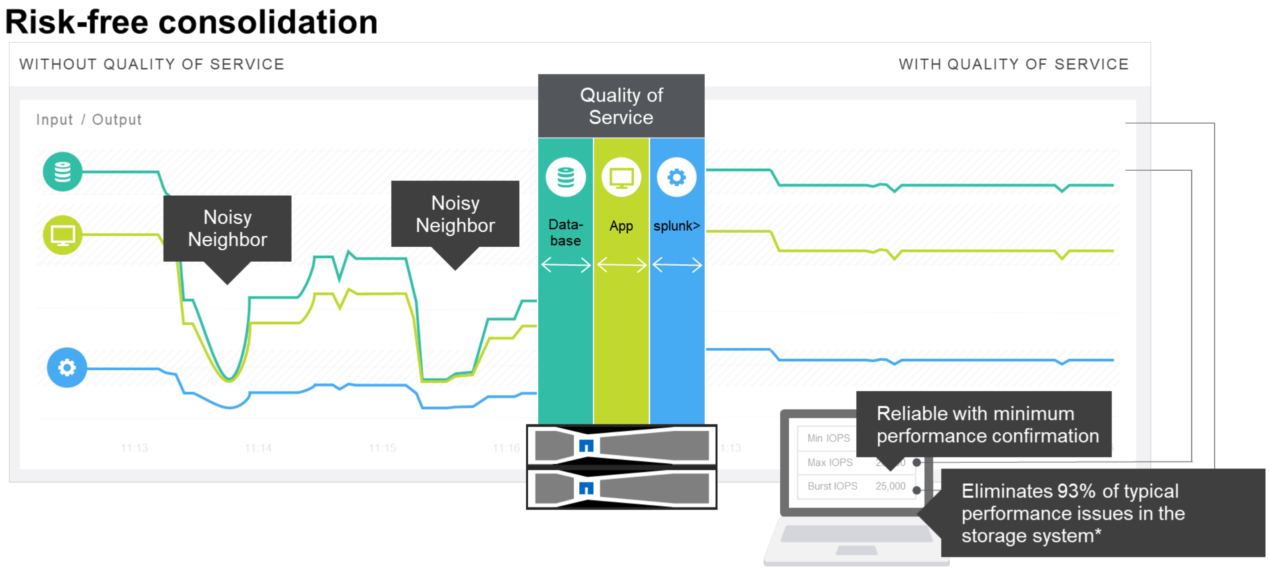
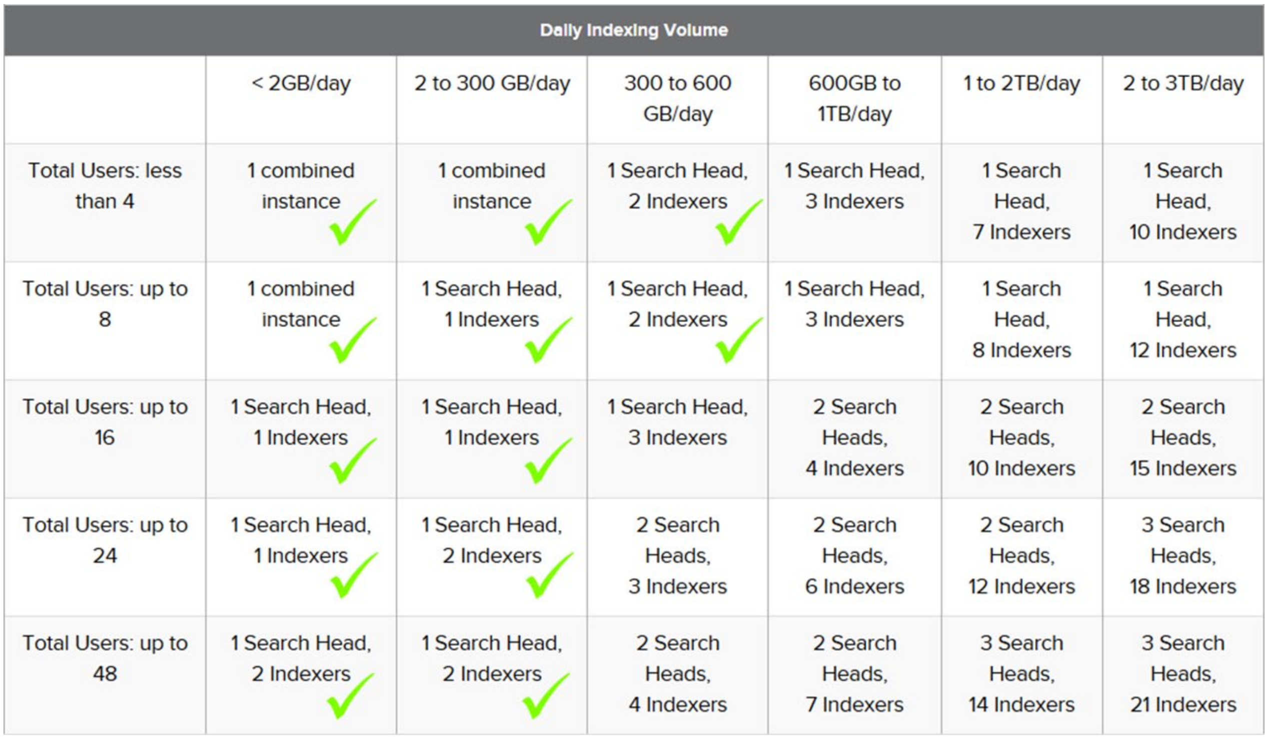


Table 8 provides examples for NetApp HCI Splunk sizing. The available configurations in the NetApp HCI/Arrow/Splunk solution are related to the sizing requirements for the amount of data ingested per day.

Table 8) Sizing requirements relative to daily ingested data.



	Daily Indexing Volume					
	< 2GB/day	2 to 300 GB/day	300 to 600 GB/day	600GB to 1TB/day	1 to 2TB/day	2 to 3TB/day
Total Users: less than 4	1 combined instance ✓	1 combined instance ✓	1 Search Head, 2 Indexers ✓	1 Search Head, 3 Indexers	1 Search Head, 7 Indexers	1 Search Head, 10 Indexers
Total Users: up to 8	1 combined instance ✓	1 Search Head, 1 Indexers ✓	1 Search Head, 2 Indexers ✓	1 Search Head, 3 Indexers	1 Search Head, 8 Indexers	1 Search Head, 12 Indexers
Total Users: up to 16	1 Search Head, 1 Indexers ✓	1 Search Head, 1 Indexers ✓	1 Search Head, 3 Indexers	2 Search Heads, 4 Indexers	2 Search Heads, 10 Indexers	2 Search Heads, 15 Indexers
Total Users: up to 24	1 Search Head, 1 Indexers ✓	1 Search Head, 2 Indexers ✓	2 Search Heads, 3 Indexers	2 Search Heads, 6 Indexers	2 Search Heads, 12 Indexers	3 Search Heads, 18 Indexers
Total Users: up to 48	1 Search Head, 2 Indexers ✓	1 Search Head, 2 Indexers ✓	2 Search Heads, 4 Indexers	2 Search Heads, 7 Indexers	3 Search Heads, 14 Indexers	3 Search Heads, 21 Indexers

Table 8 is taken from the Splunk summary of performance recommendations. The green checkmarks indicate the daily indexing volumes best suited to the solution described in this paper.

NetApp HCI Configuration Sizes

NetApp HCI is available in three sizes:

- **Small.** 16 cores for compute, 5.5TB effective storage capacity
- **Medium.** 24 cores for compute, 11TB effective storage capacity
- **Large.** 36 cores for compute, 22TB effective storage capacity

Table 9) NetApp HCI/Splunk/Arrow solution configuration table.

	S	M	L
Sizing Requirement	**2 to 300GB/Day (single/combined)**	**700GB/Day (distributed)**	**1TB/Day (distributed)**
Hot/Warm Bucket Retention	30 Days	30 Days	30 Days
Cold Bucket Retention (Configurable)	1 Year	1 Year	1 Year
Splunk Capacity Required for Hot/Warm Buckets* **	4.4TB	10.3TB	14.6TB
Splunk Capacity Required for Cold Buckets* **	52.7TB	123TB	175,8
Number of active users	8	8	8
Number/Type of Compute Nodes	2 x H300e	3 x H300e	2 x H700e
Number/Type of Storage Nodes	4 x H300s	4 x H500s	6 x H500s
Cold Storage Extension Block	**1 x** E2812 (12 x 8TB NL-SAS Drives)	**2 x** E2812 (12 x 8TB NL-SAS Drives) = 24 Drives	**3 x** E2812 (12 x 8TB NL-SAS Drives) = 36 Drives
Total Usable Capacity Provided on HCI (TB)	5.28TB	10.56TB	15.84TB
Number of Splunk Instances	Single Instance combined search head and indexer	Search Head(16 vCores): 1 Indexer(16 vCores): 3 Cluster Master(8 vCores): 1 vCenter(8 vCores): 1	Search Head(16 vCores): 1 Indexer(16 vCores): 5 Cluster Master(8 vCores): 1 vCenter(8 vCores): 1
Number of Cores/vCPUs Provided	32	96	128
Number of Cores/vCPUs Required	32	80	112
Memory Per node	384GB per H300e Node		
Network	4 x 10/25 GbE (SFP 28), 2 x 1 GbE RJ45		
Software	Splunk Enterprise Splunk Universal Forwarder Red Hat® Enterprise Linux® 64-bit VMware vSphere® Enterprise VMware vCenter Server® NetApp SolidFire Element OS® 10.0		

Figure 18) NetApp/Splunk/Arrow solution hardware diagram.

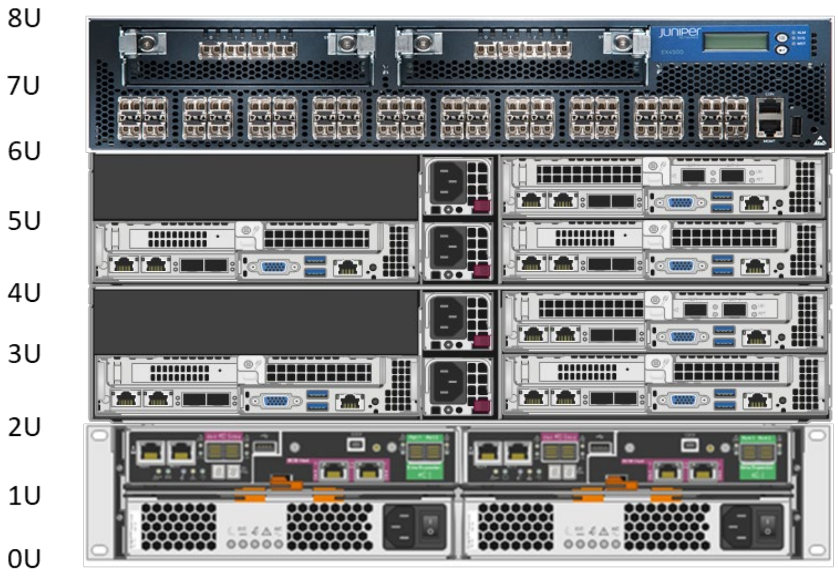


Figure 18 includes the E-series E2812 for the cold-storage tier part of the solution. This was not included in the tested configuration.

Splunk Recommended Configuration

In a minimum NetApp HCI configuration of two H300e compute nodes and four H300s nodes, NetApp recommends the following Splunk setup.

First Compute Node

Table 10) Recommended configuration for first compute node.

Splunk Component	Task	Qty	Cores	Memory
Indexer	Manages the user data	1	16 vCores	32Gb RAM
Search head	User front end, searches data in indexers	1	16 vCores	32Gb RAM

Second Compute Node

Table 11) Recommended configuration for second compute node.

Splunk Component	Task	Qty	Cores	Memory
Indexer	Manages the user data	1	16 vCores	32Gb RAM
Cluster master	Manages the Splunk installation and Indexers	1	8 vCores	16Gb RAM
vCenter	VMware management interface	1	2 to 8 vCores	8Gb to 16Gb RAM

The configuration described in Table 11 supports an ingest rate of at least 300GB per day to 600GB per day with four to eight concurrent searches per user.

11 Conclusion

The NetApp HCI solution scales in a linear manner for Splunk workloads. Despite software vendors expectations, compression of data still shows reasonable efficiency. Strictly dividing storage from compute resources results in predictable sizing, no waste of hypervisor resources, and higher efficiency.

A high-performance, flexible infrastructure solution from NetApp combined with the Splunk analysis platform gets more value from your machine data. Splunk on NetApp HCI provides real-time monitoring for business process optimization in an integrated system without a single point of failure. Combined with nearly unlimited data storage, this solution meets the highest requirements for data protection and storage.

The benefits of the Splunk NetApp HCI solution from Arrow are clear:

- Acceleration of secure analysis functions with a lower total cost of ownership
- An operational platform with scalable real-time monitoring and batch-oriented analysis
- Increased performance and a reduction in the number of commodity servers to HCI compute nodes (5 to 1).
- Automatically expandable to different performance data containers (hot, warm, cold, or frozen data)
- Low latency thanks to the efficient NetApp HCI core systems

- NetApp HCI can replace existing data analytics systems, delivering a reduction of up to 85% in the total cost of ownership for hardware.
- Up to 70% higher search performance with a duplicate data copy
- A single system manufacturer for servers and storage
- Easy transforming and flexible scaling with the Splunk NetApp HCI solution from Arrow

In addition, efficiency increases at the operational level result in significant cost savings. Splunk operations in a wide variety of business scenarios have guaranteed performance with NetApp HCI. As a part of the NetApp Data Fabric, Splunk data can be accessed across various multicloud environments (private, public, and hybrid) in a highly flexible manner.

NetApp HCI offers a cloud-like infrastructure with integrated computing, storage, and network resources in one system. The system is based on the NetApp SolidFire technology and an architecture developed by NetApp. NetApp has developed a scalable and easy to manage component solution, suitable for versatile applications and companies of any size. This solution can be flexibly integrated into any data center, from the simplest to the most complex network. It offers guaranteed application performance with sophisticated integrated replication, efficiency, data protection, and high availability services.

NetApp offers the NetApp HCI system in a minimum configuration of two RU-4N chassis with two computing nodes and four storage nodes. From this basic configuration, any number of computing nodes and storage nodes can be added and combined with one another.

Overall, NetApp HCI brings together and supports the following technologies:

- **NetApp Element software.** Storage software for scale-out block storage.
- **Intuitive (NDE) implementation engine.** Implements and configures components within storage and computing resources.
- **NetApp Monitoring Agent.** Monitors NetApp HCI and storage resources and sends information to vCenter and NetApp Active IQ.
- **SolidFire vCenter Plug-in.** Comprehensive selection of granular storage management functions.
- **SolidFire Management Node.** VM used to monitor and update NetApp HCI systems that also enables remote support.
- **VMware ESXi and vCenter v6.** Host virtualization and management software
- **NetApp Data Fabric.** Integration achieved through the NetApp Element software, including NetApp SnapMirror®, NetApp SnapCenter®, NetApp ONTAP® Select file services, NetApp Cloud Backup, and the NetApp StorageGRID® backup solution.

An intuitive UI makes centralized management easy with the VMware vCenter Plug-in, which in turn guarantees complete control of the entire infrastructure. The integration of NetApp ONTAP Select opens a range of deployment opportunities for existing NetApp customers, and anyone else looking to modernize their data center.

Benefits of the NetApp HCI solution

The primary benefits of NetApp HCI are as follows:

- **Guaranteed performance.** Through the consolidation of heterogeneous workloads and finely tuned performance management and control at the VM level.
- **Flexibility and scaling.** Through the optimization and protection of existing investments, and the independent scaling of computing and storage resources.
- **Automated infrastructure.** Through automated and optimized management, fast implementation, and simplification using a comprehensive API ecosystem.
- **The NetApp Data Fabric.** Through which company data can be moved freely as needed between private, regional, and public cloud architectures, providing greater data sovereignty and security.

NetApp solutions are among the world's most frequently used data management and storage architectures for processing, saving, and storing data securely, efficiently, and at the highest levels of performance. With guaranteed performance, even when running multiple applications, you can be confident using NetApp HCI for your entire data center.

Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp HCI, Splunk, and Arrow solution
  https://www.arrowecs.de/loesungen/arrow-solutions/splunk-netapp
- NetApp HCI Documentation Center
  http://docs.netapp.com/hci/index.jsp
- Splunk Reference hardware
  http://docs.splunk.com/Documentation/Splunk/7.0.2/Capacity/Referencehardware
- Splunk Summary of performance recommendations
  http://docs.splunk.com/Documentation/Splunk/7.0.2/Capacity/Summaryofperformancerecommendations

Version History

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

Appendix

Eventgen Information

The following section presents the input information for the configuration file used by the eventgen application for Splunk to generate data to ingest into the Splunk indexers. This data is used to replicate real-world log type data that a typical Splunk configuration would ingest and to create the index database available for searching.

```
Mount Point
Index volume: /opt/splunk/var/lib/splunk/
Input volume: /mnt/data

Data generation:

Eventgen: on the base Splunk SPL Examples

isabled = false
debug = false
generatorWorkers = 4

outputMode = splunkstream
index = splexamples

mode = sample
sampletype = raw
interval = 1
delay = 0
timeMultiple = 1
timeField = _raw
count = 20
hourOfDayRate = { "0": 0.8, "1": 1.0, "2": 0.9, "3": 0.7, "4": 0.5, "5": 0.4, "6": 0.4, "7": 0.4,
"8": 0.4, "9": 0.4, "10": 0.4, "11": 0.4, "12": 0.4, "13": 0.4, "14": 0.4, "15": 0.4, "16": 0.4,
"17": 0.4, "18": 0.4, "19": 0.4, "20": 0.4, "21": 0.4, "22": 0.5, "23": 0.6 }
dayOfWeekRate = { "0": 0.7, "1": 0.7, "2": 0.7, "3": 0.5, "4": 0.5, "5": 1.0, "6": 1.0 }
randomizeCount = 0.2
randomizeEvents = true
earliest = -300s
latest = now

############################
####    Mail Server     ####
############################

[mailsv.secure.log]
source = /mailsv/secure.log
sourcetype = secure
host = mailsv

backfill = -48h
backfillSearch = index=splexamples source=/mailsv/secure.log

## replace timestamp Thu Dec 31 2014 00:15:06
token.0.token = \w{3}\s+\w{3}\s+\d{2}\s+\d{4}\s+\d{2}:\d{2}:\d{2}
token.0.replacementType = timestamp
token.0.replacement = %a %b %d %Y %H:%M:%S

outputMode = file
fileName = /mnt/data/gen/mail_secure.log

############################
####    Vendor Sales    ####
############################

[vendor_sales.log]
source = /vendor_sales/vendor_sales.log
sourcetype = vendor_sales
host = vendor_sales
```

```
backfill = -48h
backfillSearch = index=splexamples source=/vendor_sales/vendor_sales.log

## replace timestamp [24/Dec/2014:18:23:07]
token.1.token = \d{2}\/\w{3}\/\d{4}:\d{2}:\d{2}:\d{2}
token.1.replacementType = timestamp
token.1.replacement = %d/%b/%Y:%H:%M:%S

outputMode = file
fileName = /mnt/data/gen/vendor_sales.log

############################
####    APACHE NOISE    ####
############################

[www1.access.log]
source = /www1/access.log
sourcetype = access_combined_wcookie
host = www1

backfill = -48h
backfillSearch = index=splexamples source=/www1/access.log

## replace timestamp [24/Dec/2014:18:23:07]
token.2.token = \d{2}\/\w{3}\/\d{4}:\d{2}:\d{2}:\d{2}
token.2.replacementType = timestamp
token.2.replacement = %d/%b/%Y:%H:%M:%S

outputMode = file
fileName = /mnt/data/gen/www1.access.log

[www2.access.log]
source = /www2/access.log
sourcetype = access_combined_wcookie
host = www2

backfill = -48h
backfillSearch = index=splexamples source=/www2/access.log

## replace timestamp [24/Dec/2014:18:23:07]
token.4.token = \d{2}\/\w{3}\/\d{4}:\d{2}:\d{2}:\d{2}
token.4.replacementType = timestamp
token.4.replacement = %d/%b/%Y:%H:%M:%S

outputMode = file
fileName = /mnt/data/gen/www2.access.log

[www3.access.log]
source = /www3/access.log
sourcetype = access_combined_wcookie
host = www3

backfill = -48h
backfillSearch = index=splexamples source=/www3/access.log

## replace timestamp [24/Dec/2014:18:23:07]
token.6.token = \d{2}\/\w{3}\/\d{4}:\d{2}:\d{2}:\d{2}
token.6.replacementType = timestamp
token.6.replacement = %d/%b/%Y:%H:%M:%S

outputMode = file
fileName = /mnt/data/gen/www3.access.log

############################
####    SECURE NOISE    ####
############################

[www1.secure.log]
source = /www1/secure.log
sourcetype = secure
```

```
host = www1

backfill = -48h
backfillSearch = index=splexamples source=/www1/secure.log

## replace timestamp Thu Dec 31 2014 00:15:06
token.3.token = \w{3}\s+\w{3}\s+\d{2}\s+\d{4}\s+\d{2}:\d{2}:\d{2}
token.3.replacementType = timestamp
token.3.replacement = %a %b %d %Y %H:%M:%S

outputMode = file
fileName = /mnt/data/gen/www1.secure.log

[www2.secure.log]
source = /www2/secure.log
sourcetype = secure
host = www2

backfill = -48h
backfillSearch = index=splexamples source=/www2/secure.log

## replace timestamp Thu Dec 31 2014 00:15:06
token.5.token = \w{3}\s+\w{3}\s+\d{2}\s+\d{4}\s+\d{2}:\d{2}:\d{2}
token.5.replacementType = timestamp
token.5.replacement = %a %b %d %Y %H:%M:%S

outputMode = file
fileName = /mnt/data/gen/www2.secure.log

[www3.secure.log]
source = /www3/secure.log
sourcetype = secure
host = www3

backfill = -48h
backfillSearch = index=splexamples source=/www3/secure.log

## replace timestamp Thu Dec 31 2014 00:15:06
token.7.token = \w{3}\s+\w{3}\s+\d{2}\s+\d{4}\s+\d{2}:\d{2}:\d{2}
token.7.replacementType = timestamp
token.7.replacement = %a %b %d %Y %H:%M:%S

outputMode = file
fileName = /mnt/data/gen/www3.secure.log

################################
####    Earthquake Noise    ####
################################

[earthquake.log]
source = usgs
sourcetype = earthquake-csv
host = earthquake

backfill = -7d
backfillSearch = index=splexamples source=usgs

##replace timestamp 2015-07-29T08:33:25.000
token.8.token = \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}
token.8.replacementType = timestamp
token.8.replacement = %Y-%m-%dT%H:%M:%S.%N

outputMode = file
fileName = /mnt/data/gen/earthquake.log
################################
####       Email Noise      ####
################################

[email.log]
source = email.log
sourcetype = email
```

```
host = exchange01

backfill = -48h
backfillSearch = index=splexamples source=email.log

##replace timestamp 2016-01-20T16:34:42.267+0000
token.8.token = \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}
token.8.replacementType = timestamp
token.8.replacement = %Y-%m-%dT%H:%M:%S.%N


outputMode = file
fileName = /mnt/data/gen/mail.log

###############################
####    NIX Performance    ####
###############################

[linux.vmstat]
disabled = false
mode = sample
sampletype = csv
outputMode = modinput
interval = 12
count = 9
earliest = -12s
latest = now

backfill = -60m
backfillSearch = index=splexamples sourcetype=vmstat2

host.token = (__SERVERNAME__)
host.replacement = $SPLUNK_HOME/etc/apps/spl_examples/eventgen_replace/webhosts.csv:2

token.0.token = \d{2}-\d{2}-\d{4} \d{2}:\d{2}:\d{2}
token.0.replacementType = timestamp
token.0.replacement = %m-%d-%Y %H:%M:%S

token.1.token = loadAvg1mi
token.1.replacementType = random
token.1.replacement = float[0.0:5.0]

token.2.token = memTotalMB
token.2.replacementType = random
token.2.replacement = integer[2000:4000]

token.3.token = memFreeMB
token.3.replacementType = random
token.3.replacement = integer[0:2000]

token.4.token = memUsedMB
token.4.replacementType = random
token.4.replacement = integer[1000:7000]

token.5.token = memUsedPct
token.5.replacementType = random
token.5.replacement = float[1.0:75.0]

token.6.token = memFreePct
token.6.replacementType = random
token.6.replacement = float[1.0:99.0]

token.7.token = pgPageOut
token.7.replacementType = random
token.7.replacement = integer[1000000:10000000]

token.8.token = swapUsedPct
token.8.replacementType = random
token.8.replacement = float[10.0:19.9]

token.9.token = pgSwapOut
token.9.replacementType = random
token.9.replacement = integer[10000:100000]

token.10.token = cSwitches
token.10.replacementType = random
token.10.replacement = integer[1000000:5000000]

token.11.token = interrupts
token.11.replacementType = random
token.11.replacement = integer[1000000:4000000]

token.12.token = forks
token.12.replacementType = random
token.12.replacement = integer[10000:100000]

token.13.token = processes
token.13.replacementType = random
token.13.replacement = integer[100:500]

token.14.token = threads
token.14.replacementType = random
token.14.replacement = integer[1000:3000]

###################################
####               CPU         ####
###################################
[noise.cpu]
disabled = false
mode = sample
sampletype = csv
outputMode = modinput
count = 30
interval = 20
earliest = -20s

backfill = -60m
backfillSearch = index=splexamples sourcetype=cpu2

host.token = (__SERVERNAME__)
host.replacement = $SPLUNK_HOME/etc/apps/spl_examples/eventgen_replace/webhosts.csv:2

token.0.token = \d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}
token.0.replacementType = timestamp
token.0.replacement = %Y-%m-%d %H:%M:%S

token.1.token = pctUser
token.1.replacementType = random
token.1.replacement = float[17.2:34.3]

token.2.token = pctNice
token.2.replacementType = static
token.2.replacement = 0

token.3.token = pctSystem
token.3.replacementType = random
token.3.replacement = float[6.3:20.7]

token.4.token = pctIowait
token.4.replacementType = random
token.4.replacement = float[0.1:1.1]

token.5.token = pctIdle
token.5.replacementType = random
token.5.replacement = float[49.2:86.3]

#########################################
####                DF               ####
#########################################
[noise.df]
disabled = false
mode = sample
```

```
sampletype = csv
outputMode = modinput
count = 36
interval = 20
earliest = -20s

backfill = -60m
backfillSearch = index= splexamples source=df

host.token = (__SERVERNAME__)
host.replacement = $SPLUNK_HOME/etc/apps/spl_examples/eventgen_replace/webhosts.csv:2

token.0.token = \d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}
token.0.replacementType = timestamp
token.0.replacement = %Y-%m-%d %H:%M:%S

token.1.token = (@Used)
token.1.replacementType = random
token.1.replacement = float[494.4:541.8]

token.2.token = (@Avail)
token.2.replacementType = random
token.2.replacement = float[389.2:436.6]

token.3.token = (@UsePct)
token.3.replacementType = random
token.3.replacement = float[41.8:46.9]

###################################
####           IOSTAT          ####
###################################
[noise.iostat]
disabled = false
mode = sample
sampletype = csv
outputMode = modinput
count = 35
interval = 20
earliest = -20s

backfill = -60m
backfillSearch = index= splexamples source=iostat
host.token = (@host)


Input.conf:

[monitor:///mnt/data/data0]
disabled = false
host = datahost0
index = test01_idx

[monitor:///mnt/data/data1]
disabled = false
host = datahost1
index = test01_idx

[monitor:///mnt/data/data2]
disabled = false
host = datahost2
index = test01_idx

[monitor:///mnt/data/data3]
disabled = false
host = datahost3
index = test01_idx

[monitor:///mnt/data/data4]
disabled = false
host = datahost4
index = test01_idx
```

```
[monitor:///mnt/data/data5]
disabled = false
host = datahost5
index = test01_idx

[monitor:///mnt/data/data6]
disabled = false
host = datahost6
index = test01_idx

[monitor:///mnt/data/data7]
disabled = false
host = datahost7
index = test01_idx

[monitor:///mnt/data/data8]
disabled = false
host = datahost8
index = test01_idx

[monitor:///mnt/data/data9]
disabled = false
host = datahost9
index = test01_idx

[monitor:///mnt/data/data10]
disabled = false
host = datahost10
index = test02_idx

[monitor:///mnt/data/data11]
disabled = false
host = datahost11
index = test02_idx

[monitor:///mnt/data/data12]
disabled = false
host = datahost12
index = test02_idx

[monitor:///mnt/data/data13]
disabled = false
host = datahost13
index = test02_idx

[monitor:///mnt/data/data14]
disabled = false
host = datahost14
index = test02_idx

[monitor:///mnt/data/data15]
disabled = false
host = datahost15
index = test02_idx

[monitor:///mnt/data/data16]
disabled = false
host = datahost16
index = test02_idx

[monitor:///mnt/data/data17]
disabled = false
host = datahost17
index = test02_idx

[monitor:///mnt/data/data18]
disabled = false
host = datahost18
index = test02_idx
```

```
[monitor:///mnt/data/data19]
disabled = false
host = datahost19
index = test02_idx

Modify files:

#!/usr/bin/perl
use Fcntl qw( SEEK_SET );

sub mod_first_line{
        my @files = glob($_[0]);

        foreach $file (@files) {
          print "File: $file\n" if -f $file;
          if (-f $file) {
                open (FH, "+< $file") || die "can't open $file: $!";
                my $line = <FH>;
                $random_txt=sprintf("%f",rand());
                $position=length($line)-1-length($random_txt);
#               $line = $line.' '.$random_txt.'\n';
                seek FH, $position, SEEK_SET; # go back to the start of the file
                printf FH $random_txt;
#               printf FH $line;
                close FH;
          }
          print "dir: $file\n" if -d $file;
          mod_first_line($file.'/*') if -d $file;
        }
}

mod_first_line('/mnt/data/*');
# mod_first_line('/tmp/data/*');
```

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

Copyright Information

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NetApp®