Basic Concepts for NetApp ONTAP 9.7

NAS Services, CLI Edition

April 2023 | LD00378 Version 2.2.0

TABLE OF CONTENTS

1 Introduction

This lab introduces the fundamentals of NetApp ONTAP®. In it you begin with a pre-created 2-node ONTAP cluster, provision storage for access through CIFS and NFS, configure a Windows 2019 client to access the storage over CIFS, and configure a Red Hat Enterprise Linux 7.5 client to access the storage over NFS.

There is a similar lab available that demonstrates how to configure these same clients to access storage on the cluster using SAN protocols instead, specifically using iSCSI. The NAS and SAN lab guides are designed so they can both be completed within the same copy of the lab. If that option interests you, then for the best user experience we recommend that you complete the NAS guide first, followed by the SAN guide, as that ordering will most closely match the workflow examples documented in the lab guides.

1.1 Why NetApp ONTAP?

One of the key ways to understand the benefits of ONTAP is to consider server virtualization. Before server virtualization, system administrators frequently deployed applications on dedicated servers in order to maximize application performance, and to avoid the instabilities often encountered when combining multiple applications on the same operating system instance. While this design approach was effective, it also had the following drawbacks:

- It did not scale well — adding new servers for every new application was expensive.
- It was inefficient — most servers are significantly under-utilized, and businesses are not extracting the full benefit of their hardware investment.
- It was inflexible — re-allocating standalone server resources for other purposes is time consuming, staff intensive, and highly disruptive.

Server virtualization directly addresses these limitations by decoupling the application instance from the underlying physical hardware. Multiple virtual servers can share a pool of physical hardware, allowing businesses to consolidate their server workloads to a smaller set of more effectively utilized physical servers. Additionally, the ability to transparently migrate running virtual machines across a pool of physical servers reduces the impact of downtime due to scheduled maintenance activities.

NetApp ONTAP brings these same benefits, and many others, to storage systems. As with server virtualization, ONTAP enables you to combine multiple physical storage controllers into a single logical cluster that can non-disruptively service multiple storage workload needs. With ONTAP you can:

- Combine different types and models of NetApp storage controllers (known as nodes) into a shared physical storage resource pool (referred to as a cluster).
- Support multiple data access protocols (CIFS, NFS, Fibre Channel, iSCSI, NVMeFC, etc.) concurrently on the same storage cluster.
- Consolidate various storage workloads to the cluster. Each workload can be assigned its own Storage Virtual Machine (SVM), which is essentially a dedicated virtual storage controller, and its own data volumes, LUNs, CIFS shares, and NFS exports.
- Support multi-tenancy with delegated administration of SVMs. Tenants can be different companies, business units, or even individual application owners, each with their own distinct administrators whose admin rights are limited to just the assigned SVM.
- Use Quality of Service (QoS) capabilities to manage resource utilization between storage workloads.
- Non-disruptively migrate live data volumes and client connections from one cluster node to another.
- Non-disruptively scale the cluster out by adding nodes. Nodes can likewise be non-disruptively removed from the cluster, meaning that you can non-disruptively scale a cluster up and down during hardware refresh cycles.
- Leverage multiple nodes in the cluster to simultaneously service a given SVM's storage workloads. This means that businesses can scale out their SVMs beyond the bounds of a single physical node in response to growing storage and performance requirements, all non-disruptively.
- Apply software and firmware updates, and configuration changes without downtime.

1.2 Lab Objectives

This lab explores fundamental concepts of ONTAP by walking you through the initial setup of a cluster, with a focus on support for NAS protocols (CIFS and NFS).

Here is a summary of the exercises in this lab, along with their Estimated Completion Times (ECT):

- Clusters (Required, ECT = 20 minutes).
  - Explore a cluster.
  - View Advanced Drive Partitioning.
  - Create a data aggregate.
  - Create a Subnet.
- Storage Virtual machines for NFS and CIFS (ECT = 40 minutes)
  - Create a Storage Virtual Machine.
  - Configure the Storage Virtual Machine for CIFS and NFS access.
  - Create a volume on the Storage Virtual Machine.
  - Mount a CIFS share from the Storage Virtual Machine on a Windows client.
  - Mount a NFS volume from the Storage Virtual Machine on a Linux client.
  - Create and NFS-export Qtrees. (Optional)

There are two versions of this lab guide, one uses System Manager (NetApp's graphical administration interface) to complete these tasks, while the other uses the ONTAP Command Line Interface (CLI). The lab end state produced by either method is exactly the same, so use the method you are the most comfortable with by selecting the appropriate lab guide.

Note: The System Manager version of this lab demonstrates the “classic” System Manager interface that is standard in ONTAP 9.7 and earlier ONTAP versions. It does not discuss the “new experience” System Manager interface that is previewed in ONTAP 9.7 and that becomes standard in ONTAP 9.8.

1.3 Prerequisites

This lab introduces NetApp ONTAP, and makes no assumptions that the user has previous experience with ONTAP. The lab does assume some basic familiarity with storage system related concepts such as RAID, CIFS, NFS, LUNs, and DNS.

This lab includes steps for mapping shares and/or mounting LUNs on a Windows client. These steps assume that the lab user has a basic familiarity with Microsoft Windows.

This lab also includes steps for mounting NFS volumes and/or LUNs on a Linux client. All steps are performed from the Linux command line, and assumes a basic working knowledge of the Linux command line. A basic working knowledge of a text editor such as vi may be useful, but is not required.

1.4 Accessing the Command Line

PuTTY is the terminal emulation program used in the lab to log into Linux hosts and storage controllers in order to run command line commands.

1. The launch icon for the PuTTY application is pinned to the task bar on the Windows host Jumphost as shown in the following screen shot; just double-click on the icon to launch it.

   Tip: If you already have a PuTTY session open and you want to start another (even to a different host), right-click the PuTTY icon and select **PuTTY** from the context menu.

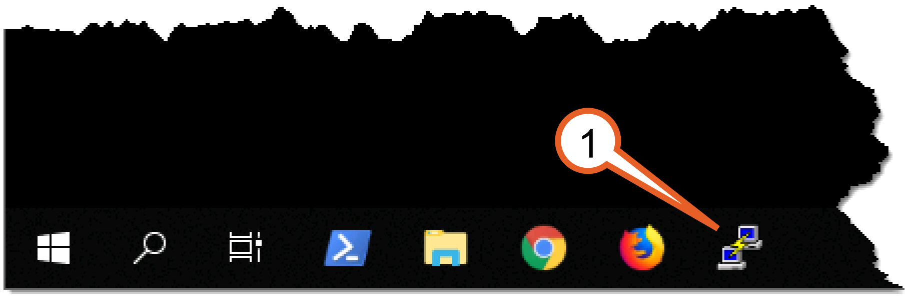

Figure 1-1:

Once PuTTY launches you need to select the lab host you want to connect to. The following example shows how to connect to the ONTAP cluster named “cluster1”.

2. By default PuTTY should launch into the “Basic options for your PuTTY session” display as shown in the screen shot. If you accidentally navigate away from this view just click on the **Session** category item to return to this view.
3. Use the scrollbar in the “Saved Sessions” box to navigate down to the desired host and double-click it to open the connection. A terminal window will open and you will be prompted to log into the host. You can find the

correct username and password for the host in the Lab Host Credentials table found in the “Lab Environment” section of this guide.

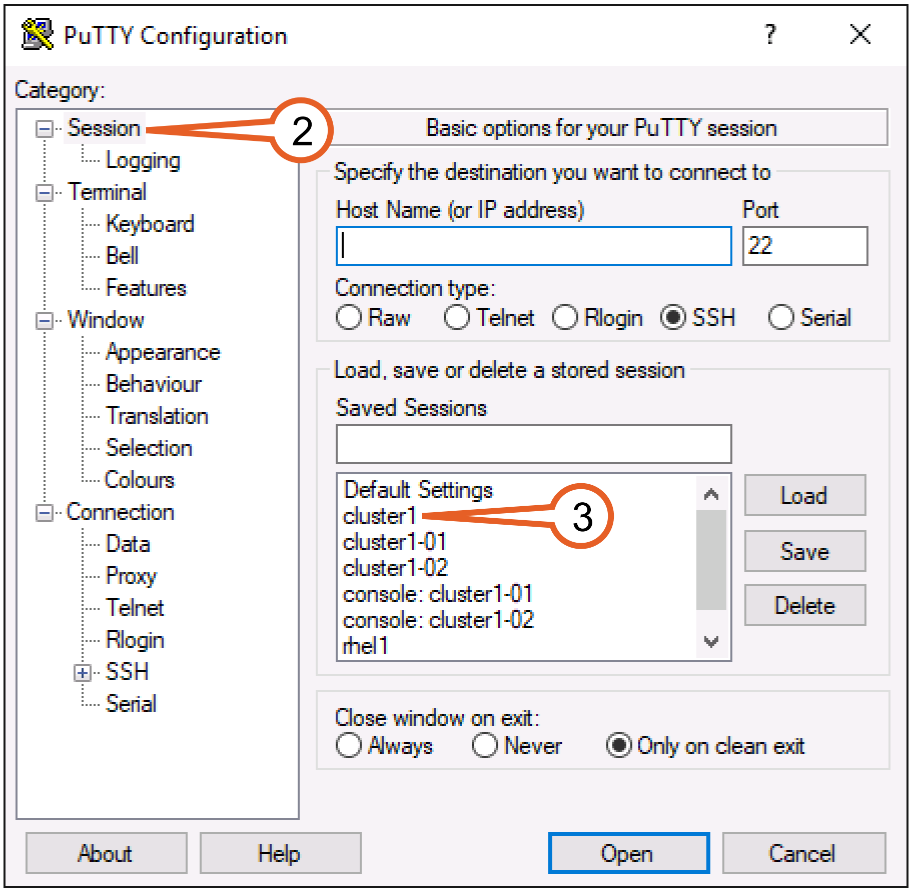

Figure 1-2:

If you are new to the ONTAP CLI, the length of the commands can seem a little intimidating. However, the commands are actually quite easy to use if you remember the following three tips:

- Make liberal use of the **Tab** key while entering commands, as the ONTAP command shell supports tab completion. If you hit the **Tab** key while entering a portion of a command word, the command shell will examine the context and try to complete the rest of the word for you. If there is insufficient context to make a single match, it will display a list of all the potential matches. Tab completion also usually works with command *argument values*, but there are some cases where there is simply not enough context for it to know what you want, in which case you will just need to type in the *argument value*.
- You can recall your previously entered commands by repeatedly pressing the **up-arrow** key, and you can then navigate up and down the list using the **up-arrow** and **down-arrow** keys. When you find a command you want to modify, you can use the **left-arrow**, **right-arrow**, and **Delete** keys to navigate around in a selected command to edit it.
- Entering a question mark character (**?**) causes the CLI to print contextual help information. You can use this character on a line by itself, or while entering a command.

The ONTAP command line supports additional usability features that make the command line easier to use. If you are interested in learning more about this topic, refer to the “Hands-On Lab for Advanced Features of ONTAP” lab, which contains an entire section dedicated to this subject.

Note: In this lab, ONTAP automatically terminates ssh sessions to cluster1 after 2 hours of idle time. If you would prefer your PuTTY sessions to cluster1 to stay open indefinitely, issue the `system timeout modify 0` command at the ONTAP CLI.

2 Lab Environment

The following figure contains a diagram of the environment for this lab.

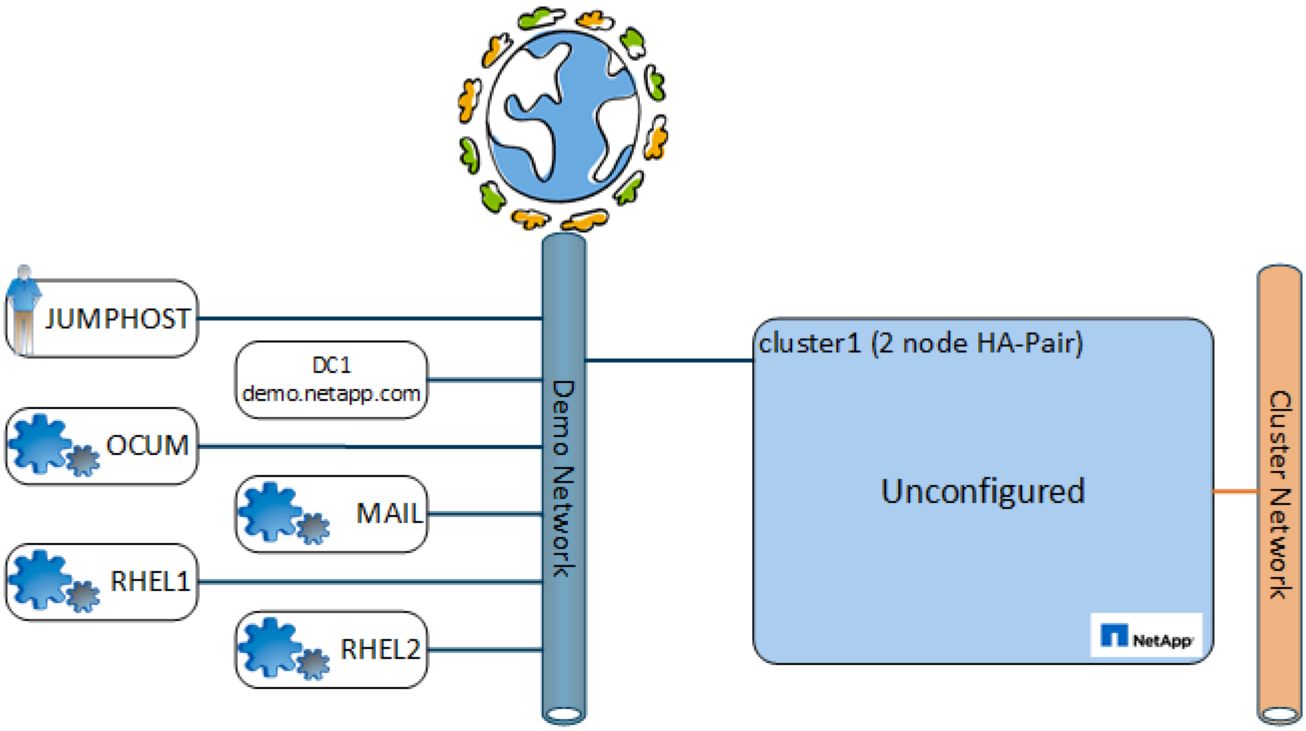

Figure 2-1:

All of the servers and storage controllers presented in this lab are virtual devices, and the networks that interconnect them are exclusive to your lab session. While we encourage you to follow the demonstration steps outlined in this lab guide, you are free to deviate from this guide and experiment with other ONTAP features that interest you. While the virtual storage controllers (vsims) used in this lab offer nearly all of the same functionality as physical storage controllers, they are not capable of providing the same performance as a physical controller, which is why these labs are not suitable for performance testing.

Table 1 provides a list of the servers and storage controller nodes in the lab, along with their IP address.

Table 1: Table 1: Lab Host Credentials

Hostname	Description	IP Address(es)	Username	Password
JUMPHOST	Windows 2019 Remote Access host	192.168.0.5	DEMO\Administrator	Netapp1!
RHEL1	Red Hat 7.5 x64 Linux host	192.168.0.61	root	Netapp1!
RHEL2	Red Hat 7.5 x64 Linux host	192.168.0.62	root	Netapp1!
DC1	Windows 2019 Active Directory Server	192.168.0.253	DEMO\Administrator	Netapp1!
cluster1	ONTAP 9.5 cluster	192.168.0.101	admin	Netapp1!
cluster1-01	ONTAP 9.5 cluster node	192.168.0.111	admin	Netapp1!
cluster1-02	ONTAP 9.5 cluster node	192.168.0.112	admin	Netapp1!
MAIL	E-Mail server/web email client	192.168.0.89	admin	Netapp1!

Hostname	Description	IP Address(es)	Username	Password
OCUM	OnCommand Unified Manager v9.5	192.168.0.71	DEMO\Administrator	Netapp1!

Table 2 lists the NetApp software that is pre-installed on the various hosts in this lab.

Table 2: Table 2: Preinstalled NetApp Software

Hostname	Description
JUMPHOST	Windows Unified Host Utility Kit v7.1.0, NetApp PowerShell Toolkit v4.7.0
RHEL1, RHEL2	Linux Unified Host Utilities Kit v7.1

3 Lab Activities

3.1 Clusters

Expected Completion Time: 20 Minutes

3.1.1 Cluster Concepts

A cluster is a group of physical storage controllers, or nodes, that are joined together for the purpose of serving data to end users. The nodes in a cluster can pool their resources together so that the cluster can distribute its work across the member nodes. Communication and data transfer between member nodes (such as when a client accesses data on a node other than the one actually hosting the data) takes place over a high-speed cluster-interconnect network (10 to 100 Gb Ethernet) to which all the nodes are connected, while management and client data traffic passes over separate management and data networks configured on the member nodes.

Clusters typically consist of one, or more, NetApp storage controller High Availability (HA) pairs. Both controllers in an HA pair actively host and serve data, but they are also capable of taking over their partner's responsibilities in the event of a service disruption by virtue of their redundant cable paths to each other's disk storage. Having multiple HA pairs in a cluster allows the cluster to scale out to handle greater workloads, and to support non-disruptive migrations of volumes and client connections to other nodes in the cluster resource pool. This means that cluster expansion and technology refreshes can take place while the cluster remains fully online, and serving data.

Since clusters are almost always comprised of one or more HA pairs, a cluster almost always contains an even number of controller nodes. There is one exception to this rule, the “single node cluster”, which is a special cluster configuration that supports small storage deployments using a single physical controller head. The primary difference between single node and standard clusters, besides the number of nodes, is that a single node cluster does not have a cluster network. Single node clusters can be converted into traditional multi-node clusters, at which point they become subject to all the standard cluster requirements like the need to utilize an even number of nodes consisting of HA pairs. This lab does not contain a single node cluster, so does not discuss them further.

ONTAP 9 clusters that only serve NFS and CIFS can scale up to a maximum of 24 nodes, although the node limit can be lower depending on the model of FAS controller and the specific ONTAP version in use. ONTAP 9 clusters that also host iSCSI and FC can scale up to a maximum of 12 nodes, but once again the limit may be lower depending on the FAS controller model.

This lab utilizes simulated NetApp storage controllers rather than physical FAS controllers. The simulated controller, also known as a “VSIM”, is a virtual machine that simulates the functionality of a physical controller without the need for dedicated controller hardware. The vsim is not designed for performance testing, but does offer much of the same functionality as a physical FAS controller, including the ability to generate I/O to disks. This makes the vsim a powerful tool to explore and experiment with ONTAP product features. The vsim is limited when a feature requires a specific physical capability that the vsim does not support. For example, vsims do not support Fibre Channel connections, which is why this lab uses iSCSI to demonstrate block storage functionality.

This lab starts with a pre-created, minimally configured cluster. The pre-created cluster already includes ONTAP licenses, the cluster's basic network configuration, and a pair of pre-configured HA controllers. In this next section you will create the aggregates that are used by the SVMs that you will create in later sections of the lab. You will also take a look at the Advanced Drive Partitioning feature.

3.1.2 Advanced Drive Partitioning

3.1.2.1 Advanced Drive Partitioning Concepts

Disks, whether Hard Disk Drives (HDD) or Solid State Disks (SSD), are the fundamental unit of physical storage in ONTAP, and are tied to a specific cluster node by virtue of their physical connectivity (i.e., cabling) to a given controller head.

ONTAP manages disks in groups called aggregates. An aggregate defines the RAID properties for a group of disks that are all physically attached to the same node. A given disk can only be a member of a single aggregate.

By default each cluster node has one aggregate known as the root aggregate, which is a group of the node's local disks that host the node's ONTAP operating system. A node's root aggregate is automatically created during ONTAP installation in a minimal RAID-DP configuration This means it is initially comprised of 3 disks (1 data, 2 parity), and it has a name that begins the string aggr0. For example, in this lab the root aggregate of the node cluster1-01 is named “aggr0_cluster1_01”, and the root aggregate of the node cluster1-02 is named “aggr0_cluster1_02”.

On higher end FAS systems that have many disks, the requirement to dedicate 3 disks for each controller's root aggregate is not a burden, but for FAS systems that only have 24 or 12 disks, or for AFF systems with expensive high-capacity SSDs, this root aggregate disk overhead requirement significantly reduces the disks available for storing user data. To improve usable capacity, NetApp introduced Advanced Drive Partitioning (ADP), which divides the disks on nodes that have this feature enabled into two partitions; a small root partition, and a much larger data partition. ONTAP allocates the root partitions to the node root aggregate, and the data partitions for data aggregates. Each partition behaves like a virtual disk, so in terms of RAID, ONTAP treats these partitions just like physical disks when creating aggregates. The key benefit is that a much higher percentage of the node's overall disk capacity is now available to host user data.

For a FAS controller that uses Advanced Drive Partitioning, ONTAP dynamically determines the size of the root and data disk partitions at system installation time based on the quantity and size of the available disks assigned to each node.

ONTAP only supports HDD ADP for entry-level FAS controllers, and only for HDDs installed in their internal shelf on those models. ADP can only be enabled at system installation time, and there is no way to convert an existing system to use ADP other than to completely evacuate the affected HDDs and re-install ONTAP.

All-Flash FAS (AFF) supports ADP V2 that utilizes SSDs instead of HDDs, and divides the SSD into three partitions; a small root partition, plus two equal-sized data partitions. ADP V2 supports entry-level, mid-range, and high-end AFF platforms.

3.1.2.2 Advanced Drive Partitioning Exercise

In this section, you will use the ONTAP CLI to determine if a cluster node is utilizing Advanced Drive Partitioning.

1. If you do not already have a PuTTY session established to cluster1, launch PuTTY as described in the “Accessing the Command Line” section at the beginning of this guide, and connect to the host cluster1 using the username **admin** and the password **Netapp1!**.
2. List all of the physical disks attached to the cluster.

```
cluster1::> storage disk show

 Usable           Disk    Container   Container
Disk                  Size Shelf Bay Type    Type        Name      Owner
---------------- ---------- ----- --- ------- ----------- --------- --------

Info: This cluster has partitioned disks. To get a complete list of spare disk capacity use
 "storage aggregate
      show-spare-disks".
VMw-1.25            28.44GB     -   0 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.26            28.44GB     -   1 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.27            28.44GB     -   2 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.28            28.44GB     -   3 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.29            28.44GB     -   4 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.30            28.44GB     -   5 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.31            28.44GB     -   6 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.32            28.44GB     -   8 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
VMw-1.33            28.44GB     -   9 SSD     shared      aggr0_cluster1_02
                                                                   cluster1-02
```

```
VMw-1.34              28.44GB     -  10 SSD     shared      aggr0_cluster1_02
                                                                      cluster1-02
VMw-1.35              28.44GB     -  11 SSD     shared      -         cluster1-02
VMw-1.36              28.44GB     -  12 SSD     shared      -         cluster1-02
VMw-1.37              28.44GB     -   0 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.38              28.44GB     -   1 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.39              28.44GB     -   2 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.40              28.44GB     -   3 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.41              28.44GB     -   4 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.42              28.44GB     -   5 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.43              28.44GB     -   6 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.44              28.44GB     -   8 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.45              28.44GB     -   9 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.46              28.44GB     -  10 SSD     shared      aggr0_cluster1_01
                                                                      cluster1-01
VMw-1.47              28.44GB     -  11 SSD     shared      -         cluster1-01
VMw-1.48              28.44GB     -  12 SSD     shared      -         cluster1-01
24 entries were displayed.

cluster1::>
```

The “Info” message at the beginning of the command output declares that this system has partitioned disks. Before you run the command the “Info” message recommends, examine the rest of the output produced by this command.

The output listed a total of 24 disks, 12 for each of the nodes in this two-node cluster. The container type for all the disks is “shared”, which indicates that the disks are partitioned. For disks that are not partitioned, you would typically see values like “spare”, “data”, “parity”, and “dparity”. The “Owner” field indicates which node the disk is assigned to, and the “Container Name” field indicates which aggregate the disk is assigned to.

3. List the available spare disks as directed by the previous command output, but limit the context to just the cluster1-01 node.

```
cluster1::> storage aggregate show-spare-disks -owner cluster1-01

Original Owner: cluster1-01
 Pool0
  Root-Data1-Data2 Partitioned Spares
                                                            Local    Local
                                                             Data     Root Physical
 Disk             Type   Class          RPM Checksum       Usable   Usable     Size Status
 ---------------- ------ ----------- ------ -------------- -------- -------- -------- --------
 VMw-1.37         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.38         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.39         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.40         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.41         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.42         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.43         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.44         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.45         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.46         SSD    solid-state      - block           24.80GB       0B  28.44GB zeroed
 VMw-1.47         SSD    solid-state      - block           24.80GB   3.58GB  28.44GB zeroed
 VMw-1.48         SSD    solid-state      - block           24.80GB   3.58GB  28.44GB zeroed
12 entries were displayed.

cluster1::>
```

The “Local Data Usable” column represents available space in each disk’s data partitions. You have not created any data aggregates yet, and none of the data partitions are currently being used, so all of them list an available capacity of 24.80 GB.

The “Local Root Usable” column represents the space available of the disks root partitions. Ten of the disks show that they have no available space because the partitions are being used by cluster1-01’s root aggregate. The other two disks are also partitioned, but each has 3.58 GB of space available because they are configured as spares to protect against potential drive failures.

4. List the aggregates that currently exist on the cluster:

```
cluster1::> aggr show


Aggregate     Size Available Used% State   #Vols  Nodes            RAID Status
--------- -------- --------- ----- ------- ------ ---------------- ------------
aggr0_cluster1_01
           24.32GB    1.18GB   95% online       1 cluster1-01      raid_dp,
                                                                   normal
aggr0_cluster1_02
           24.32GB    1.17GB   95% online       1 cluster1-02      raid_dp,
                                                                   normal
2 entries were displayed.

cluster1::>
```

The only existing aggregates are the node root aggregates.

5. Now list the disks that are members of the root aggregate for the node cluster-01. Here is the command that you would ordinarily use to display that information for an aggregate that is not using partitioned disks.

```
cluster1::> storage disk show -aggregate aggr0_cluster1_01
There are no entries matching your query.

Info: This cluster has partitioned disks. To get a complete list of spare disk capacity use
 "storage aggregate
      show-spare-disks".
      One or more aggregates queried for use shared disks. Use "storage aggregate show-status"
 to get correct set of
      disks associated with these aggregates.

cluster1::>
```

6. As you can see, in this instance the preceding command is not able to produce a list of disks because this aggregate is using shared disks. Instead it refers you to use the `storage aggregate show-status` command to query the aggregate for a list of its assigned disk partitions.

```
cluster1::> storage aggregate show-status -aggregate aggr0_cluster1_01

Owner Node: cluster1-01
 Aggregate: aggr0_cluster1_01 (online, raid_dp) (block checksums)
  Plex: /aggr0_cluster1_01/plex0 (online, normal, active, pool0)
   RAID Group /aggr0_cluster1_01/plex0/rg0 (normal, block checksums)
                                                              Usable Physical
     Position Disk                        Pool Type     RPM     Size     Size Status
     -------- --------------------------- ---- ----- ------ -------- -------- ----------
     shared   VMw-1.37                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.38                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.39                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.40                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.41                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.42                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.43                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.44                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.45                     0   SSD        -   3.58GB  28.44GB (normal)
     shared   VMw-1.46                     0   SSD        -   3.58GB  28.44GB (normal)
10 entries were displayed.

cluster1::>
```

The output shows that aggr0_cluster1_01 is comprised of 10 disks, each with a usable size of 3.58 GB, which is the size of the disks’s root partition.

7. The ONTAP CLI includes a diagnostic level command that provides a more comprehensive single view of a system's partitioned disks. The following commands shows all the disk partitions on the disks that belong to the node cluster1-01.

```
cluster1::> set -priv diag

Warning: These diagnostic commands are for use by NetApp personnel only.
```

```
Do you want to continue? {y|n}: y

cluster1::*> disk partition show -owner-node-name cluster1-01
                          Usable  Container     Container
Partition                 Size    Type          Name              Owner
------------------------- ------- ------------- ----------------- -----------------
VMw-1.37.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.37.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.37.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.38.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.38.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.38.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.39.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.39.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.39.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.40.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.40.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.40.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.41.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.41.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.41.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.42.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.42.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.42.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.43.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.43.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.43.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.44.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.44.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.44.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.45.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.45.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.45.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.46.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.46.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.46.P3                3.58GB aggregate     /aggr0_cluster1_01/plex0/rg0
                                                                  cluster1-01
VMw-1.47.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.47.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.47.P3                3.58GB spare         Pool0             cluster1-01
VMw-1.48.P1               12.41GB spare         Pool0             cluster1-01
VMw-1.48.P2               12.41GB spare         Pool0             cluster1-01
VMw-1.48.P3                3.58GB spare         Pool0             cluster1-01
36 entries were displayed.

cluster1::*> set -priv admin

cluster1::>
```

This command reveals that each disk contains 3 partitions. The *.P3 entries are the partitions that comprise the node’s root aggregate, as their size of 3.58 GB matches the “local root usable” value you saw reported in step 3. The *.P1 and *.P2 entries are 12.41 GB data partitions, which when added together matches the 24.80 “local data usable” value you saw reported for the spare disks in step 3.

3.1.3 Create a New Aggregate on Each Cluster Node

3.1.3.1 Aggregate Concepts

An aggregate is a group of disks on a node. A node can host multiple aggregates depending on the data sizing, performance, and isolation needs of the storage workloads that it will be hosting. When you create a Storage Virtual Machine (SVM) you assign it to use one or more specific aggregates to host the SVM's volumes. You can assign multiple SVMs to use the same aggregate, which offers greater flexibility in managing storage space, whereas dedicating an aggregate to just a single SVM provides greater workload isolation.

Aggregates utilize RAID to span storage across multiple disks. Although ONTAP supports several different RAID configuration options, NetApp recommends using RAID-DP, which is a double-parity RAID-6 implementation that prevents data loss when two disks fail in the RAID group.

The only aggregates that exist on a newly created cluster are the node root aggregates. The root aggregate should not be used to host user data, so in this section you will create a new aggregate on each of the nodes in cluster1 so they can host any storage virtual machines, volumes, and LUNs that you create later in this lab.

3.1.3.2 Create Aggregate Exercise

In this lab activity, you create a single user data aggregate on each node in the cluster.

1. Display a list of the disks attached to the node cluster-01. (Note that you can omit the **-nodelist** option if you want to display a list of the disks in the entire cluster.)

   Tip: By default the PuTTY window may wrap output lines because the window is too small. If this occurs, simply expand the window by selecting its edge and dragging it wider, after which any subsequent output will utilize the visible width of the window.

```
cluster1::> disk show -nodelist cluster1-01

 Usable           Disk    Container   Container
Disk                   Size Shelf Bay Type    Type        Name      Owner
---------------- ---------- ----- --- ------- ----------- --------- --------

Info: This cluster has partitioned disks. To get a complete list of spare disk capacity use
 "storage aggregate
      show-spare-disks".
VMw-1.25             28.44GB     -   0 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.26             28.44GB     -   1 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.27             28.44GB     -   2 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.28             28.44GB     -   3 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.29             28.44GB     -   4 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.30             28.44GB     -   5 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.31             28.44GB     -   6 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.32             28.44GB     -   8 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.33             28.44GB     -   9 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.34             28.44GB     -  10 SSD     shared      aggr0_cluster1_02 cluster1-02
VMw-1.35             28.44GB     -  11 SSD     shared      -         cluster1-02
VMw-1.36             28.44GB     -  12 SSD     shared      -         cluster1-02
VMw-1.37             28.44GB     -   0 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.38             28.44GB     -   1 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.39             28.44GB     -   2 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.40             28.44GB     -   3 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.41             28.44GB     -   4 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.42             28.44GB     -   5 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.43             28.44GB     -   6 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.44             28.44GB     -   8 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.45             28.44GB     -   9 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.46             28.44GB     -  10 SSD     shared      aggr0_cluster1_01 cluster1-01
VMw-1.47             28.44GB     -  11 SSD     shared      -         cluster1-01
VMw-1.48             28.44GB     -  12 SSD     shared      -         cluster1-01
24 entries were displayed.

cluster1::>
```

2. Display a list of the aggregates on the cluster.

```
cluster1::> aggr show


Aggregate     Size Available Used% State   #Vols  Nodes            RAID Status
--------- -------- --------- ----- ------- ------ ---------------- ------------
aggr0_cluster1_01 24.32GB 1.18GB 95% online     1 cluster1-01      raid_dp,
                                                                   normal
aggr0_cluster1_02 24.32GB 1.18GB 95% online     1 cluster1-02      raid_dp,
                                                                   normal
2 entries were displayed.

cluster1::>
```

3. Create the aggregate named “aggr1_cluster1_01” on the node cluster1-01.

```
cluster1::> aggr create -aggregate aggr1_cluster1_01 -node cluster1-01 -diskcount 5
```

```
Info: The layout for aggregate "aggr1_cluster1_01" on node "cluster1-01" would be:

      First Plex

        RAID Group rg0, 5 disks (block checksum, raid_dp)
                                                            Usable Physical
          Position   Disk                      Type           Size     Size
          ---------- ------------------------- ---------- -------- --------
          shared     VMw-1.37                  SSD               -        -
          shared     VMw-1.38                  SSD               -        -
          shared     VMw-1.39                  SSD         12.39GB  12.42GB
          shared     VMw-1.40                  SSD         12.39GB  12.42GB
          shared     VMw-1.41                  SSD         12.39GB  12.42GB

        RAID Group rg1, 5 disks (block checksum, raid_dp)
                                                            Usable Physical
          Position   Disk                      Type           Size     Size
          ---------- ------------------------- ---------- -------- --------
          shared     VMw-1.37                  SSD               -        -
          shared     VMw-1.38                  SSD               -        -
          shared     VMw-1.39                  SSD         12.39GB  12.42GB
          shared     VMw-1.40                  SSD         12.39GB  12.42GB
          shared     VMw-1.41                  SSD         12.39GB  12.42GB

      Aggregate capacity available for volume use would be 66.90GB.

Do you want to continue? {y|n}: y
[Job 57] Job is queued: Create aggr1_cluster1_01.
[Job 57] Job succeeded: DONE

cluster1::>
```

4. Create the aggregate named “aggr1_cluster1_02” on the node cluster1-02.

```
cluster1::> aggr create -aggregate aggr1_cluster1_02 -node cluster1-02 -diskcount 5

Info: The layout for aggregate "aggr1_cluster1_02" on node "cluster1-02" would be:

      First Plex

        RAID Group rg0, 5 disks (block checksum, raid_dp)
                                                            Usable Physical
          Position   Disk                      Type           Size     Size
          ---------- ------------------------- ---------- -------- --------
          shared     VMw-1.25                  SSD               -        -
          shared     VMw-1.26                  SSD               -        -
          shared     VMw-1.27                  SSD         12.39GB  12.42GB
          shared     VMw-1.28                  SSD         12.39GB  12.42GB
          shared     VMw-1.29                  SSD         12.39GB  12.42GB

        RAID Group rg1, 5 disks (block checksum, raid_dp)
                                                            Usable Physical
          Position   Disk                      Type           Size     Size
          ---------- ------------------------- ---------- -------- --------
          shared     VMw-1.25                  SSD               -        -
          shared     VMw-1.26                  SSD               -        -
          shared     VMw-1.27                  SSD         12.39GB  12.42GB
          shared     VMw-1.28                  SSD         12.39GB  12.42GB
          shared     VMw-1.29                  SSD         12.39GB  12.42GB

      Aggregate capacity available for volume use would be 66.90GB.

Do you want to continue? {y|n}: y
[Job 58] Job is queued: Create aggr1_cluster1_02.
[Job 58] Job succeeded: DONE

cluster1::>
```

5. Display the list of aggregates on the cluster again.

```
cluster1::> aggr show


Aggregate     Size Available Used% State   #Vols  Nodes            RAID Status
--------- -------- --------- ----- ------- ------ ---------------- ------------
aggr0_cluster1_01 24.32GB 1.18GB 95% online     1 cluster1-01      raid_dp,
                                                                   normal
aggr0_cluster1_02 24.32GB 1.18GB 95% online     1 cluster1-02      raid_dp,
```

```
                                                      normal
aggr1_cluster1_01 66.90GB 66.90GB 0% online     0 cluster1-01      raid_dp,
                                                                    normal
aggr1_cluster1_02 66.90GB 66.90GB 0% online     0 cluster1-02      raid_dp,
                                                                    normal
4 entries were displayed.

cluster1::>
```

6. Display all the details for the aggr1_cluster1_01 aggregate.

```
cluster1::> aggr show -aggregate aggr1_cluster1_01 -instance

                                         Aggregate: aggr1_cluster1_01
                                      Storage Type: ssd
                                    Checksum Style: block
                                   Number Of Disks: 5
                                            Mirror: false
                              Disks for First Plex: VMw-1.37, VMw-1.38,
                                                    VMw-1.39, VMw-1.40,
                                                    VMw-1.41
                           Disks for Mirrored Plex: -
                         Partitions for First Plex: -
                      Partitions for Mirrored Plex: -
                                              Node: cluster1-01
                           Free Space Reallocation: off
                                         HA Policy: sfo
                               Ignore Inconsistent: off
                    Space Reserved for Snapshot Copies: -
               Aggregate Nearly Full Threshold Percent: 95%
                      Aggregate Full Threshold Percent: 98%
                             Checksum Verification: on
                                   RAID Lost Write: on
                             Enable Thorough Scrub: off
                                    Hybrid Enabled: false
                                    Available Size: 66.90GB
                                  Checksum Enabled: true
                                   Checksum Status: active
                                           Cluster: cluster1
                                   Home Cluster ID: 8459305e-ac85-11e8-815b-005056b0ac41
                                        DR Home ID: -
                                      DR Home Name: -
                                   Inofile Version: 4
                                  Has Mroot Volume: false
                     Has Partner Node Mroot Volume: false
                                           Home ID: 4053517983
                                         Home Name: cluster1-01
                           Total Hybrid Cache Size: 0B
                                            Hybrid: false
                                      Inconsistent: false
                                 Is Aggregate Home: true
                                     Max RAID Size: 24
       Flash Pool SSD Tier Maximum RAID Group Size: -
                                          Owner ID: 4053517983
                                        Owner Name: cluster1-01
                                   Used Percentage: 0%
                                            Plexes: /aggr1_cluster1_01/plex0
                                       RAID Groups: /aggr1_cluster1_01/plex0/rg0 (block)
                                                    /aggr1_cluster1_01/plex0/rg1 (block)
                             RAID Lost Write State: on
                                       RAID Status: raid_dp, normal
                                         RAID Type: raid_dp
   SyncMirror Resync Snapshot Frequency in Minutes: 5
                                           Is Root: false
      Space Used by Metadata for Volume Efficiency: 0B
                                              Size: 66.90GB
                                             State: online
                        Maximum Write Alloc Blocks: 0
                                         Used Size: 172KB
                                 Uses Shared Disks: true
                                       UUID String: 134e1d09-ecdd-41de-830d-38804b05160d
                                 Number Of Volumes: 0
                             Is Flash Pool Caching: -
            Is Eligible for Auto Balance Aggregate: false
             State of the aggregate being balanced: ineligible
                          Total Physical Used Size: 220KB
                          Physical Used Percentage: 0%
            State Change Counter for Auto Balancer: 0
                                      Is Encrypted: false
                                     SnapLock Type: non-snaplock
```

```
                                 Encryption Key ID: -
 Is in the precommit phase of Copy-Free Transition: false
Is a 7-Mode transitioning aggregate that is not yet committed in clustered Data ONTAP and is
 currently out of space: false
Threshold When Aggregate Is Considered Unbalanced (%): 70
Threshold When Aggregate Is Considered Balanced (%): 40
                        Resynchronization Priority: -
                    Space Saved by Data Compaction: 0B
               Percentage Saved by Data Compaction: 0%
                          Amount of compacted data: 0B
                   Timestamp of Aggregate Creation: 4/9/2019 20:59:43
                                       Enable SIDL: off
                                         Composite: false
                           Capacity Tier Used Size: 0B
                 Space Saved by Storage Efficiency: 0B
   Percentage of Space Saved by Storage Efficiency: 0%
Amount of Shared bytes count by Storage Efficiency: 0B
                   Inactive Data Reporting Enabled: false
                    azcs-read-optimization Enabled: false
                 Enable Aggregate level Encryption: false
                Aggregate uses data protected SEDs: false

cluster1::>
```

3.1.4 Networks

3.1.4.1 Network Concepts

This section discusses the network components that ONTAP provides to manage your cluster.

Ports are the physical Ethernet and Fibre Channel connections on each node, the interface groups (ifgrps) you can create to aggregate those connections, and the VLANs you can use to subdivide them.

A logical interface (LIF) is essentially an IP address that is associated with a port, and has a number of associated characteristics such as an assigned home node, an assigned physical home port, a list of physical ports it can fail over to, an assigned SVM, a role, a routing group, and so on. A given LIF can only be assigned to a single SVM, and since LIFs are mapped to physical network ports on cluster nodes this means that an SVM runs, in part, on all nodes that are hosting its LIFs.

Routing tables in ONTAP are defined for each Storage Virtual Machine. Since each SVM has its own routing table, changes to one SVM's routing table do not have impact on any other SVM's routing table.

IPspaces allow you to configure an ONTAP cluster to logically separate one IP network from another, even if those two networks are using the same IP address range. IPspaces are a multi-tenancy feature that allow storage service providers to share a cluster between different companies while still separating storage traffic for privacy and security. Every cluster includes a default IPspace to which ONTAP automatically assigns new SVMs, and that default IPspace is usually sufficient for NetApp customers who deploy a cluster within a single company or organization that uses a non-conflicting IP address range.

Broadcast Domains are collections of ports that all have access to the same layer 2 networks, both physical and virtual (i.e., VLANs). Every IPspace has its own set of Broadcast Domains, and ONTAP provides a default broadcast domain to go along with the default IPspace. Broadcast domains are used by ONTAP to determine what ports an SVM can use for its LIFs.

Subnets in ONTAP are a convenience feature, intended to make LIF creation and management easier for ONTAP administrators. A subnet is a pool of IP addresses that you can specify by name when creating a LIF. ONTAP will automatically assign an available IP address from the pool to the LIF, along with a subnet mask and a gateway. A subnet is scoped to a specific broadcast domain, so all the subnet's addresses belong to the same layer 3 network. ONTAP manages the pool automatically as you create or delete LIFs, and if you manually configure a LIF with an address from the pool, it will detect that the address is in use and mark it as such in the pool.

DNS Zones allow an SVM to manage DNS name resolution for its own LIFs. Since multiple LIFs can share the same DNS name, this allows the SVM to load balance traffic by IP address across a set of LIFs. To use DNS Zones you must configure your DNS server to delegate DNS authority for the subdomain to the SVM.

3.1.4.2 Create Subnet Exercise

In this lab activity, you use the ONTAP CLI to create a subnet that you will leverage in later exercises while provisioning SVMs and LIFs. You will not create IPspaces or Broadcast Domains, as the system defaults are sufficient for this lab.

1. Display a list of the cluster's IPspaces. A cluster actually contains two IPspaces by default; the “Cluster” IPspace, which correlates to the cluster network that ONTAP uses to have cluster nodes communicate with each other, and the ”Default” IPspace to which ONTAP automatically assigns all new SVMs. You can create more IPspaces if necessary, but that activity is not covered in this lab.

```
cluster1::> network ipspace show
IPspace             Vserver List                  Broadcast Domains
------------------- ----------------------------- ----------------------------
Cluster
                    Cluster                       Cluster
Default
                    cluster1                      Default
2 entries were displayed.

cluster1::>
```

2. Display a list of the cluster's broadcast domains. Remember that broadcast domains are scoped to a single IPspace. The e0a ports on the cluster nodes are part of the “Cluster” broadcast domain in the “Cluster” IPspace. The remaining ports are part of the “Default” broadcast domain in the “Default” IPspace.

```
cluster1::> network port broadcast-domain show
IPspace Broadcast                                         Update
Name    Domain Name    MTU  Port List                     Status Details
------- ----------- ------  ----------------------------- --------------
Cluster Cluster       1500
                            cluster1-01:e0a               complete
                            cluster1-01:e0b               complete
                            cluster1-02:e0a               complete
                            cluster1-02:e0b               complete
Default Default       1500
                            cluster1-01:e0c               complete
                            cluster1-01:e0d               complete
                            cluster1-01:e0e               complete
                            cluster1-01:e0f               complete
                            cluster1-01:e0g               complete
                            cluster1-02:e0c               complete
                            cluster1-02:e0d               complete
                            cluster1-02:e0e               complete
                            cluster1-02:e0f               complete
                            cluster1-02:e0g               complete
2 entries were displayed.

cluster1::>
```

3. Display a list of the cluster's subnets.

```
cluster1::> network subnet show
This table is currently empty.

cluster1::>
```

4. ONTAP does not include a default subnet, so you will need to create a subnet now. The specific command you use depends on which sections of this lab guide you plan to complete, because you need to correctly align the IP address pool in your lab with the IP addresses used in those sections.

   - **If you plan to complete the NAS portion of this lab**, issue the following command. Use this command as well if you plan to complete both the NAS and SAN portions of this lab.

```
cluster1::> network subnet create -subnet-name Demo -broadcast-domain Default
 -ipspace Default -subnet 192.168.0.0/24 -gateway 192.168.0.1
 -ip-ranges 192.168.0.131-192.168.0.139
cluster1::>
```

- **If you only plan to complete the SAN portion of this lab**, then issue the following command instead.

```
cluster1::> network subnet create -subnet-name Demo -broadcast-domain Default
 -ipspace Default -subnet 192.168.0.0/24 -gateway 192.168.0.1
 -ip-ranges 192.168.0.133-192.168.0.139
cluster1::>
```

5. Re-display the list of the cluster's subnets. This example assumes you plan to complete the whole lab.

```
cluster1::> network subnet show

IPspace: Default
Subnet                     Broadcast                    Avail/
Name      Subnet           Domain    Gateway            Total    Ranges
--------- ---------------- --------- --------------- --------- ---------------
Demo      192.168.0.0/24   Default   192.168.0.1        9/9    192.168.0.131-192.168.0.139

cluster1::>
```

6. If you want to see a list of all of the network ports on your cluster, issue the following command.

```
cluster1::> network port show

Node: cluster1-01
                                                  Speed(Mbps) Health
Port      IPspace      Broadcast Domain Link MTU  Admin/Oper  Status
--------- ------------ ---------------- ---- ---- ----------- --------
e0a       Cluster      Cluster          up   1500  auto/1000  healthy
e0b       Cluster      Cluster          up   1500  auto/1000  healthy
e0c       Default      Default          up   1500  auto/1000  healthy
e0d       Default      Default          up   1500  auto/1000  healthy
e0e       Default      Default          up   1500  auto/1000  healthy
e0f       Default      Default          up   1500  auto/1000  healthy
e0g       Default      Default          up   1500  auto/1000  healthy

Node: cluster1-02
                                                  Speed(Mbps) Health
Port      IPspace      Broadcast Domain Link MTU  Admin/Oper  Status
--------- ------------ ---------------- ---- ---- ----------- --------
e0a       Cluster      Cluster          up   1500  auto/1000  healthy
e0b       Cluster      Cluster          up   1500  auto/1000  healthy
e0c       Default      Default          up   1500  auto/1000  healthy
e0d       Default      Default          up   1500  auto/1000  healthy
e0e       Default      Default          up   1500  auto/1000  healthy
e0f       Default      Default          up   1500  auto/1000  healthy
e0g       Default      Default          up   1500  auto/1000  healthy
14 entries were displayed.

cluster1::>
```

3.2 Create Storage for NFS and CIFS

Expected Completion Time: 40 Minutes

You must complete the Clusters exercises before you can proceed.

3.2.1 Create a Storage Virtual Machine for NAS

3.2.1.1 General SVM Concepts

Storage Virtual Machines (SVMs), previously known as Vservers, are the logical storage servers within a cluster that serve data out to storage clients. A single cluster can host hundreds of SVMs, with each SVM managing its own set of volumes (FlexVols), Logical Network Interfaces (LIFs), storage access protocols (e.g., NFS/CIFS/iSCSI/FC/FCoE), and for NAS clients, its own namespace.

A single SVM can host any combination of the supported storage protocols. You explicitly configure which storage protocols you want a given SVM to support at the time you create the SVM. You can later add or remove protocols as desired.

The ability to support many SVMs in a single cluster is a key feature in ONTAP, and NetApp encourages customers to embrace this feature in order to take full advantage of a cluster's capabilities. A cluster that utilizes multiple SVMs will scale signficantly better than a cluster that only has a single SVM, and provides administrators with much greater flexibility when managing workloads and access controls. The number of ONTAP storage controller nodes in a cluster determines the total number of SVMs that can run in the cluster. Each storage controller node can host a maximum of 125 SVMs, so you can calculate the cluster's effective SVM limit by multiplying the number of nodes by 125. As cluster size gets larger, the maximum number of SVMs a cluster can support approaches 1000 SVMs.

An SVM's assigned aggregates and LIFs determine which cluster nodes handle processing for that SVM. As you saw earlier, an aggregate is directly connected to the specific node hosting its disks, which means that an SVM runs in part on any nodes whose aggregates are hosting volumes for the SVM. An SVM also has a direct relationship to any nodes that are hosting its LIFs. LIFs are essentially an IP address with a number of associated characteristics, such as an assigned home node, an assigned physical home port, a list of physical ports it can fail over to, an assigned SVM, a role, a routing group, and so on. You can only assign a given LIF to a single SVM, and since LIFs map to physical network ports on cluster nodes, this means that an SVM runs in part on all the nodes that are hosting its LIFs.

The most efficient client access path to a volume's data is through a LIF that is mapped to a physical network port located on the same node as the aggregate that hosts the volume's storage. However, clients can also access volume data through LIFs bound to physical network ports on other nodes in the cluster; in these cases ONTAP uses the high speed cluster network to bridge communication between the node hosting the LIF and the node hosting the volume.

When you configure an SVM with multiple data LIFs, clients can use any of those LIFs to access volumes hosted by the SVM. The specific LIF IP address (and by extension, which LIF) a client will choose is a function of the storage protocol and of name resolution.

3.2.1.2 Additional SVM Concepts For NAS

For NAS, the mapping of a hostname to an IP address is a function of name resolution services. Under NetBIOS, CIFS Servers have the responsibility for resolving requests from clients for their hostnames, and in so doing can perform some load balancing by responding to different clients with different LIF addresses. However, this distribution is not sophisticated and requires external NetBIOS name servers in order to deal with clients that are not on the local network. NFS Servers do not handle name resolution on their own.

DNS provides basic name resolution load balancing by advertising multiple IP addresses for the same hostname. DNS is supported by both NFS and CIFS clients, and works equally well with clients on local area and wide area networks. Since DNS is an external service that resides outside of ONTAP, this architecture creates the potential for service disruptions if the DNS server is advertising IP addresses for LIFs that are temporarily offline. To compensate for this condition you can configure DNS servers to delegate the name resolution responsibility for the SVM's hostname records to the SVM itself, so that it can directly respond to name resolution requests involving its LIFs. This allows the SVM to consider LIF availability and LIF utilization levels when deciding what LIF address to return in response to a DNS name resolution request.

A NAS LIF (a LIF supporting only NFS and/or CIFS) can automatically failover from one cluster node to another in the event of a component failure. Any existing connections to that LIF from NFS and SMB 2.0 (and later) clients can non-disruptively tolerate the LIF failover event. When a LIF failover happens the NAS LIF migrates to a different physical NIC, potentially to a NIC on a different node in the cluster, and continues servicing network requests from that new node/port. Throughout this operation the NAS LIF maintains its IP address. Clients connected to the LIF may notice a brief delay while the failover is in progress, but as soon as it completes the clients resume any in-process NAS operations without any loss of data.

NetApp best practice is to create at least one NAS LIF for a given SVM on each cluster node that has an aggregate that is hosting volumes for that SVM. If you desire additional resiliency you can also create a NAS LIF on nodes not hosting aggregates for the SVM.

There is no limit on the number of LIFs that an SVM can host, but there is a limit on the number of LIFs that can run on a given node. For low-to-mid range controllers that limit is 256 LIFs per node, and high-end controllers can support 512 LIFs per node. If the node in question is part of an HA pair configured for failover, then this limit is half the usual per-node value so that the node can accommodate its HA partner's LIFs in the event of an HA failover event.

3.2.1.3 Create SVM Exercise

In this section you create a new SVM named “svm1” on the cluster, and configure it to serve out a volume over NFS and CIFS. You will configure two NAS data LIFs on the SVM, one per node in the cluster.

Start by creating the storage virtual machine.

1. If you do not already have a PuTTY connection open to cluster1 then open one now following the directions in the “Accessing the Command Line” section at the beginning of this lab guide. The username is **`admin`**, and the password is **`Netapp1!`**.
2. Create the SVM named “svm1”.

   Tip: The ONTAP command line syntax refers to storage virtual machines as vservers.

```
cluster1::> vserver create -vserver svm1 -rootvolume svm1_root -aggregate aggr1_cluster1_01
 -language C.UTF-8 -rootvolume-security ntfs -snapshot-policy default
[Job 39] Job is queued: Create svm1.
[Job 39]
[Job 39] Job succeeded:
Vserver creation completed

cluster1::>
```

3. Display the list of protocols the SVM supports by default.

```
cluster1::> vserver show-protocols -vserver svm1

  Vserver: svm1
Protocols: nfs, cifs, fcp, iscsi, ndmp

cluster1::>
```

4. Remove the FCP, iSCSI, and NDMP protocols from the SVM svm1, leaving only CIFS and NFS.

```
cluster1::> vserver remove-protocols -vserver svm1 -protocols fcp,iscsi,ndmp
cluster1::>
```

5. Display the list of protocols assigned to the SVM “svm1”.

```
cluster1::> vserver show-protocols -vserver svm1

  Vserver: svm1
Protocols: nfs, cifs

cluster1::>
```

6. Display a list of the vservers in the cluster.

```
cluster1::> vserver show
                               Admin      Operational Root
Vserver     Type    Subtype    State      State       Volume     Aggregate
----------- ------- ---------- ---------- ----------- ---------- ----------
cluster1    admin   -          -          -           -          -
cluster1-01 node    -          -          -           -          -
cluster1-02 node    -          -          -           -          -
svm1        data    default    running    running     svm1_root  aggr1_
                                                                 cluster1_
                                                                 01
4 entries were displayed.

cluster1::>
```

7. Display a list of the cluster's network interfaces.

```
cluster1::> network interface show
            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
```

```
Cluster
            cluster1-01_clus1
                         up/up    169.254.172.65/16  cluster1-01   e0a     true
            cluster1-01_clus2
                         up/up    169.254.109.18/16  cluster1-01   e0b     true
            cluster1-02_clus1
                         up/up    169.254.82.149/16  cluster1-02   e0a     true
            cluster1-02_clus2
                         up/up    169.254.126.160/16 cluster1-02   e0b     true
cluster1
            cluster1-01_mgmt1
                         up/up    192.168.0.111/24   cluster1-01   e0c     true
            cluster1-02_mgmt1
                         up/up    192.168.0.112/24   cluster1-02   e0c     true
            cluster_mgmt up/up    192.168.0.101/24   cluster1-01   e0c     true
7 entries were displayed.

cluster1::>
```

8. Notice that there are not any LIFs defined for the SVM “svm1” yet. Create the “svm1_cifs_nfs_lif1” data LIF for svm1.

```
cluster1::> network interface create -vserver svm1 -lif svm1_cifs_nfs_lif1 -role data
 -data-protocol nfs,cifs -home-node cluster1-01 -home-port e0c -subnet-name Demo
 -firewall-policy mgmt-nfs
cluster1::>
```

9. Create the “svm1_cifs_nfs_lif2” data LIF for the SVM svm1.

```
cluster1::> network interface create -vserver svm1 -lif svm1_cifs_nfs_lif2 -role data
 -data-protocol nfs,cifs -home-node cluster1-02 -home-port e0c -subnet-name Demo
 -firewall-policy mgmt-nfs
cluster1::>
```

10. Display all of the LIFs owned by svm1.

```
cluster1::> network interface show -vserver svm1
            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
svm1
            svm1_cifs_nfs_lif1
                         up/up    192.168.0.131/24   cluster1-01   e0c     true
            svm1_cifs_nfs_lif2
                         up/up    192.168.0.132/24   cluster1-02   e0c     true
2 entries were displayed.

cluster1::>
```

11. Display the SVM svm1's DNS configuration.

```
cluster1::> vserver services dns show
                                                              Name
Vserver         State     Domains                             Servers
--------------- --------- ----------------------------------- ----------------
cluster1        enabled   demo.netapp.com                     192.168.0.253
cluster1::>
```

There is currently only a DNS server defined for cluster1.

12. Configure the DNS domain and nameservers for the svm1 SVM.

```
cluster1::> vserver services dns create -vserver svm1 -name-servers 192.168.0.253
 -domains demo.netapp.com

Warning: Only one DNS server is configured. Configure more than one DNS server to avoid a
         single-point-of-failure.

cluster1::>
```

This lab only contains a single DNS server, so ignore the warning message.

13. Display the DNS configuration for all SVMs.

```
cluster1::> vserver services dns show
                                                              Name
```

```
Vserver         State     Domains                             Servers
--------------- --------- ----------------------------------- ----------------
cluster1        enabled   demo.netapp.com                     192.168.0.253
svm1            enabled   demo.netapp.com                     192.168.0.253
2 entries were displayed.
cluster1::>
```

Configure the LIFs to accept DNS delegation responsibility for the svm1.demo.netapp.com zone so that you can advertise addresses for both of the NAS data LIFs that belong to svm1. You could have done this as part of the `network interface create` commands, but perform it separately here so you can see how to modify an existing LIF.

14. Configure lif1 to accept DNS delegation responsibility for the svm1.demo.netapp.com zone.

```
cluster1::> network interface modify -vserver svm1 -lif svm1_cifs_nfs_lif1
 -dns-zone svm1.demo.netapp.com
cluster1::>
```

15. Configure lif2 to accept DNS delegation responsibility for the svm1.demo.netapp.com zone.

```
cluster1::> network interface modify -vserver svm1 -lif svm1_cifs_nfs_lif2
 -dns-zone svm1.demo.netapp.com
cluster1::>
```

16. Display the DNS delegation for svm1.

```
cluster1::> network interface show -vserver svm1 -fields dns-zone,address
vserver lif                 address       dns-zone
------- ------------------ ------------- --------------------
svm1    svm1_cifs_nfs_lif1 192.168.0.131 svm1.demo.netapp.com
svm1    svm1_cifs_nfs_lif2 192.168.0.132 svm1.demo.netapp.com
2 entries were displayed.
cluster1::>
```

17. To verify that DNS delegation is working correctly, open a PuTTY connection to the Linux host rhel1 (username **`root`** and password **`Netapp1!`**) and execute the following commands. If the delegation is working correctly you should see IP addresses returned for the host svm1.demo.netapp.com, and if you run the command several times you will eventually see that the responses vary the returned address between the SVM's two LIFs.

```
[root@rhel1 ~]# nslookup svm1.demo.netapp.com
Server:  192.168.0.253
Address:      192.168.0.253#53
Non-authoritative answer:
Name: svm1.demo.netapp.com
Address: 192.168.0.132
[root@rhel1 ~]# nslookup svm1.demo.netapp.com
Server:  192.168.0.253
Address:      192.168.0.253#53
Non-authoritative answer:
Name: svm1.demo.netapp.com
Address: 192.168.0.131
[root@rhel1 ~]#
```

18. This completes the planned LIF configuration changes for svm1, so now display a detailed configuration report for the LIF svm1_cifs_nfs_lif1.

```
cluster1::> network interface show -lif svm1_cifs_nfs_lif1 -instance

                    Vserver Name: svm1
          Logical Interface Name: svm1_cifs_nfs_lif1
                  Service Policy: custom-data-58975
                    Service List: data-core, data-nfs, data-cifs,
                                  management-ssh, management-https
               (DEPRECATED)-Role: data
                   Data Protocol: nfs, cifs
                 Network Address: 192.168.0.131
                         Netmask: 255.255.255.0
             Bits in the Netmask: 24
                      Is VIP LIF: false
                     Subnet Name: Demo
                       Home Node: cluster1-01
                       Home Port: e0c
                    Current Node: cluster1-01
```

```
                   Current Port: e0c
             Operational Status: up
                Extended Status: -
                        Is Home: true
          Administrative Status: up
                Failover Policy: system-defined
                Firewall Policy: mgmt-nfs
                    Auto Revert: false
  Fully Qualified DNS Zone Name: svm1.demo.netapp.com
        DNS Query Listen Enable: true
            Failover Group Name: Default
                       FCP WWPN: -
                 Address family: ipv4
                        Comment: -
                 IPspace of LIF: Default
 Is Dynamic DNS Update Enabled?: true
       Probe-port for Azure ILB: -
               Broadcast Domain: Default

cluster1::>
```

When you issued the `vserver create` command to create svm1 you included an option to enable CIFS, but that command did not actually create a CIFS server for the SVM. Now it is time to create that CIFS server.

19. Display the status of the cluster's CIFS servers.

```
cluster1::> vserver cifs show
This table is currently empty.
cluster1::>
```

20. Create a CIFS server for svm1.

```
cluster1::> vserver cifs create -vserver svm1 -cifs-server svm1 -domain demo.netapp.com
In order to create an Active Directory machine account for the CIFS server, you
must supply the name and password of a Windows account with sufficient
privileges to add computers to the "CN=Computers" container within the
"DEMO.NETAPP.COM" domain.

Enter the user name: Administrator

Enter the password: Netapp1!

Info: SMB1 protocol version is disabled on this CIFS server. If required, use the (privilege:
 advanced)
command "vserver cifs options modify -vserver svm1 -smb1-enabled true" to enable it.
cluster1::>
```

21. Display the status of the cluster's CIFS servers.

```
cluster1::> vserver cifs show
            Server          Status    Domain/Workgroup Authentication
Vserver     Name            Admin     Name             Style
----------- --------------- --------- ---------------- --------------
svm1        SVM1            up        DEMO             domain
cluster1::>
```

As with CIFS, when you created svm1 you included an option to enable NFS, but that command did not actually create the NFS server. Now it is time to create that NFS server.

22. Display the status of the NFS server for svm1.

```
cluster1::> vserver nfs status -vserver svm1
The NFS server is not running on Vserver "svm1".
cluster1::>
```

23. Create an NFS v3 NFS server for svm1.

```
cluster1::> vserver nfs create -vserver svm1 -v3 enabled -access true
cluster1::>
```

24. Display the status of the NFS server for svm1 again.

```
cluster1::> vserver nfs status -vserver svm1
The NFS server is running on Vserver "svm1".
```

```
cluster1::>
```

3.2.2 Configure CIFS and NFS

3.2.2.1 CIFS and NFS Concepts

ONTAP configures CIFS and NFS on a per SVM basis. When you created the “svm1” SVM in the previous section, you set up and enabled CIFS and NFS for that SVM. However, clients cannot yet access the SVM using CIFS and NFS. That is partially because you have not yet created any volumes on the SVM, but also because you have not told the SVM what to share, and who to share it with.

Each SVM has its own NAS namespace, a logical grouping of the SVM's CIFS and NFS volumes into a single logical filesystem view. Clients can access the entire namespace by mounting a single share or NFS export that references the top of the namespace tree. This means that SVM administrators can maintain a single, centrally managed, consistent, real-time view of the SVM's data for all clients rather than having to reproduce that view structure on each individual client. As an administrator maps and unmaps volumes from the namespace, those volumes instantly become visible or disappear from clients that have mounted CIFS and NFS volumes higher in the SVM's namespace. Administrators can also create NFS exports at individual junction points within the namespace, and can create CIFS shares at any directory path in the namespace.

The root of an SVM’s namespace is hosted on the SVM's root volume (“svm1_root” in the case of the “svm1” SVM). The SVM's other volumes are junctioned (i.e., mounted) within that root volume, or within other volumes that are already junctioned into the namespace. A key point to remember is that CIFS and NFS clients cannot access a volume that has not been junctioned into the namespace.

When you create an SVM, ONTAP automatically creates a root volume to hold that SVM's namespace. An SVM always has a root volume, whether or not it is configured to support NAS protocols.

Once CIFS is enabled for an SVM, clients still cannot use SMB to access data on your SVM until you create a share. Share creation on ONTAP works very similarly to how it works on Windows; you specify the directory you want to share, name the share, and assign share permissions. As you saw in the previous exercise, ONTAP supports Active Directory integration, so you can leverage your existing Active Directory groups and users when configuring share access rules. Although NetApp recommends using Active Directory integration for your CIFS SVMs (as that provides the best user experience), you can also set up CIFS services without using Active Directory.

CIFS and NFS clients can access the entire namespace by mounting a single NFS export or CIFS share that references the top of the root volume directory hierarchy. While this is a very powerful capability, there is no requirement to make the whole namespace accessible. You can create CIFS shares at any directory level in the namespace, and you can create different NFS export rules at junction boundaries for individual volumes, and for individual qtrees within a junctioned volume.

ONTAP does not utilize an /etc/exports file to export NFS volumes; instead it uses a policy model that dictates the NFS client access rules for the associated volumes. An NFS-enabled SVM implicitly exports the root of its namespace and automatically associates that export with the SVM's default export policy, but that default policy is initially empty, and until it is populated with access rules no NFS clients is able to access the namespace.

The SVM's default export policy applies to the root volume, and also to any volumes that an administrator junctions into the namespace, but an administrator can optionally create additional export policies in order to implement different access rules within the namespace. You can apply export policies to a volume as a whole and to individual qtrees within a volume, but a given volume or qtree can only have one associated export policy. While you cannot create NFS exports at any other directory level in the namespace, NFS clients can mount from any level in the namespace by leveraging the namespace's root export.

3.2.2.2 Configure CIFS and NFS Exercise

In this activity, you configure a default export policy for your SVM so that any volumes you junction into its namespace will automatically pick up the same NFS export rules. You will also create a single CIFS share at the top of the namespace so that all the volumes you junction into that namespace are accessible through that one share.

1. Before you configure NFS and CIFS for your newly created SVM, take a quick look at svm1's root volume.

```
cluster1::> vol show -vserver svm1
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
svm1      svm1_root    aggr1_cluster1_01
                                    online     RW         20MB    18.66MB    1%

cluster1::>
```

The root volume hosts the namespace for the SVM. Root volumes are small because they are only intended to house the junctions that organize the SVM's volumes. All of the files hosted on the SVM should reside inside other volumes that are junctioned into the namespace, rather than directly in the SVM's root volume.

2. Verify that CIFS is running by default for the SVM svm1.

```
cluster1::> vserver cifs show
            Server          Status    Domain/Workgroup Authentication
Vserver     Name            Admin     Name             Style
----------- --------------- --------- ---------------- --------------
svm1        SVM1            up        DEMO             domain

cluster1::>
```

3. Display the status of the NFS server for svm1 again.

```
cluster1::> vserver nfs status -vserver svm1
The NFS server is running on Vserver "svm1".

cluster1::>
```

4. Display the NFS server's configuration.

```
cluster1::> vserver nfs show -vserver svm1

                                        Vserver: svm1
                             General NFS Access: true
                                         NFS v3: enabled
                                       NFS v4.0: disabled
                                   UDP Protocol: enabled
                                   TCP Protocol: enabled
                           Default Windows User: -
                            NFSv4.0 ACL Support: disabled
                NFSv4.0 Read Delegation Support: disabled
               NFSv4.0 Write Delegation Support: disabled
                        NFSv4 ID Mapping Domain: defaultv4iddomain.com
            NFSv4 Grace Timeout Value (in secs): 45
Preserves and Modifies NFSv4 ACL (and NTFS File Permissions in Unified Security Style):
 enabled
                  NFSv4.1 Minor Version Support: disabled
                                 Rquota Enable: disabled
                   NFSv4.1 Parallel NFS Support: enabled
                            NFSv4.1 ACL Support: disabled
                           NFS vStorage Support: disabled
            NFSv4 Support for Numeric Owner IDs: enabled
                          Default Windows Group: -
                NFSv4.1 Read Delegation Support: disabled
               NFSv4.1 Write Delegation Support: disabled
                            NFS Mount Root Only: enabled
                                  NFS Root Only: disabled
            Permitted Kerberos Encryption Types: des, des3, aes-128, aes-256
                               Showmount Enabled: enabled
Set the Protocol Used for Name Services Lookups for Exports: udp
                    NFSv3 MS-DOS Client Support: disabled
     Idle Connection Timeout Value (in seconds): 360
             Are Idle NFS Connections Supported: disabled
Hide Snapshot Directory under NFSv3 Mount Point: disabled
           Provide Root Path as Showmount State: disabled

cluster1::>
```

5. Display a list of all the export policies.

```
cluster1::> vserver export-policy show
Vserver          Policy Name
---------------  -------------------
svm1             default

cluster1::>
```

The only defined policy is “default”.

6. Display a list of all the export policy rules.

```
cluster1::> vserver export-policy rule show
This table is currently empty.

cluster1::>
```

There are no rules defined for the “default” export policy.

7. Add a rule to the default export policy granting read-write access to all hosts over CIFS and NFS.

```
cluster1::> vserver export-policy rule create -vserver svm1 -policyname default
 -clientmatch 0.0.0.0/0 -protocol cifs,nfs -rorule any -rwrule any -superuser any
 -anon 65534 -ruleindex 1

cluster1::>
```

8. Display a list of all the export policy rules.

```
cluster1::> vserver export-policy rule show
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
svm1         default         1       cifs,    0.0.0.0/0             any
                                     nfs

cluster1::>
```

9. Display a detailed list of all the export policy rules.

```
cluster1::> vserver export-policy rule show -policyname default -instance

                                    Vserver: svm1
                                Policy Name: default
                                 Rule Index: 1
                            Access Protocol: cifs, nfs
List of Client Match Hostnames, IP Addresses, Netgroups, or Domains: 0.0.0.0/0
                             RO Access Rule: any
                             RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                   Superuser Security Types: any
               Honor SetUID Bits in SETATTR: true
                  Allow Creation of Devices: true

cluster1::>
```

10. Display a list of the shares in the cluster.

```
cluster1::> vserver cifs share show
Vserver        Share         Path              Properties Comment  ACL
-------------- ------------- ----------------- ---------- -------- -----------
svm1           admin$        /                 browsable  -        -
svm1           c$            /                 oplocks    -        BUILTIN\Administrators /
 Full Control
                                               browsable
                                               changenotify
                                               show-previous-versions
svm1           ipc$          /                 browsable  -        -
3 entries were displayed.

cluster1::>
```

11. Create a share at the root of the namespace for the SVM svm1.

```
cluster1::> vserver cifs share create -vserver svm1 -share-name nsroot -path /
```

```
cluster1::>
```

12. Display a list of the shares in the cluster again.

```
cluster1::> vserver cifs share show
Vserver        Share         Path              Properties Comment  ACL
-------------- ------------- ----------------- ---------- -------- -----------
svm1           admin$        /                 browsable  -        -
svm1           c$            /                 oplocks    -        BUILTIN\Administrators /
 Full Control
                                               browsable
                                               changenotify
                                               show-previous-versions
svm1           ipc$          /                 browsable  -        -
svm1           nsroot        /                 oplocks    -        Everyone / Full Control
                                               browsable
                                               changenotify
                                               show-previous-versions
4 entries were displayed.

cluster1::>
```

3.2.2.3 Username Mapping Exercise

In this activity, you set up name mapping between UNIX and Windows user accounts to facilitate smooth multiprotocol access to the volumes and files in the namespace.

1. Display a list of the current name mappings.

```
cluster1::> vserver name-mapping show
This table is currently empty.
cluster1::>
```

2. Create a name mapping of DEMO\Administrator (specified in the command as “demo\\administrator”) to root.

```
cluster1::> vserver name-mapping create -vserver svm1 -direction win-unix -position 1
 -pattern demo\\administrator -replacement root
cluster1::>
```

3. Create a name mapping of root to DEMO\Administrator.

```
cluster1::> vserver name-mapping create -vserver svm1 -direction unix-win -position 1
 -pattern root -replacement demo\\administrator
cluster1::>
```

4. Display a list of the current name mappings.

```
cluster1::> vserver name-mapping show

Vserver:   svm1
Direction: win-unix
Position Hostname         IP Address/Mask
-------- ---------------- ----------------
1        -                -                   Pattern: demo\\administrator
                                          Replacement: root

Vserver:   svm1
Direction: unix-win
Position Hostname         IP Address/Mask
-------- ---------------- ----------------
1        -                -                   Pattern: root
                                          Replacement: demo\\administrator
2 entries were displayed.

cluster1::>
```

3.2.3 Create a Volume and Map It to the Namespace

The Configure CIFS and NFS section discussed the concepts related to namespaces, so that information will not be repeated here.

3.2.3.1 Volume Concepts

Volumes, or FlexVols, are the dynamically sized containers used by ONTAP to store data. A volume only resides in a single aggregate at a time, but any given aggregate can host multiple volumes. Unlike an aggregate, which can associate with multiple SVMs, a volume can only associate to a single SVM. The maximum size of a volume can vary depending on what storage controller model is hosting it.

An SVM can host multiple volumes. While there is no specific limit on the number of FlexVols that can be configured for a given SVM, each storage controller node is limited to hosting no more than 1000 FlexVols, and there are higher cluster-wide maximums as well. Depending on the number of nodes in your cluster, the controller models, and the protocol mix (NAS vs. SAN) of the volumes, a given cluster will have a supported maximum of 1000 to 12,000 FlexVols.

Each storage controller node has a root aggregate (e.g., aggr0_<nodename>) that contains the node's ONTAP operating system.

Important: Do not use the node's root aggregate to host any other volumes or user data; always create additional aggregates and volumes for that purpose.

ONTAP FlexVols support a number of storage efficiency features including thin provisioning, deduplication, and compression. One specific storage efficiency feature you will see in this section of the lab is thin provisioning, which dictates how space for a FlexVol is allocated in its containing aggregate.

When you create a FlexVol with a volume guarantee of type “volume” you are thickly provisioning the volume, pre-allocating all of the space for the volume on the containing aggregate, which ensures that the volume will never run out of space unless the volume reaches 100% capacity.

When you create a FlexVol with a volume guarantee of “none” you are thinly provisioning the volume, which directs ONTAP to only allocate space in the aggregate for the volume as data is added to the volume. Thin provisioning allows you to increase your overall space utilization, and even oversubscribe an aggregate by allocating more volumes on it than the aggregate could actually accommodate if all those volumes reached their full size at the same time. If an oversubscribed aggregate does fill up, then all of its volumes will run out of space before they reach their maximum volume size, likely resulting in a service disruption. If you choose to oversubscribe an aggregate you should regularly monitor aggregate and volume space utilization so you can head off unanticipated growth before it might cause a problem.

3.2.3.2 Create and Map a Volume Exercise

In the Clusters section, you created a new aggregate named “aggr1_cluster1_01”; you will now use that aggregate to host a new thinly provisioned volume named “engineering” for the SVM named “svm1”.

1. Display basic information about the SVM's current list of volumes.

```
cluster1::> volume show -vserver svm1
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
svm1      svm1_root    aggr1_cluster1_01
                                    online     RW         20MB    18.57MB    2%

cluster1::>
```

2. Display the junctions in the SVM's namespace.

```
cluster1::> volume show -vserver svm1 -junction
                                Junction                            Junction
Vserver   Volume       Language Active    Junction Path             Path Source
--------- ------------ -------- --------  ------------------------- -----------
svm1      svm1_root    C.UTF-8  true      /                         -

cluster1::>
```

3. Create the volume “engineering”, junctioning it into the namespace at “/engineering”.

```
cluster1::> volume create -vserver svm1 -volume engineering -aggregate aggr1_cluster1_01 -size
 10GB -percent-snapshot-space 5 -space-guarantee none -tiering-policy none -policy default -
junction-path /engineering
[Job 83] Job is queued: Create engineering.
```

```
[Job 83] Job succeeded: Successful

cluster1::>
```

4. Display a list of svm1's volumes.

```
cluster1::> volume show -vserver svm1
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
svm1      engineering  aggr1_cluster1_01
                                    online     RW         10GB     9.50GB    0%
svm1      svm1_root    aggr1_cluster1_01
                                    online     RW         20MB    18.51MB    2%
2 entries were displayed.

cluster1::>
```

5. Display a list of svm1's volume junction points.

```
cluster1::> volume show -vserver svm1 -junction
                                Junction                            Junction
Vserver   Volume       Language Active    Junction Path             Path Source
--------- ------------ -------- --------  ------------------------- -----------
svm1      engineering  C.UTF-8  true      /engineering              RW_volume
svm1      svm1_root    C.UTF-8  true      /                         -
2 entries were displayed.

cluster1::>
```

6. Create the volume “eng_users”, junctioning it into the namespace at “/engineering/users”.

```
cluster1::> volume create -vserver svm1 -volume eng_users -aggregate aggr1_cluster1_01 -size
 10GB -percent-snapshot-space 5 -space-guarantee none -tiering-policy none -policy default -
junction-path /engineering/users
[Job 85] Job is queued: Create eng_users.
[Job 85] Job succeeded: Successful

cluster1::>
```

7. Display a list of svm1's volume junction points.

```
cluster1::> volume show -vserver svm1 -junction
                                Junction                            Junction
Vserver   Volume       Language Active    Junction Path             Path Source
--------- ------------ -------- --------  ------------------------- -----------
svm1      eng_users    C.UTF-8  true      /engineering/users        RW_volume
svm1      engineering  C.UTF-8  true      /engineering              RW_volume
svm1      svm1_root    C.UTF-8  true      /                         -
3 entries were displayed.

cluster1::>
```

8. Display detailed information about the volume engineering. Notice that the volume is reporting as thin provisioned (Space Guarantee Style is set to “none”), and that the Export Policy is set to “default”.

```
cluster1::> volume show -vserver svm1 -volume engineering -instance

                                   Vserver Name: svm1
                                    Volume Name: engineering
                                 Aggregate Name: aggr1_cluster1_01
  List of Aggregates for FlexGroup Constituents: aggr1_cluster1_01
                                    Volume Size: 10GB
                             Volume Data Set ID: 1028
                      Volume Master Data Set ID: 2147551010
                                   Volume State: online
                                   Volume Style: flex
                          Extended Volume Style: flexvol
                         Is Cluster-Mode Volume: true
                          Is Constituent Volume: false
                                  Export Policy: default
                                        User ID: -
                                       Group ID: -
                                 Security Style: ntfs
                               UNIX Permissions: ------------
                                  Junction Path: /engineering
                           Junction Path Source: RW_volume
                                Junction Active: true
```

```
                     Junction Parent Volume: svm1_root
                                    Comment:
                             Available Size: 9.50GB
                            Filesystem Size: 10GB
                    Total User-Visible Size: 9.50GB
                                  Used Size: 300KB
                            Used Percentage: 0%
       Volume Nearly Full Threshold Percent: 95%
              Volume Full Threshold Percent: 98%
                           Maximum Autosize: 12GB
                           Minimum Autosize: 10GB
         Autosize Grow Threshold Percentage: 85%
       Autosize Shrink Threshold Percentage: 50%
                              Autosize Mode: off
        Total Files (for user-visible data): 311287
         Files Used (for user-visible data): 98
                  Space Guarantee in Effect: true
                        Space SLO in Effect: true
                                  Space SLO: none
                      Space Guarantee Style: none
                         Fractional Reserve: 0%
                                Volume Type: RW
          Snapshot Directory Access Enabled: true
         Space Reserved for Snapshot Copies: 5%
                      Snapshot Reserve Used: 0%
                            Snapshot Policy: default
                              Creation Time: Tue Apr 09 22:02:44 2019
                                   Language: C.UTF-8
                               Clone Volume: false
                                  Node name: cluster1-01
                  Clone Parent Vserver Name: -
                    FlexClone Parent Volume: -
                              NVFAIL Option: off
                      Volume's NVFAIL State: false
    Force NVFAIL on MetroCluster Switchover: off
                  Is File System Size Fixed: false
                 (DEPRECATED)-Extent Option: off
              Reserved Space for Overwrites: 0B
          Primary Space Management Strategy: volume_grow
                   Read Reallocation Option: off
Naming Scheme for Automatic Snapshot Copies: create_time
           Inconsistency in the File System: false
               Is Volume Quiesced (On-Disk): false
             Is Volume Quiesced (In-Memory): false
   Volume Contains Shared or Compressed Data: true
          Space Saved by Storage Efficiency: 0B
     Percentage Saved by Storage Efficiency: 0%
               Space Saved by Deduplication: 0B
          Percentage Saved by Deduplication: 0%
              Space Shared by Deduplication: 0B
                 Space Saved by Compression: 0B
      Percentage Space Saved by Compression: 0%
        Volume Size Used by Snapshot Copies: 0B
                                 Block Type: 64-bit
                           Is Volume Moving: false
             Flash Pool Caching Eligibility: read-write
  Flash Pool Write Caching Ineligibility Reason: -
                  Managed By Storage Service: -
                        Storage Service Name: -
Create Namespace Mirror Constituents For SnapDiff Use: -
      Action When Attributes Are Not Reachable: -
                    Constituent Volume Role: -
                        Directory Bit Width: -
          Maximum Size of Namespace Constituent: -
          Maximum Size of Each Data Constituent: -
                      QoS Policy Group Name: -
             QoS Adaptive Policy Group Name: -
                        Caching Policy Name: -
              Is Volume Move in Cutover Phase: false
         Number of Snapshot Copies in the Volume: 0
VBN_BAD may be present in the active filesystem: false
              Is Volume on a hybrid aggregate: false
                   Total Physical Used Size: 300KB
                   Physical Used Percentage: 0%
                              List of Nodes: cluster1-01
                      Is Volume a FlexGroup: false
                              SnapLock Type: non-snaplock
                      Vserver DR Protection: -
                          Enable Encryption: false
                        Is Volume Encrypted: false
```

```
                          Volume Encryption State: none
                                Encryption Key ID:
                                      Application: -
                                  Is SIDL enabled: off
                            Over Provisioned Size: 0B
                  Available Snapshot Reserve Size: 512MB
                                Logical Used Size: 300KB
                          Logical Used Percentage: 0%
                           Logical Available Size: -
           Logical Size Used by Active Filesystem: 300KB
               Logical Size Used by All Snapshots: 0B
                          Logical Space Reporting: false
                            Volume Tiering Policy: none
              Performance Tier Inactive User Data: -
      Performance Tier Inactive User Data Percent: -

cluster1::>
```

9. View how much disk space this volume is actually consuming in its containing aggregate. The "Total Footprint" value represents the volume's total consumption. The value here is so small because this volume is thin provisioned, and you have not yet added any data to it. If you had thick provisioned the volume, then the footprint here would have been 1 GB, the full size of the volume.

```
cluster1::> volume show-footprint -volume engineering

      Vserver : svm1
      Volume  : engineering

      Feature                                         Used    Used%
      --------------------------------          ----------    -----
      Volume Data Footprint                          300KB       0%
      Volume Guarantee                                  0B       0%
      Flexible Volume Metadata                     13.38MB       0%
      Delayed Frees                                  816KB       0%

      Total Footprint                              14.47MB       0%


cluster1::>
```

3.2.4 Connect NAS Clients

The "svm1" SVM is up and running and is configured for NFS and CIFS access, so it's time to validate that everything is working properly by mounting the NFS export on a Linux host, and the CIFS share on a Windows host. You should complete the exercises for both clients so you can see that they are both able to seamlessly access the volume and its files.

3.2.4.1 Windows Client

This lab activity demonstrates how to connect the Windows client Jumphost to the CIFS share "\\svm1\nsroot" using the Windows GUI.

1. On the Windows host Jumphost, open Windows Explorer by clicking on the **folder icon** on the task bar.

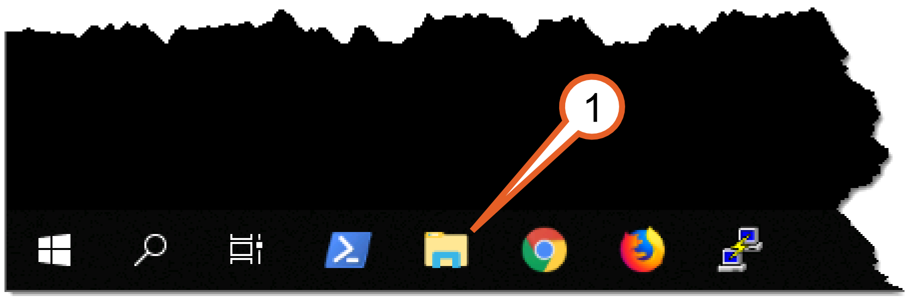

Figure 3-1:

A Windows Explorer window opens.

2. In the left pane of Windows Explorer click on **This PC**.
3. On the menu bar, click on **Computer**.
4. Click on **Map network drive** to launch the Map Network Drive wizard.

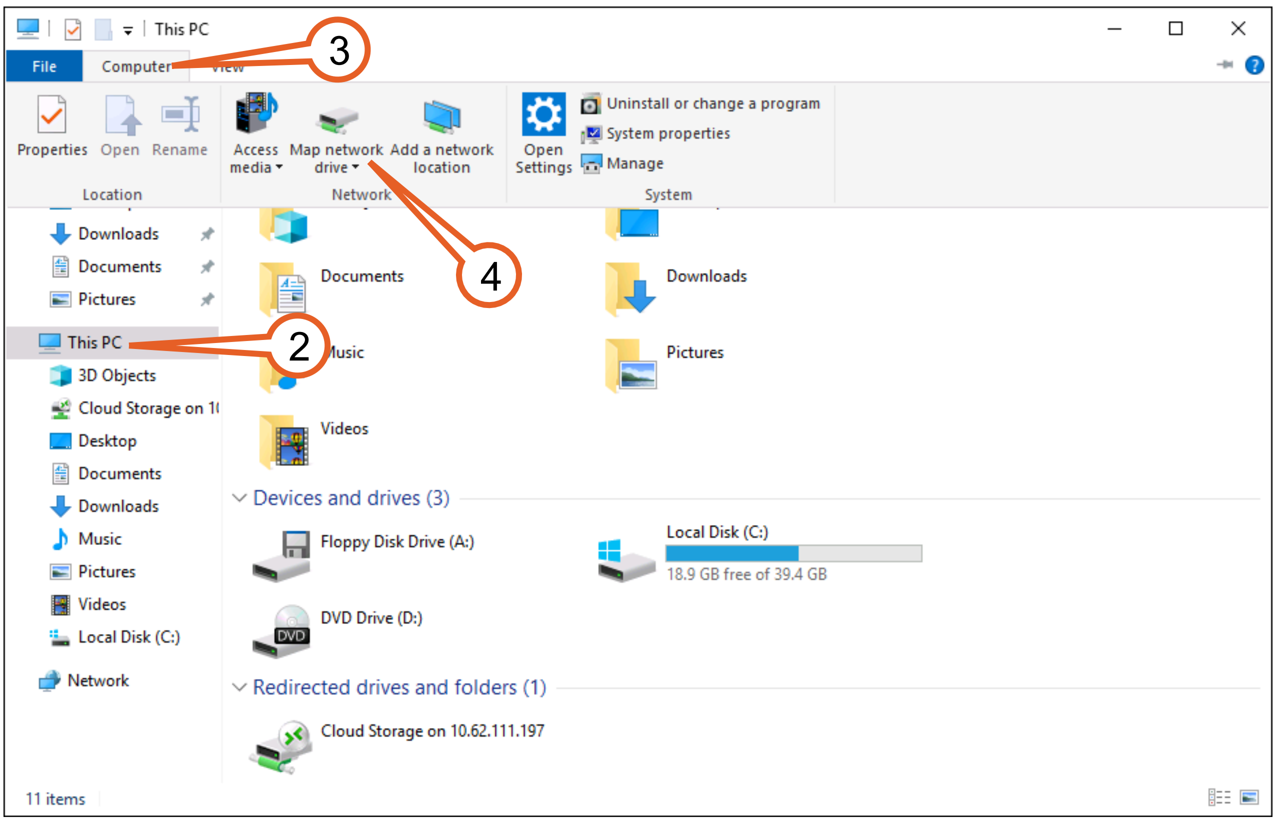

Figure 3-2:

The “Map Network Drive” wizard opens.

5. Set the fields in the window to the following values.
   - “Drive”: **S:**.
   - “Folder”: **\\svm1\nsroot**.
   - Check the **Reconnect at sign-in** check box.

6. When finished click **Finish**.

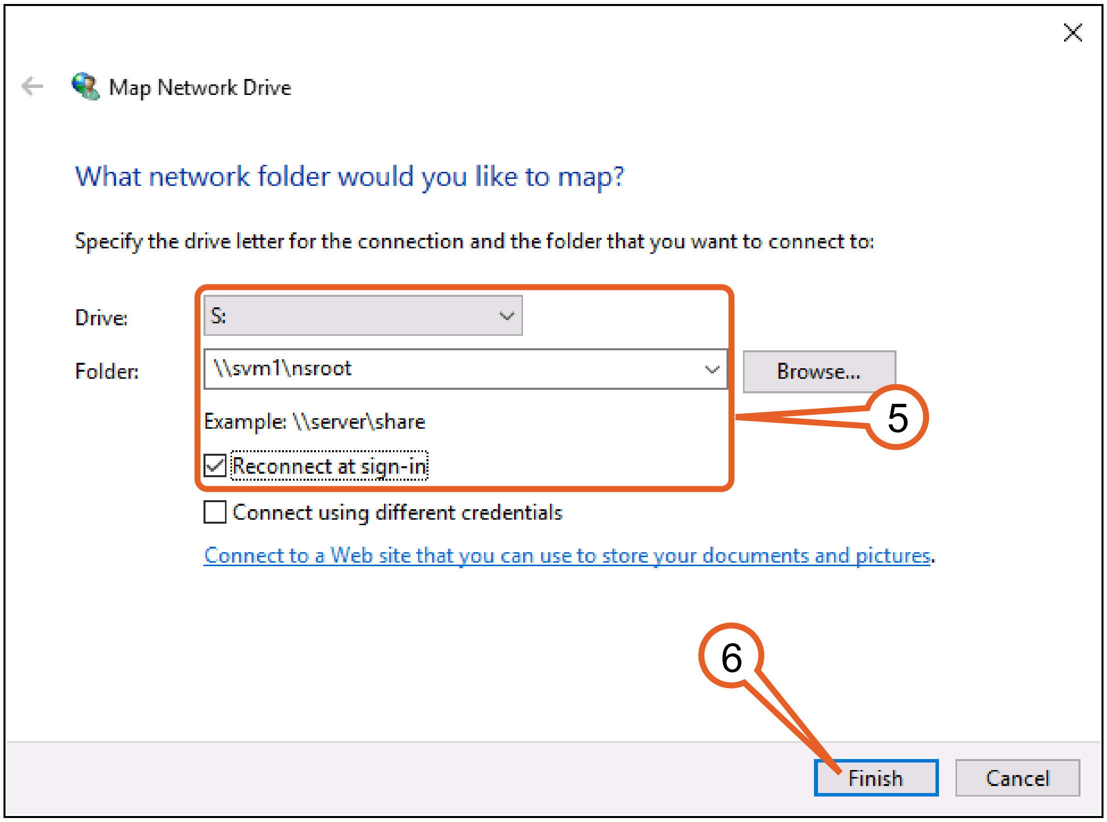

Figure 3-3:

A new Windows Explorer window opens.

7. The engineering volume you previously junctioned into the svm1's namespace is visible at the top of the nsroot share that points to the root of the namespace. If you created another volume on svm1 right now and mounted it under the root of the namespace, that new volume would instantly become visible

in this share, and to clients like Jumphost that have already mounted the share. Double-click on the **engineering** folder to open it.

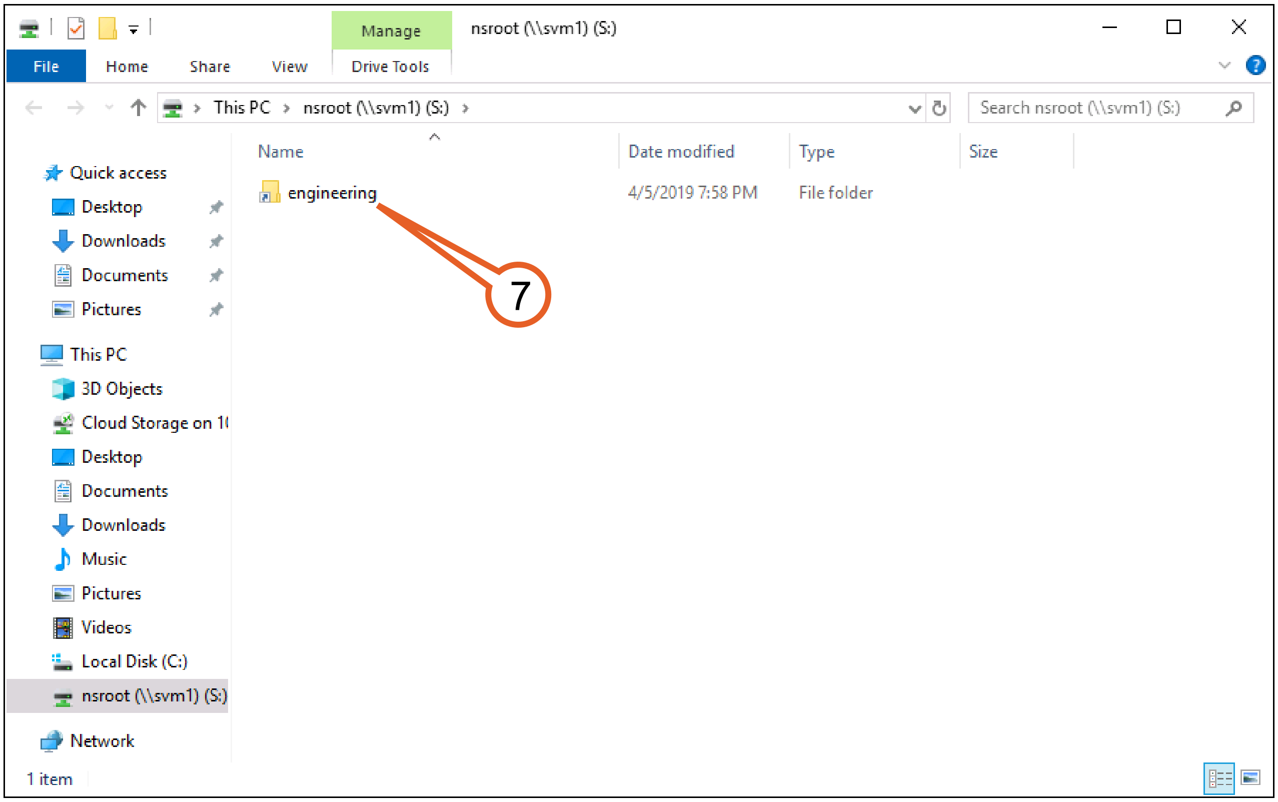

Figure 3-4:

File Explorer displays the contents of the engineering folder. Now create a file in this folder to confirm that you can write to it.

8. Notice that the “eng_users” volume that you junctioned in as “users” is visible inside this folder.

9. Right-click in the empty space in the right pane of File Explorer, and select **New** > **Text Document** from the context menu.

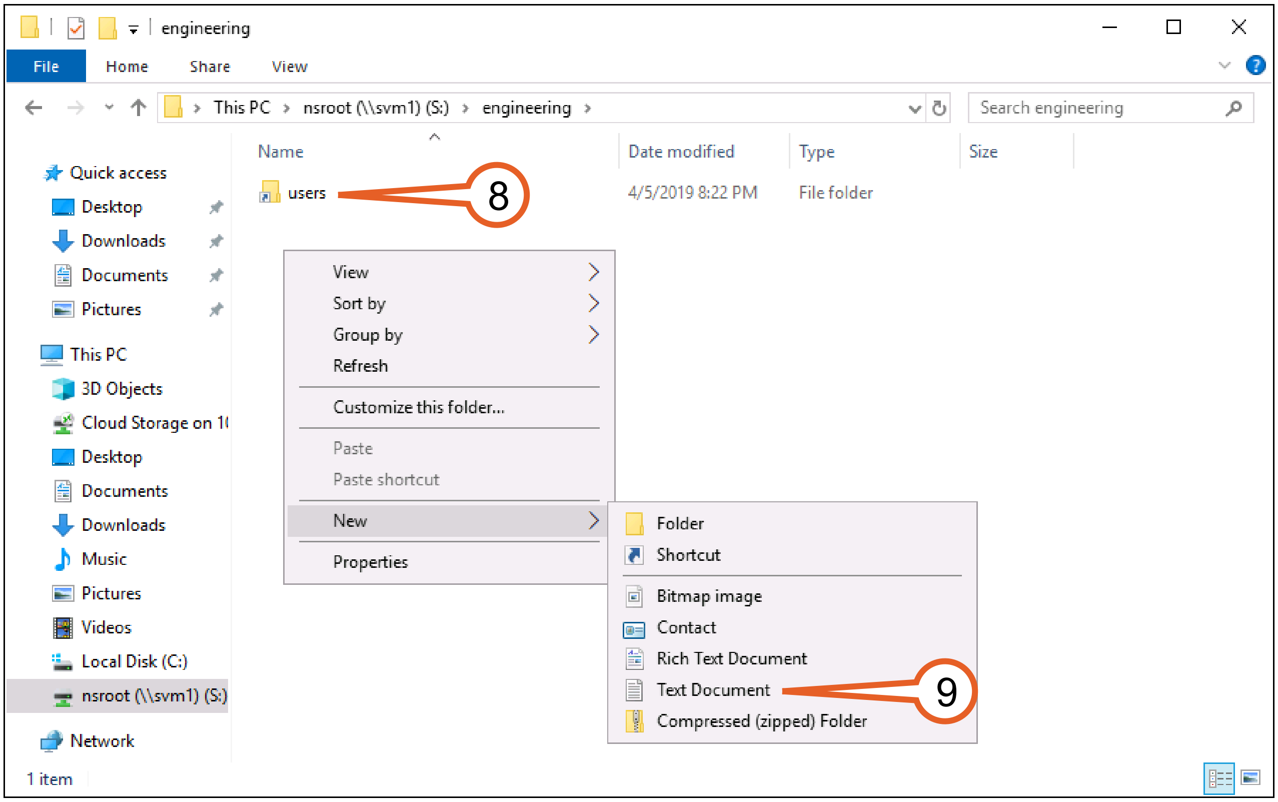

Figure 3-5:

10. Name the resulting file **`cifs.txt`**.
11. Double-click the **cifs.txt** file to open it with Notepad.

Tip: If you do not see file extensions in your lab, you can enable them by going to the **View** menu at the top of Windows Explorer and checking the **File Name Extensions** check box.

12. In Notepad, enter some text. Ensure that you put a carriage return at the end of the line, otherwise when you later view the contents of this file on Linux, the command shell prompt will appear on the same line as the file contents.
13. Use the **File** > **Save** menu in Notepad to save the file's updated contents to the share. If write access is working properly then the save operation will complete silently (i.e., you will not receive an error message).
14. Close the Notepad window.

15. Close the File Explorer window.

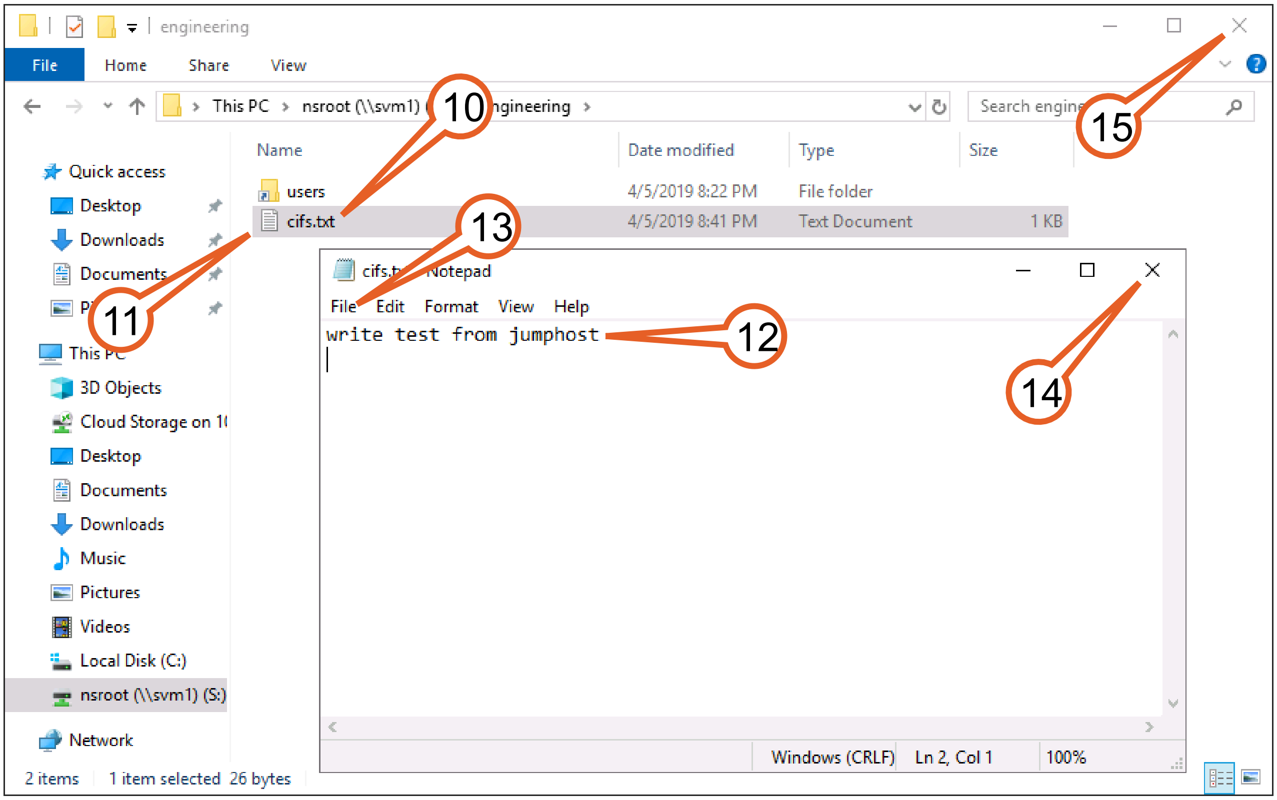

Figure 3-6:

3.2.4.2 Linux Client

This section demonstrates how to connect a Linux client to the NFS volume svm1:/ using the Linux command line.

1. Follow the instructions in the “Accessing the Command Line” section at the beginning of this lab guide to open PuTTY and connect to the system **rhel1**. Log in as the user **`root`** with the password **`Netapp1!`**.
2. Verify that there are no NFS volumes currently mounted on rhel1.

```
[root@rhel1 ~]# df
Filesystem             1K-blocks    Used Available Use% Mounted on
/dev/mapper/rhel-root   39265556 2810364  36455192   8% /
devtmpfs                  930956       0    930956   0% /dev
tmpfs                     941996       0    941996   0% /dev/shm
tmpfs                     941996   98820    843176  11% /run
tmpfs                     941996       0    941996   0% /sys/fs/cgroup
/dev/sda1                 508588  129304    379284  26% /boot
tmpfs                     188400       0    188400   0% /run/user/0
[root@rhel1 ~]#
```

3. Create the /svm1 directory to serve as a mount point for the NFS volume you will be mounting.

```
[root@rhel1 ~]# mkdir /svm1
[root@rhel1 ~]#
```

4. Add an entry for the NFS mount to the fstab file.

```
[root@rhel1 ~]# echo "svm1:/ /svm1 nfs rw,defaults 0 0" >> /etc/fstab
[root@rhel1 ~]#
```

5. Verify the fstab file contains the new entry you just created.

```
[root@rhel1 ~]# grep svm1 /etc/fstab
svm1:/ /svm1 nfs rw,defaults 0 0
```

```
[root@rhel1 ~]#
```

6. Mount all the file systems listed in the fstab file.

```
[root@rhel1 ~]# mount -a
[root@rhel1 ~]#
```

7. View a list of the mounted file systems.

```
[root@rhel1 ~]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/mapper/rhel-root  39265556 2810304  36455252   8% /
devtmpfs                 930956       0    930956   0% /dev
tmpfs                    941996       0    941996   0% /dev/shm
tmpfs                    941996   98828    843168  11% /run
tmpfs                    941996       0    941996   0% /sys/fs/cgroup
/dev/sda1                508588  129304    379284  26% /boot
tmpfs                    188400       0    188400   0% /run/user/0
svm1:/                    19456     576     18880   3% /svm1
[root@rhel1 ~]#
```

The NFS file system svm1:/ now shows as mounted on /svm1.

8. Navigate into the /svm1 directory.

```
[root@rhel1 ~]# cd /svm1
[root@rhel1 svm1]#
```

9. Notice that you can see the engineering volume that you previously junctioned into the SVM's namespace.

```
[root@rhel1 svm1]# ls
engineering
[root@rhel1 svm1]#
```

10. Navigate into engineering and list its contents.

    Attention: The following command output assumes that you have already performed the Windows client connection steps found earlier in this lab guide, including creating the cifs.txt file.

```
[root@rhel1 svm1]# cd engineering
[root@rhel1 engineering]# ls
cifs.txt  users
[root@rhel1 engineering]#
```

11. Display the contents of the cifs.txt file you created earlier.

    Tip: When you `cat` the cifs.txt file, if the shell prompt winds up on the same line as the file output, that indicates that you forgot to include a new line at the end of the file when you created the file on Windows.

```
[root@rhel1 engineering]# cat cifs.txt
write test from jumphost
[root@rhel1 engineering]#
```

12. Verify that you can create file in this directory.

```
[root@rhel1 engineering]# echo "write test from rhel1" > nfs.txt
[root@rhel1 engineering]# cat nfs.txt
write test from rhel1
[root@rhel1 engineering]# ls -l
total 4
-rwxrwxrwx 1 root bin    26 Sep 22 16:15 cifs.txt
-rwxrwxrwx 1 root root   22 Sep 22 19:59 nfs.txt
drwxrwxrwx 4 root root 4096 Sep 22 04:54 users
[root@rhel1 engineering]#
```

3.2.5 Creating and NFS Exporting Qtrees (Optional)

This is an optional exercise for customers with very large NFS environments who need more NFS exports than can be supported by volumes alone in ONTAP.

3.2.5.1 Qtree Concepts

Qtrees had many capabilities in Data ONTAP 7-mode that are no longer present in cluster mode. Qtrees do still exist, but their purpose is now mostly limited to quota management, with most other 7-mode qtree features, including NFS exports, now taken over by volumes. This functionality change created challenges for 7-mode customers with large numbers of NFS qtree exports who were trying to transition to cluster mode and could not convert those qtrees to volumes because they would exceed ONTAP's maximum limit for the number of supported volumes.

To address this problem, ONTAP introduced qtree NFS. NetApp continues to recommend that customers favor volumes over qtrees in cluster mode whenever practical, but customers who require a larger numbers of exports than can be supported by volumes alone can now use qtree NFS to meet those needs.

3.2.5.2 Creating Qtrees Exercise

This optional exercise explains how to create qtrees. You will create a couple of qtrees within the “eng_users” volume, one for each of the users “bob” and “susan”.

1. Create a qtree in the eng_users volume named “bob”.

```
cluster1::> volume qtree create -vserver svm1 -volume eng_users -qtree bob
cluster1::>
```

2. Create a qtree in the eng_users volume named “susan”.

```
cluster1::> volume qtree create -vserver svm1 -volume eng_users -qtree susan
cluster1::>
```

3. Generate a list of all the qtrees that belong to svm1.

```
cluster1::> volume qtree show -vserver svm1
Vserver    Volume        Qtree        Style        Oplocks   Status
---------- ------------- ------------ ------------ --------- --------
svm1       eng_users     ""           ntfs         enable    normal
svm1       eng_users     bob          ntfs         enable    normal
svm1       eng_users     susan        ntfs         enable    normal
svm1       engineering   ""           ntfs         enable    normal
svm1       svm1_root     ""           ntfs         enable    normal
5 entries were displayed.

cluster1::>
```

4. Produce a detailed report of the configuration for the qtree “bob”.

```
cluster1::> volume qtree show -qtree bob -instance

                      Vserver Name: svm1
                       Volume Name: eng_users
                        Qtree Name: bob
  Actual (Non-Junction) Qtree Path: /vol/eng_users/bob
                    Security Style: ntfs
                       Oplock Mode: enable
                  Unix Permissions: -
                          Qtree Id: 1
                      Qtree Status: normal
                     Export Policy: default
        Is Export Policy Inherited: true

cluster1::>
```

3.2.5.3 NFS Exporting Qtrees Exercise

This optional exercise explains how to configure qtree exports and create different export rules for a given qtree. You will use the qtrees you created in the preceding exercise, so you must complete that exercise before you proceed.

Note: While this section provides a graphical method to configure qtree NFS exports, you must still use the command line to accomplish some configuration tasks.

In this exercise you create a new export policy and export rules that permit NFS access from only the Linux host rhel1.

1. Display a list of the export policies.

```
cluster1::> vserver export-policy show
Vserver          Policy Name
---------------  -------------------
svm1             default
cluster1::>
```

2. Create the export policy named “rhel1-only”.

```
cluster1::> vserver export-policy create -vserver svm1 -policyname rhel1-only
cluster1::>
```

3. Re-display the list of export policies.

```
cluster1::> vserver export-policy show
Vserver          Policy Name
---------------  -------------------
svm1             default
svm1             rhel1-only
2 entries were displayed.

cluster1::>
```

4. Display a list of the rules for the “rhel1-only” export policy.

```
cluster1::> vserver export-policy rule show -vserver svm1 -policyname rhel1-only
There are no entries matching your query.
cluster1::>
```

5. Add a rule to the policy so that only the Linux host rhel1 is granted access.

```
cluster1::> vserver export-policy rule create -vserver svm1 -policyname rhel1-only
 -clientmatch 192.168.0.61 -rorule any -rwrule any -superuser any -anon 65534
 -ruleindex 1

cluster1::>
```

6. Display a list of all the export policy rules.

```
cluster1::> vserver export-policy rule show
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
svm1         default         1       cifs,    0.0.0.0/0             any
                                     nfs
svm1         rhel1-only      1       any      192.168.0.61          any
2 entries were displayed.

cluster1::>
```

7. Display a detailed report of the rhel1-only export policy rules.

```
cluster1::> vserver export-policy rule show -vserver svm1
-policyname rhel1-only -instance

                                    Vserver: svm1
                                Policy Name: rhel1-only
                                 Rule Index: 1
                            Access Protocol: any
List of Client Match Hostnames, IP Addresses, Netgroups, or Domains: 192.168.0.61
                             RO Access Rule: any
                             RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                   Superuser Security Types: any
               Honor SetUID Bits in SETATTR: true
                  Allow Creation of Devices: true

cluster1::>
```

8. Produce a list of svm1's export policies.

```
cluster1::> vserver export-policy show
Vserver          Policy Name
---------------  -------------------
svm1             default
svm1             rhel1-only
2 entries were displayed.

cluster1::>
```

9. List svm1's qtrees.

```
cluster1::> volume qtree show
Vserver    Volume       Qtree        Style        Oplocks   Status
---------- ------------ ------------ ------------ --------- --------
svm1       eng_users    ""           ntfs         enable    normal
svm1       eng_users    bob          ntfs         enable    normal
svm1       eng_users    susan        ntfs         enable    normal
svm1       engineering  ""           ntfs         enable    normal
svm1       svm1_root    ""           ntfs         enable    normal
5 entries were displayed.

cluster1::>
```

10. Apply the rhel1-only export policy to the “susan” qtree.

```
cluster1::> volume qtree modify -vserver svm1 -volume eng_users -qtree susan
 -export-policy rhel1-only
Info: The newly configured qtree export policies may not be enforced on existing NFS mount
 points.
cluster1::>
```

11. Display the configuration of the “susan” qtree. Notice the Export Policy field shows that this qtree is using the “rhel1-only” export policy.

```
cluster1::> volume qtree show -vserver svm1 -volume eng_users -qtree susan

                      Vserver Name: svm1
                       Volume Name: eng_users
                        Qtree Name: susan
  Actual (Non-Junction) Qtree Path: /vol/eng_users/susan
                    Security Style: ntfs
                       Oplock Mode: enable
                  Unix Permissions: -
                          Qtree Id: 2
                      Qtree Status: normal
                     Export Policy: rhel1-only
        Is Export Policy Inherited: false

cluster1::>
```

12. Produce a report showing the export policy assignments for all the volumes and qtrees that belong to svm1.

```
cluster1::> volume qtree show -vserver svm1 -fields export-policy
vserver volume    qtree export-policy
------- --------- ----- -------------
svm1    eng_users ""    default
svm1    eng_users bob   default
svm1    eng_users susan rhel1-only
svm1    engineering
                  ""    default
svm1    svm1_root ""    default
5 entries were displayed.

cluster1::>
```

Now you need to validate that the more restrictive export policy that you applied to the qtree “susan” is working as expected from rhel1.

Note: If you still have an active PuTTY session open to the Linux host rhel1 then bring that window up now, otherwise open a new PuTTY session to that host (username **`root`**, password **`Netapp1!`**).

13. Change directory to /svm1/engineering/users.

```
[root@rhel1 ~]# cd /svm1/engineering/users
[root@rhel1 users]#
```

14. List the directory contents.

```
[root@rhel1 users]# ls
bob susan
[root@rhel1 users]#
```

15. Enter the “susan” sub-directory.

```
[root@rhel1 users]# cd susan
[root@rhel1 susan]#
```

16. Create a file in this directory.

```
[root@rhel1 susan]# echo "hello from rhel1" > rhel1.txt
[root@rhel1 susan]#
```

17. Display the contents of the newly created file.

```
[root@rhel1 susan]# cat rhel1.txt
hello from rhel1
[root@rhel1 susan]#
```

Validate that rhel2 has different access rights to the qtree. This host should be able to access all the volumes and qtrees in the svm1 namespace *except* “susan”, which should give a permission denied error because that qtree's associated export policy only grants access to the host rhel1.

Note: Open a PuTTY connection to the Linux host rhel2 (again, username is **`root`** and password is **`Netapp1!`**).

18. Create a mount point for the svm1 NFS volume.

```
[root@rhel2 ~]# mkdir /svm1
[root@rhel2 ~]#
```

19. Mount the NFS volume svm1:/ on /svm1.

```
[root@rhel2 ~]# mount svm1:/ /svm1
[root@rhel2 ~]#
```

20. Change directory to /svm1/engineering/users.

```
[root@rhel2 ~]# cd /svm1/engineering/users
[root@rhel2 users]#
```

21. List the directory's contents.

```
[root@rhel2 users]# ls
bob susan
[root@rhel2 users]#
```

22. Attempt to enter the “susan” sub-directory.

```
[root@rhel2 users]# cd susan
bash: cd: susan: Permission denied
[root@rhel2 users]#
```

23. Attempt to enter the “bob” sub-directory.

```
[root@rhel2 users]# cd bob
[root@rhel2 bob]
```

4 References

The following references were used in writing this lab guide.

- TR-3982: NetApp Clustered Data ONTAP 8.3.X and 8.2.X – an Introduction, November 2015
- TR-4100: Nondisruptive Operations with SMB File Shares ONTAP 9.x, November 2016
- TR-4129: Namespaces in clustered Data ONTAP, July 2014
- TR-4523: DNS Load Balancing in ONTAP - Configuration and Best Practices, May 2020
- NetApp Hardware Universe: https://hwu.netapp.com/Home/Index, NetApp Support site login required.

5 Version History

Version	Date	Document Version History
Version 1.0	October 2014	Initial Release for Hands On Labs
Version 1.0.1	December 2014	Updates for Lab on Demand
Version 1.1	April 2015	Updated for Data ONTAP 8.3GA and other application software. NDO section spun out into a separate lab guide.
Version 1.2	October 2015	Updated for Data ONTAP 8.3.1GA and other application software.
Version 1.3	September 2016	Updated for ONTAP 9.0RC1 and other application software.
Version 1.3 Rev 1	November 2016	Updated for ONTAP 9.0P1, various errata.
Version 2.0	October 2018	Updated for ONTAP 9.4, created separate NAS and SAN lab guides.
Version 2.1	April 2019	Updated for ONTAP 9.5
Version 2.2	December 2020	Updated for ONTAP 9.7

Legal Notices

Note: This document was created using machine translation. If you have any issues with this translated content please consult the English version.

Legal notices provide access to copyright statements, trademarks, patents, and more.

- **Copyright**

  http://www.netapp.com/us/legal/copyright.aspx
- **Trademarks**

  NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

  http://www.netapp.com/us/legal/netapptmlist.aspx
- **Patents**

  A current list of NetApp owned patents can be found at

  https://www.netapp.com/us/media/patents-page.pdf
- **Privacy Policy**

  https://www.netapp.com/us/legal/privacypolicy/index.aspx
- **Open Source**

  Provides information about third-party copyright and licenses used in this product.

  - Notice for ONTAP 9.10.1
  - Notice for ONTAP 9.9.1
  - Notice for ONTAP 9.8
  - Notice for ONTAP 9.7
  - Notice for ONTAP 9.6
  - Notice for ONTAP 9.5
  - Notice for ONTAP 9.4
  - Notice for ONTAP 9.3
  - Notice for ONTAP 9.2
  - Notice for ONTAP 9.1

NetApp®