Basic Concepts for NetApp ONTAP 9.7

SAN Services, CLI Edition

April 2023 | LD00378 Version 2.2.0

TABLE OF CONTENTS

1 Introduction

This lab introduces the fundamentals of NetApp ONTAP®. Using a pre-created 2-node cluster, you will provision storage for access through iSCSI, and configure Windows 2019 and Red Hat Enterprise Linux 7.5 clients to access storage on the cluster over iSCSI.

There is a similar lab available that demonstrates how to configure these same clients to access storage on the cluster using NAS protocols instead, specifically using CIFS and NFS. The NAS and SAN lab guides are designed so they can both be completed within the same running lab instance. 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 you will see 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 SAN protocols (iSCSI).

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 iSCSI (Optional, ECT = 90 minutes including all optional subsections)
  - Create a Storage Virtual Machine.
  - For Windows (Optional, ECT = 40 minutes).
    - Create a Windows LUN and map the LUN to an igroup.
    - Configure a Windows client for iSCSI and MPIO and mount the LUN.
  - For Linux (Optional, ECT = 40 minutes).
    - Create a Linux LUN and map the LUN to an igroup.
    - Configure a Linux client for iSCSI and multipath and mount the LUN.

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.

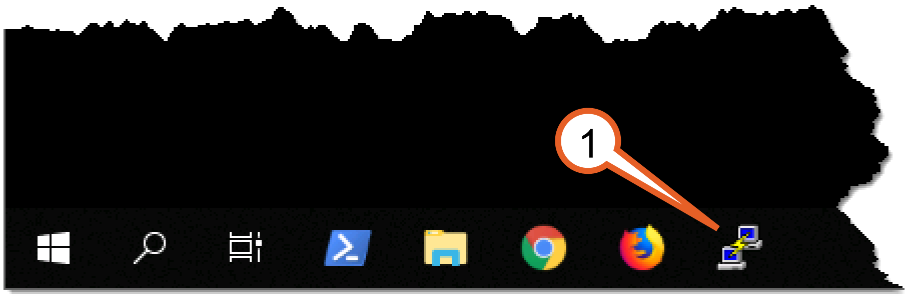


Figure 1-1:

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.

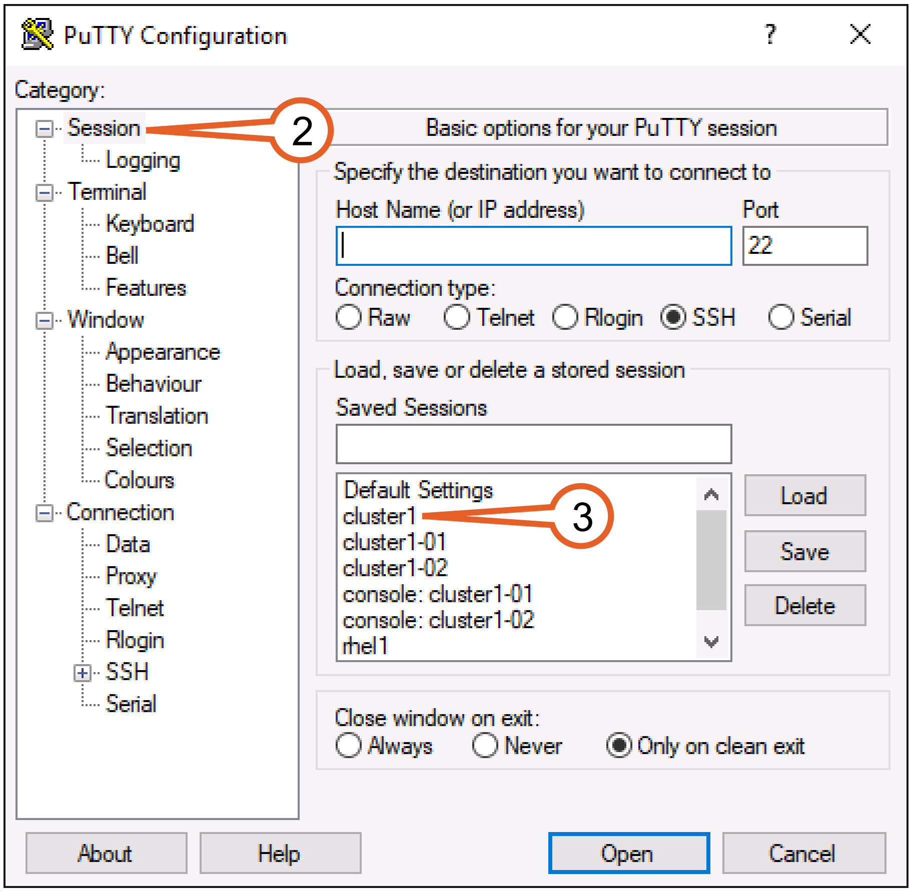


Figure 1-2:

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.

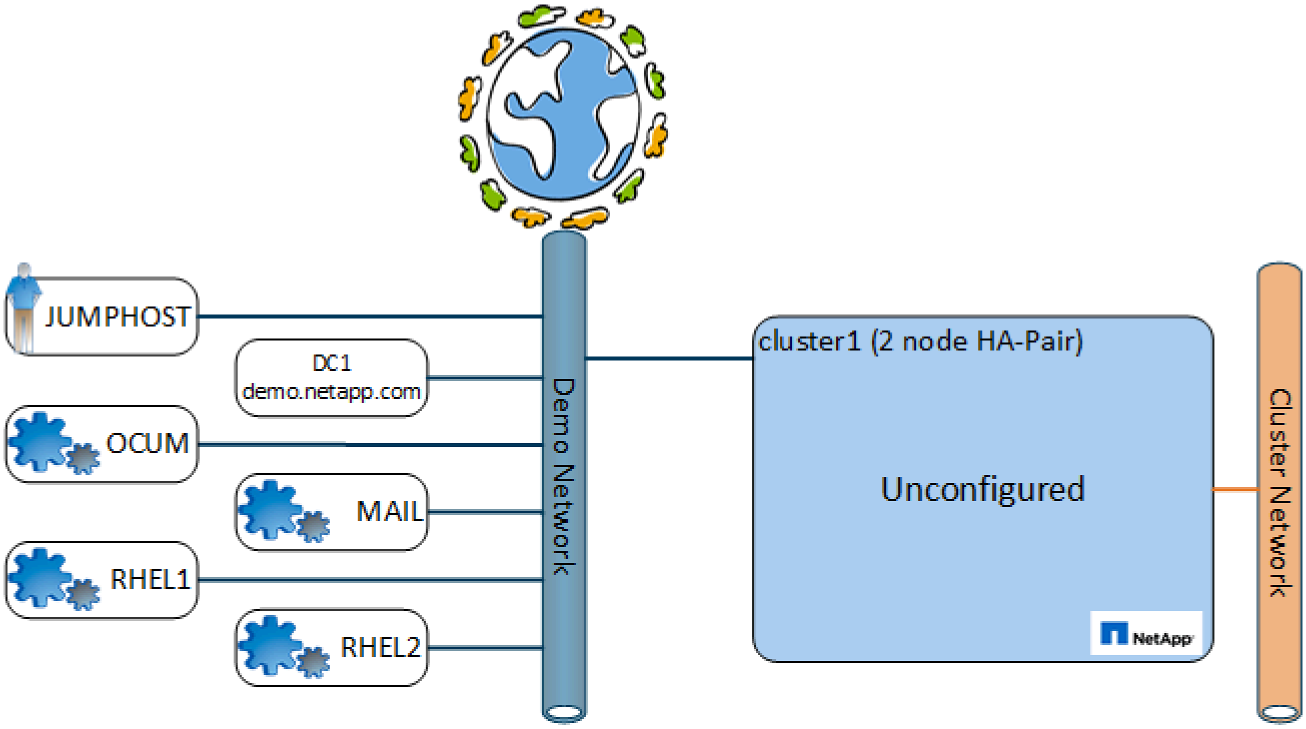


Figure 2-1:

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 iSCSI

Expected Completion Time: 50 Minutes

The 50 minute time estimate assumes you complete only one of the Windows or Linux “Create, Map, and Mount a LUN” sections. You are welcome to complete both of those section if you choose, but you should plan on needing approximately 90 minutes to complete the entire exercise.

In this section you are going to create a new SVM and configure it for iSCSI. The configuration steps for iSCSI and FC are similar, so the information provided here is very applicable for FC deployments too. After you create the SVM and configure it for iSCSI, you will create a LUN for Windows and/or a LUN for Linux, and then mount the LUN(s) on their respective hosts.

In this lab the cluster contains two nodes connected to a single storage network. You will still configure a total of 4 SAN LIFs, because it is common to see implementations with 2 paths per cluster node for redundancy.

3.2.1 Create a Storage Virtual Machine for iSCSI

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 SAN-Specific SVM Concepts

NetApp supports configuring an SVM to serve data over both SAN and NAS protocols, but it is common to see customers use separate SVMs for each in order to separate administrative responsibilities, or for architectural and operational clarity. For example, SAN protocols do not support LIF failover, so you cannot use NAS LIFs to support SAN protocols. You must instead create dedicated LIFs just for SAN. Implementing separate SVMs for SAN and NAS can, in this example, simplify the operational complexity of each SVM's configuration, making each easier to understand and manage. But ultimately, whether to mix or separate is a customer decision and not a NetApp recommendation.

Since SAN LIFs do not support migration to different nodes, an SVM must have dedicated SAN LIFs on every node that you want to service SAN requests, and you must utilize MPIO and ALUA to manage the controller's available paths to the LUNs. In the event of a path disruption, MPIO and ALUA will compensated by re-routing the LUN communication over an alternate controller path (i.e., over a different SAN LIF).

NetApp best practice is to configure at least one SAN LIF per storage fabric/network on each node in the cluster so that all nodes can provide a path to the LUNs. In large clusters where this would result in the presentation of a large number of paths for a given LUN we recommend that you use portsets to limit the LUN to seeing no more than 8 LIFs.

ONTAP also supports Selective LUN Mapping (SLM) to provide further assistance in managing fabric paths. SLM limits LUN path access to just the node that owns the LUN and its HA partner. ONTAP automatically applies SLM to all new LUN map operations.

3.2.1.3 Create an SVM for iSCSI Exercise

In this exercise you will create a new SVM named "svmluns" on the cluster. You will then configure it for iSCSI, and create four data LIFs to support LUN access to the SVM (two on each cluster node).

If you do not already have a PuTTY session open to cluster1, open one now following the instructions in the "Accessing the Command Line" section at the beginning of this lab guide and enter the following commands.

1. Display the available aggregates so you can decide which one you want to use to host the root volume for the SVM you are creating.

```
cluster1::> aggr show

Aggregate     Size Available Used% State   #Vols  Nodes            RAID Status
--------- -------- --------- ----- ------- ------ ---------------- ------------
aggr0_cluster1_01 97.28GB 52.21GB 46% online    1 cluster1-01      raid_dp,
                                                                   normal
aggr0_cluster1_02 97.28GB 52.21GB 46% online    1 cluster1-02      raid_dp,
                                                                   normal
aggr1_cluster1_01 38.18GB 38.12GB 0% online     3 cluster1-01      raid_dp,
                                                                   normal
aggr1_cluster1_02 38.18GB 38.18GB 0% online     0 cluster1-02      raid_dp,
                                                                   normal
4 entries were displayed.

cluster1::>
```

2. Create the SVM svmluns on aggregate aggr1_cluster1_01. Note that the ONTAP command line syntax still refers to storage virtual machines as vservers.

```
cluster1::> vserver create -vserver svmluns -rootvolume svmluns_root
 -aggregate aggr1_cluster1_01 -language C.UTF-8 -rootvolume-security-style unix
 -snapshot-policy default
[Job 52] Job is queued: Create svmluns.
[Job 52] Job succeeded:
Vserver creation completed

cluster1::>
```

3. Add the iSCSI protocol to the SVM "svmluns".

```
cluster1::> vserver iscsi create -vserver svmluns
cluster1::>
```

4. Display svmlun's configured protocols.

```
cluster1::> vserver show-protocols -vserver svmluns
  Vserver: svmluns
Protocols: nfs, cifs, fcp, iscsi, ndmp
cluster1::>
```

5. Remove all the protocols other than iscsi.

```
cluster1::> vserver remove-protocols -vserver svmluns -protocols nfs,cifs,fcp,ndmp
cluster1::>
```

6. Display the configured protocols for svmluns.

```
cluster1::> vserver show-protocols -vserver svmluns
  Vserver: svmluns
Protocols: iscsi
cluster1::>
```

7. Display detailed configuration for the svmlun SVM.

```
cluster1::> vserver show -vserver svmluns

                                    Vserver: svmluns
```

```
                                Vserver Type: data
                             Vserver Subtype: default
                                Vserver UUID: fe75684a-61c8-11e6-b805-005056986697
                                 Root Volume: svmluns_root
                                   Aggregate: aggr1_cluster1_01
                                  NIS Domain: -
                  Root Volume Security Style: unix
                                 LDAP Client: -
                Default Volume Language Code: C.UTF-8
                             Snapshot Policy: default
                                     Comment:
                                Quota Policy: default
                 List of Aggregates Assigned: -
 Limit on Maximum Number of Volumes allowed: unlimited
                         Vserver Admin State: running
                   Vserver Operational State: running
   Vserver Operational State Stopped Reason: -
                           Allowed Protocols: iscsi
                        Disallowed Protocols: nfs, cifs, fcp, ndmp
             Is Vserver with Infinite Volume: false
                            QoS Policy Group: -
                         Caching Policy Name: -
                                 Config Lock: false
                                IPspace Name: Default
                          Foreground Process: -
                     Logical Space Reporting: false
                   Logical Space Enforcement: false

cluster1::>
```

8. Create 4 SAN LIFs for the SVM svmluns, 2 per node.

Tip: To save some typing, remember that you can use the up arrow to recall previous commands that you can then edit and execute.

```
cluster1::> network interface create -vserver svmluns -lif cluster1-01_iscsi_lif_1
-role data -data-protocol iscsi -home-node cluster1-01 -home-port e0d -subnet-name Demo
-failover-policy disabled -firewall-policy data

cluster1::> network interface create -vserver svmluns -lif cluster1-01_iscsi_lif_2
-role data -data-protocol iscsi -home-node cluster1-01 -home-port e0e -subnet-name Demo
-failover-policy disabled -firewall-policy data

cluster1::> network interface create -vserver svmluns -lif cluster1-02_iscsi_lif_1
-role data -data-protocol iscsi -home-node cluster1-02 -home-port e0d -subnet-name Demo
-failover-policy disabled -firewall-policy data

cluster1::> network interface create -vserver svmluns -lif cluster1-02_iscsi_lif_2
-role data -data-protocol iscsi -home-node cluster1-02 -home-port e0e -subnet-name Demo
-failover-policy disabled -firewall-policy data
cluster1::>
```

9. Now create a Management Interface LIF for the SVM.

```
cluster1::> network interface create -vserver svmluns -lif svmluns_admin_lif1 -role data
 -data-protocol none -home-node cluster1-01 -home-port e0c -subnet-name Demo
 -failover-policy system-defined -firewall-policy mgmt

cluster1::>
```

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

```
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 169.254.102.151/16 cluster1-01 e0a     true
            cluster1-01_clus2 up/up 169.254.95.159/16 cluster1-01  e0b     true
            cluster1-02_clus1 up/up 169.254.78.229/16 cluster1-02  e0a     true
            cluster1-02_clus2 up/up 169.254.100.67/16 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
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
```

```
svmluns
            cluster1-01_iscsi_lif_1 up/up 192.168.0.133/24 cluster1-01 e0d true
            cluster1-01_iscsi_lif_2 up/up 192.168.0.134/24 cluster1-01 e0e true
            cluster1-02_iscsi_lif_1 up/up 192.168.0.135/24 cluster1-02 e0d true
            cluster1-02_iscsi_lif_2 up/up 192.168.0.136/24 cluster1-02 e0e true
            svmluns_admin_lif1 up/up 192.168.0.137/24 cluster1-01  e0c     true
14 entries were displayed.

cluster1::>
```

11. Display detailed information for the LIF cluster1-01_iscsi_lif_1.

```
cluster1::> network interface show -lif cluster1-01_iscsi_lif_1 -instance

                    Vserver Name: svmluns
          Logical Interface Name: cluster1-01_iscsi_lif_1
                  Service Policy: default-data-blocks
                    Service List: data-core, data-iscsi
               (DEPRECATED)-Role: data
                   Data Protocol: iscsi
                 Network Address: 192.168.0.133
                         Netmask: 255.255.255.0
             Bits in the Netmask: 24
                      Is VIP LIF: false
                     Subnet Name: Demo
                       Home Node: cluster1-01
                       Home Port: e0d
                    Current Node: cluster1-01
                    Current Port: e0d
              Operational Status: up
                 Extended Status: -
                         Is Home: true
           Administrative Status: up
                 Failover Policy: disabled
                 Firewall Policy: data
                     Auto Revert: false
   Fully Qualified DNS Zone Name: none
         DNS Query Listen Enable: false
             Failover Group Name: -
                        FCP WWPN: -
                  Address family: ipv4
                         Comment: -
                  IPspace of LIF: Default
  Is Dynamic DNS Update Enabled?: false
        Probe-port for Azure ILB: -
                Broadcast Domain: Default

cluster1::>
```

12. Display a list of all the volumes on the cluster to see the root volume for the svmluns SVM.

```
cluster1::> volume show
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
cluster1-01 vol0       aggr0_cluster1_01 online RW     44.82GB    41.56GB    7%
cluster1-02 vol0       aggr0_cluster1_02 online RW     44.82GB    41.64GB    7%
svm1      eng_users    aggr1_cluster1_01 online RW        10GB     9.50GB    5%
svm1      engineering  aggr1_cluster1_01 online RW        10GB     9.50GB    5%
svm1      svm1_root    aggr1_cluster1_01 online RW        20MB    18.64MB    6%
svmluns   svmluns_root aggr1_cluster1_01 online RW        20MB    18.83MB    5%
6 entries were displayed.

cluster1::>
```

3.2.2 Create, Map, and Mount a Windows LUN

In the preceding section you created a new SVM and configured it for iSCSI. In the following sub-sections you will perform the remaining steps needed to configure and use a LUN under Windows:

- Gather the iSCSI Initiator Name of the Windows client.
- Create a thin provisioned Windows volume, create a thin provisioned Windows LUN within that volume, and map the LUN so it can be accessed by the Windows client.
- Mount the LUN on a Windows client leveraging multi-pathing.

You must complete all of the subsections of this section in order to use the LUN from the Windows client.

3.2.2.1 LUN Concepts

In ONTAP, LUNs are simply large files that reside within a FlexVol. A single FlexVol can support multiple LUNs, and it is quite common for a client that is using multiple LUNs to have all of those LUNs co-located on the same FlexVol. This facilitates taking application-consistent snapshots, and generally makes LUN management related to the client easier.

While it is possible to configure your SVM so that a volume containing a LUN is also accessible through CIFS or NFS, there is usually no advantage to be gained by doing so. Additionally, if you do not carefully manage export rules and file permissions, you can inadvertently expose the LUN file to unintended NAS clients. While it is perfectly acceptable to serve NAS and SAN clients off of the same SVM, best practice is to not mix NAS files and LUNs within the same volume, or to expose LUN volumes over NAS protocols.

An initiator group, or igroup, defines a list of the Fibre Channel WWPNs or iSCSI node names of the SAN clients that are permitted to access (or even see) a set of LUNs. Mapping a LUN to an igroup applies the igroup's access restrictions to the LUN.

Network access to an iSCSI (or FC) LUN occurs over a LIF. Unlike NAS LIFs, that can migrate online from one node to another in a cluster, SAN LIFs cannot be moved while online. Instead, Asymmetric Logical Unit Access (ALUA) provides redundant paths and automatic path selection as part of any ONTAP SAN solution. When there is an I/O interruption on a given LIF, the client simply (and transparently) retries by switching the I/O to another LIF.

Clients utilize multipathing software in conjunction with ALUA to support multiple paths to a LUN, and for path selection. Windows and Linux both include native multipathing software.

NetApp recommends configuring two SAN LIFs on each node that participate in a SAN configuration in order to provide path redundancy. The optimal path for accessing a LUN is usually to connect directly to the node that is hosting the LUN (i.e., the node that is directly connected to the SSDs/HDDs comprising the aggregate on which the volume hosting the LUN resides). In an HA configuration, that includes the node owning the LUN and its HA partner. LIFs on other nodes can also be used to access the LUN, but those paths may suffer a slight performance penalty as data will have to transfer between the involved nodes over the cluster network.

Selective LUN Mapping (SLM) is an ONTAP feature that reduces the number of paths from the host to the LUN. With SLM, when a new LUN map is created, the LUN is accessible only through paths on the node owning the LUN and its HA partner. This is very helpful in reducing complexity for large clusters (with many nodes) that may present many potential paths to a given LUN.

Important: Beginning with ONTAP 9.7, NetApp has started shipping a 2-node AFF All SAN Array option available for factory-ordered and shipped AFF systems. The new AFF All SAN Array (ASA) offers mission-critical workloads (such as databases), a symmetric active-active pathing in a 2-node configuration to offer an even higher level of continuous availability in a simple setup for SAN workloads. While a standard AFF offers a more scalable multi-protocol solution with active-passive (non-optimized) pathing, the AFF ASA SAN system provide a higher level of uninterrupted access for SAN protocols during a planned or unplanned storage failover event.

3.2.2.2 Gather the Windows Client iSCSI Initiator Name

You need to determine the Windows client's iSCSI initiator name so that when you create the LUN you can set up an appropriate initiator group to control access to the LUN.

On the desktop of the Windows client named Jumphost (the main Windows host you use in the lab), perform the following tasks:

1. Click on the **Windows** icon on the far left side of the task bar.

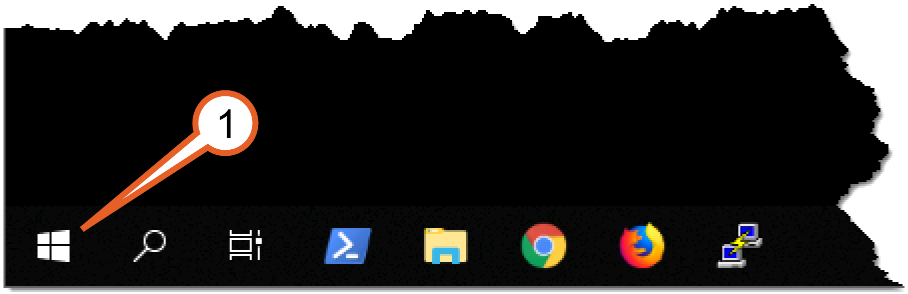


Figure 3-1:

The “Start” screen opens.

2. Click on **Windows Administrative Tools**.

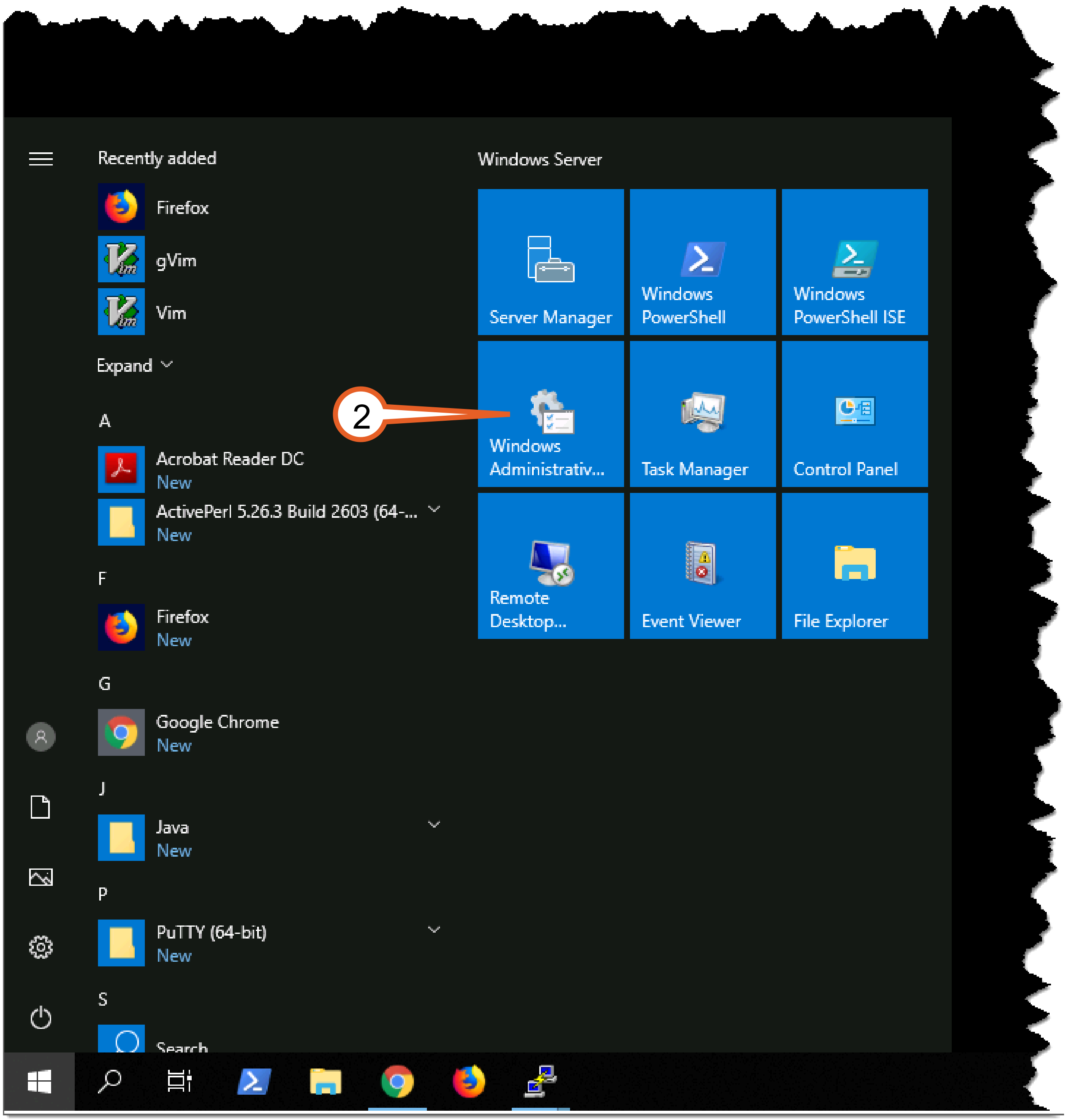


Figure 3-2:

Windows Explorer opens to the List of Administrative Tools.

3. Double-click the entry for the **iSCSI Initiator** tool.

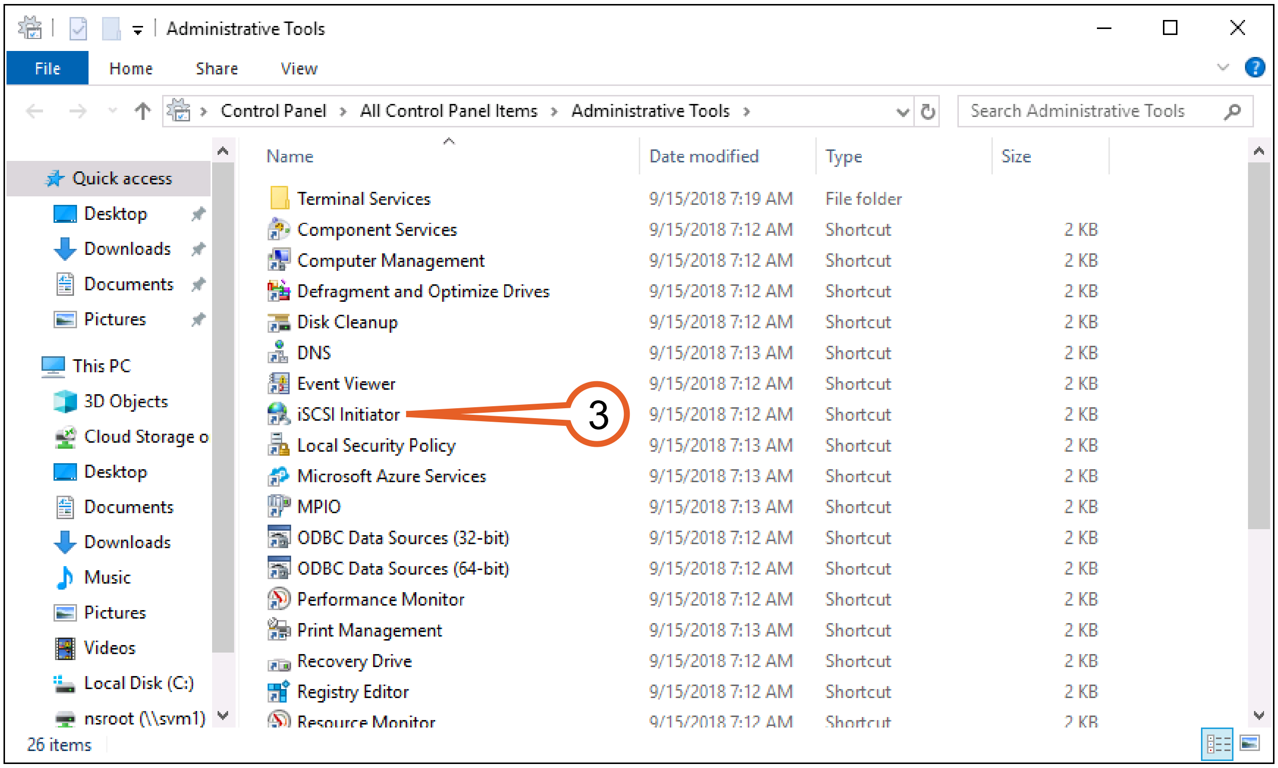


Figure 3-3:

The “iSCSI Initiator Properties” window opens.

4. Select the **Configuration** tab.
5. Take note of the value in the “Initiator Name” field, which contains the initiator name for Jumphost.

Attention: The initiator name is “**iqn.1991-05.com.microsoft:jumphost.demo.netapp.com**”. You will need this value later, so you might want to copy this value from the properties window and paste it into a text file on your lab's desktop so you have it readily available when that time comes.

6. Click **Cancel**.

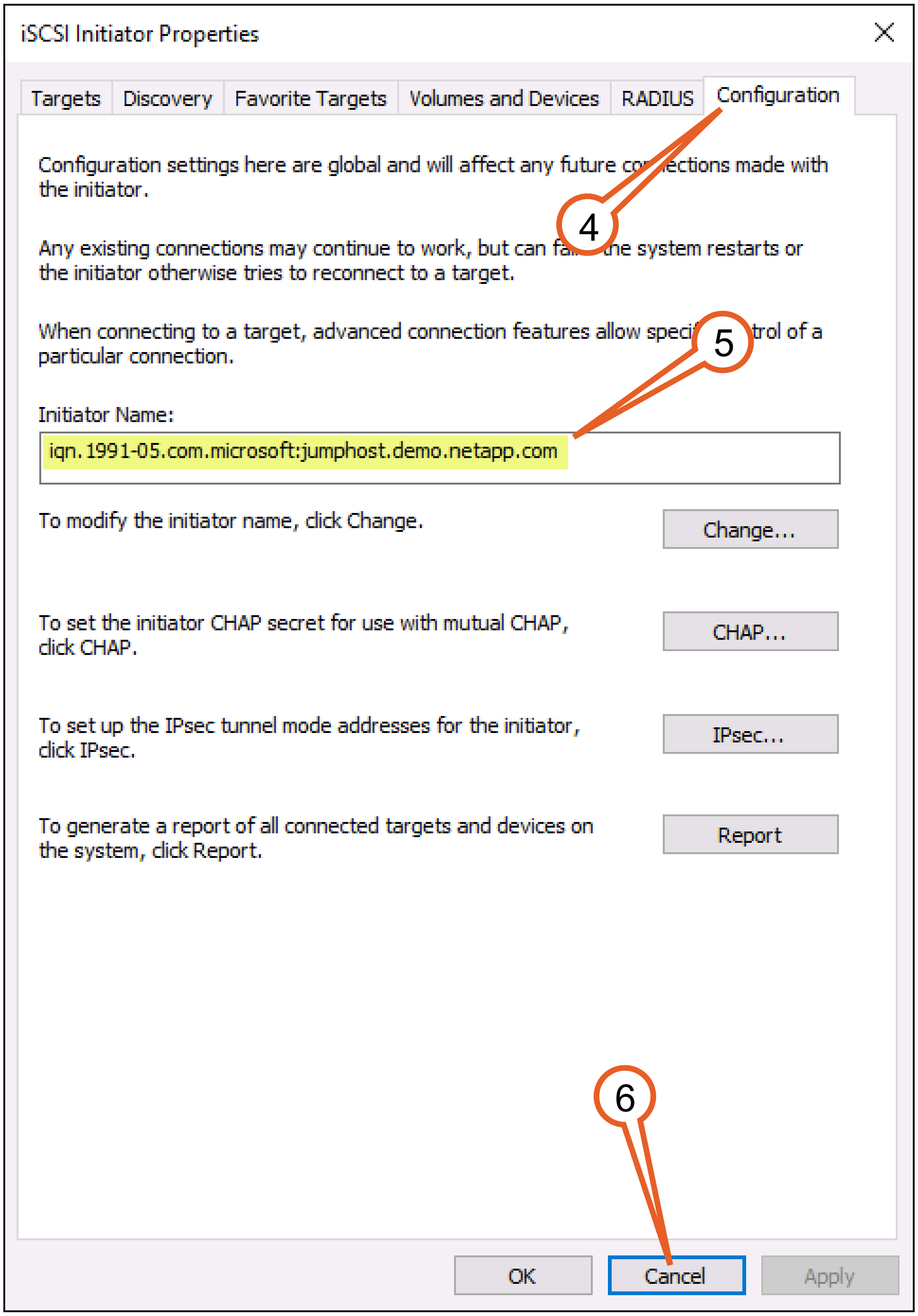


Figure 3-4:

The “iSCSI Properties” window closes, and focus returns to the “Windows Explorer Administrator Tools” window. Leave this window open because you will need to access other Administrator tools later in the lab.

3.2.2.3 Create and Map a Windows LUN

You will now create a new thin provisioned Windows LUN named “windows.lun” in the volume winluns on the SVM “svmluns”. You will also create an initiator igroup for the LUN and populate it so that only the Windows host “jumphost” can access it.

1. If you do not already have a PuTTY connection open to cluster1 then please open one now following the instructions in the “Accessing the Command Line” section at the beginning of this lab guide.
2. Create the volume “winluns” to host the Windows LUN you will be creating in a later step.

```
cluster1::> volume create -vserver svmluns -volume winluns -aggregate aggr1_cluster1_01
 -size 10.31GB -percent-snapshot-space 0 -snapshot-policy none -space-guarantee none
 -autosize-mode grow -nvfail on -tiering-policy none

Warning: The export-policy "default" has no rules in it. The volume will therefore be
         inaccessible.
Do you want to continue? {y|n}: y
[Job 53] Job is queued: Create winluns.
[Job 53] Job succeeded: Successful

cluster1::>
```

Note: In your lab you may not get the warning about the default export policy containing no rules, as it appears to have been eliminated in ONTAP 9.5. Remember that export policies are only applicable for NAS protocols, so if you do get the warning in this exercise then you can safely ignore it since the svmluns SVM only has the the iscsi protocol enabled.

3. Display a list of the volumes on the cluster.

```
cluster1::> volume show
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
cluster1-01 vol0       aggr0_cluster1_01 online RW     23.01GB    20.07GB    7%
cluster1-02 vol0       aggr0_cluster1_02 online RW     23.01GB    20.32GB    7%
svm1      eng_users    aggr1_cluster1_01 online RW        10GB     9.50GB    5%
svm1      engineering  aggr1_cluster1_01 online RW        10GB     9.50GB    5%
svm1      svm1_root    aggr1_cluster1_01 online RW        20MB    18.62MB    6%
svmluns   svmluns_root aggr1_cluster1_01 online RW        20MB    18.80MB    6%
svmluns   winluns      aggr1_cluster1_01 online RW     10.31GB    10.31GB    0%
7 entries were displayed.

cluster1::>
```

4. Create the Windows LUN named “windows.lun”.

```
cluster1::> lun create -vserver svmluns -volume winluns -lun windows.lun
 -size 10GB -ostype windows_2008 -space-reserve disabled
Created a LUN of size 10g (10742215680)

cluster1::>
```

5. Add a comment to the LUN definition.

```
cluster1::> lun modify -vserver svmluns -volume winluns -lun windows.lun
 -comment "Windows LUN"
cluster1::>
```

6. Display the LUNs on the cluster.

```
cluster1::> lun show
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------- ------- -------- -------- --------
svmluns   /vol/winluns/windows.lun        online  unmapped windows_2008 10.00GB

cluster1::>
```

7. Display a list of the defined igroups.

```
cluster1::> igroup show
This table is currently empty.
cluster1::>
```

8. Create a new igroup named “winigrp” that you will use to manage access to the new LUN, and add Jumphost's initiator to the group.

```
cluster1::> igroup create -vserver svmluns -igroup winigrp -protocol iscsi
 -ostype windows -initiator iqn.1991-05.com.microsoft:jumphost.demo.netapp.com
cluster1::>
```

9. Verify the winigrp igroup's existence and member initiator..

```
cluster1::> igroup show
Vserver   Igroup       Protocol OS Type  Initiators
--------- ------------ -------- -------- ------------------------------------
svmluns   winigrp      iscsi    windows  iqn.1991-05.com.microsoft:Jumphost.
                                         demo.netapp.com
cluster1::>
```

10. Map the LUN “windows.lun” to the igroup “winigrp”.

```
cluster1::> lun map -vserver svmluns -volume winluns -lun windows.lun -igroup winigrp
cluster1::>
```

11. Display a list of all the LUNs.

```
cluster1::> lun show
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------- ------- -------- -------- --------
svmluns   /vol/winluns/windows.lun        online  mapped   windows_2008 10.00GB

cluster1::>
```

12. Display a list of all the mapped LUNs.

```
cluster1::> lun mapped show
Vserver    Path                                      Igroup   LUN ID  Protocol
---------- ---------------------------------------- -------  ------  --------
svmluns    /vol/winluns/windows.lun                  winigrp       0  iscsi

cluster1::>
```

13. Display a detailed report on the configuration of the LUN “windows.lun”.

```
cluster1::> lun show -lun windows.lun -instance

                  Vserver Name: svmluns
                      LUN Path: /vol/winluns/windows.lun
                   Volume Name: winluns
                    Qtree Name: ""
                      LUN Name: windows.lun
                      LUN Size: 10.00GB
                       OS Type: windows_2008
             Space Reservation: disabled
                 Serial Number: wOj6b]QPl/dT
           Serial Number (Hex): 774f6a36625d51506c2f6454
                       Comment: Windows LUN
    Space Reservations Honored: false
              Space Allocation: disabled
                         State: online
                      LUN UUID: 5553bf4a-f607-4b59-bffb-3befdf8b3880
                        Mapped: mapped
Physical Size of Logical Block: 512B
              Device Legacy ID: -
              Device Binary ID: -
                Device Text ID: -
                     Read Only: false
         Fenced Due to Restore: false
                     Used Size: 0
           Maximum Resize Size: 502.0GB
                 Creation Time: 12/18/2020 00:01:54
                         Class: regular
          Node Hosting the LUN: cluster1-01
              QoS Policy Group: -
     QoS Adaptive Policy Group: -
           Caching Policy Name: -
                         Clone: false
      Clone Autodelete Enabled: false
           Inconsistent Import: false
                   Application: -

cluster1::>
```

ONTAP supports a space reclamation feature that allows it to reclaim space from a thin provisioned LUN when the client deletes data from the LUN, and also allows ONTAP to notify the client when the LUN cannot accept writes due to lack of space on the volume. This feature is supported by VMware

ESX 5.0 and later, Red Hat Enterprise Linux 6.2 and later, and Microsoft Windows 2012. Jumphost is running Windows 2012R2, so you will enable the space reclamation feature for your Windows LUN. You can only enable space reclamation through the Data ONTAP command line.

14. In the cluster1 CLI, view whether space reclamation is enabled for the LUN.

```
cluster1::> lun show -vserver svmluns -path /vol/winluns/windows.lun
 -fields space-allocation
vserver path                     space-allocation
------- ------------------------ ----------------
svmluns /vol/winluns/windows.lun disabled

cluster1::>
```

15. Enable space reclamation for the LUN “windows.lun”.

```
cluster1::> lun modify -vserver svmluns -path /vol/winluns/windows.lun
 -space-allocation enabled

cluster1::>
```

16. View the LUN's space reclamation setting again.

```
cluster1::> lun show -vserver svmluns -path /vol/winluns/windows.lun
 -fields space-allocation
vserver path                     space-allocation
------- ------------------------ ----------------
svmluns /vol/winluns/windows.lun enabled

cluster1::>
```

3.2.2.4 Mount the LUN on a Windows Client

In this lab activity you mount the LUN on the Windows client.

Begin by validating that the Multi-Path I/O (MPIO) software is working properly on this windows host.

1. On the desktop of Jumphost, in the “Administrative Tools” window (which you should still have open from a previous exercise), double-click the **MPIO** tool.

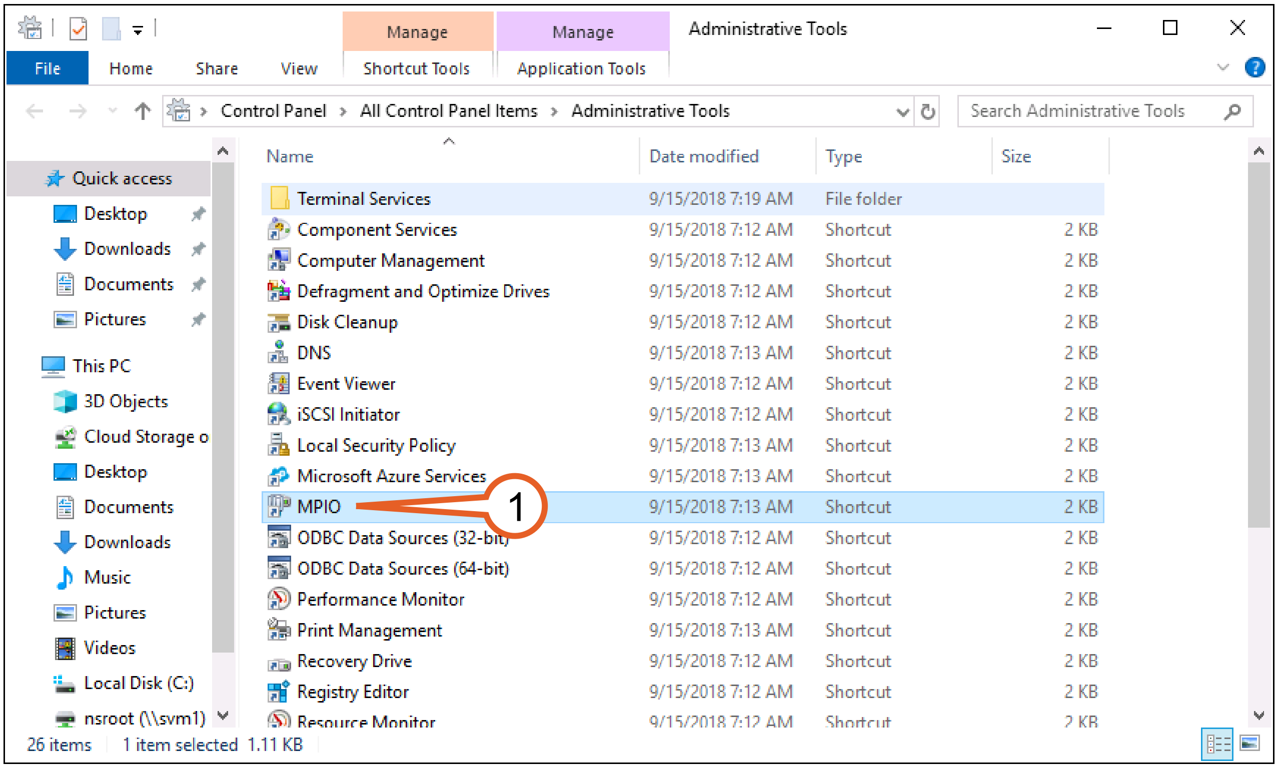


Figure 3-5:

2. Select the **Discover Multi-Paths** tab.
3. Examine the **Add Support for iSCSI devices** checkbox. If this checkbox is NOT greyed out then MPIO is improperly configured.

> **Tip:** This checkbox should be greyed out for this lab. If it is not, place a check in that checkbox, click **Add**, then if prompted to reboot click **Yes**. Once the system finishes rebooting, return to this window to verify that the checkbox is now greyed out, indicating that MPIO is properly configured.

4. Click **Cancel**.

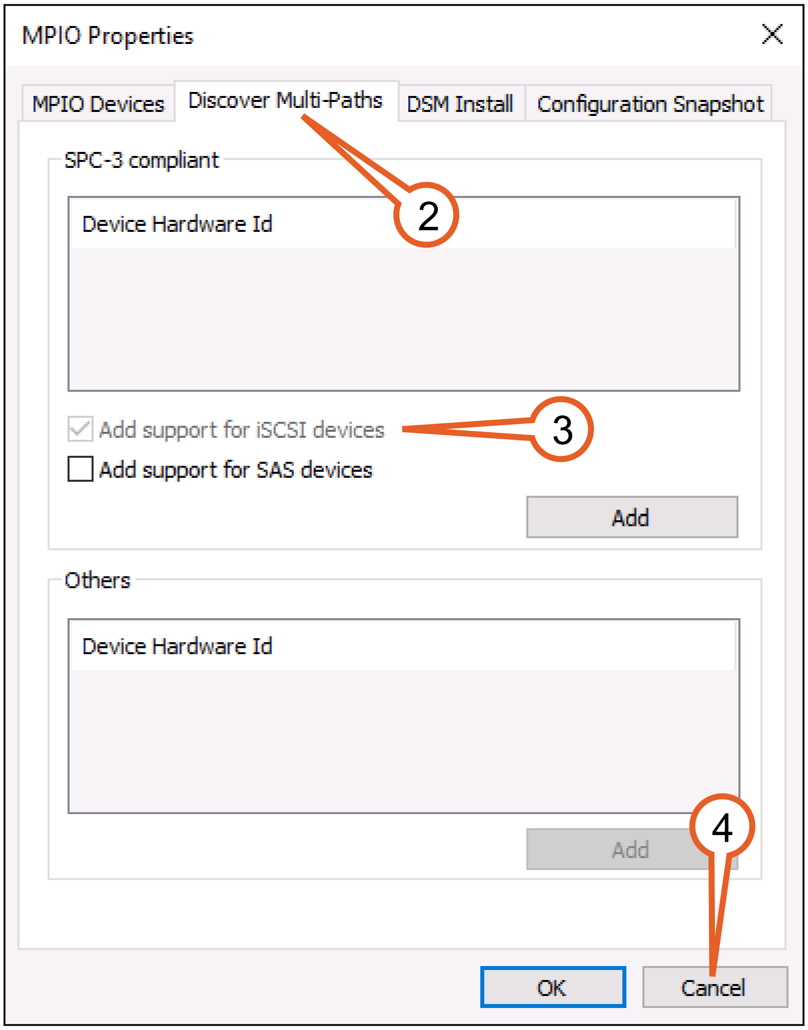


Figure 3-6:

The “MPIO Properties” dialog closes and focus returns to the “Administrative Tools” window for Jumphost. Now you can start connecting Jumphost to the LUN.

5. In “Administrative Tools”, double-click the **iSCSI Initiator** tool.

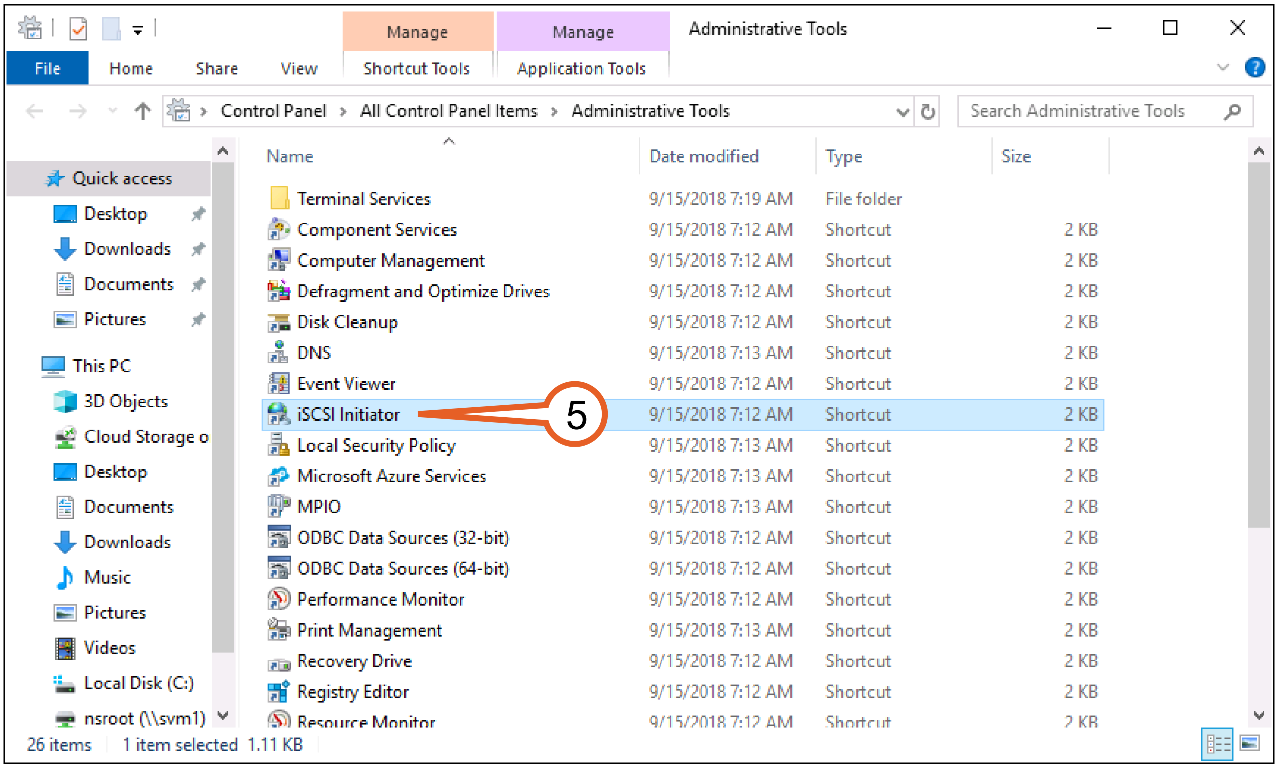


Figure 3-7:

The “iSCSI Initiator Properties” dialog opens.

6. Select the **Targets** tab.
7. Notice that there are no targets listed in the “Discovered Targets” list box. This indicates that there are currently no iSCSI targets mapped to this host.

8. Click the **Discovery** tab.

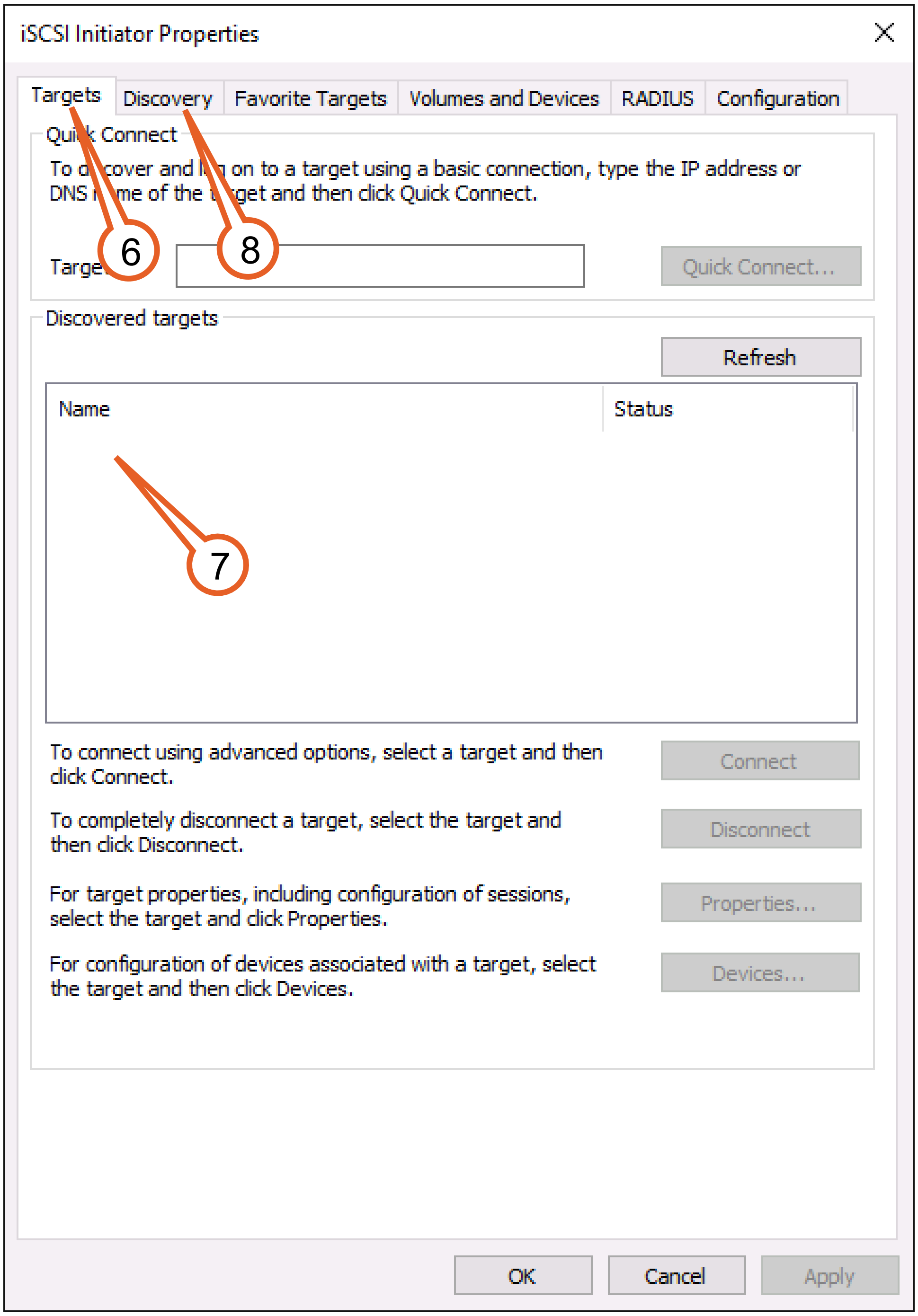


Figure 3-8:

The **Discovery** tab is where you begin the process of discovering LUNs, and to do that you must define a target portal to scan. You are going to manually add a target portal.

9. Click the **Discover Portal…** button.

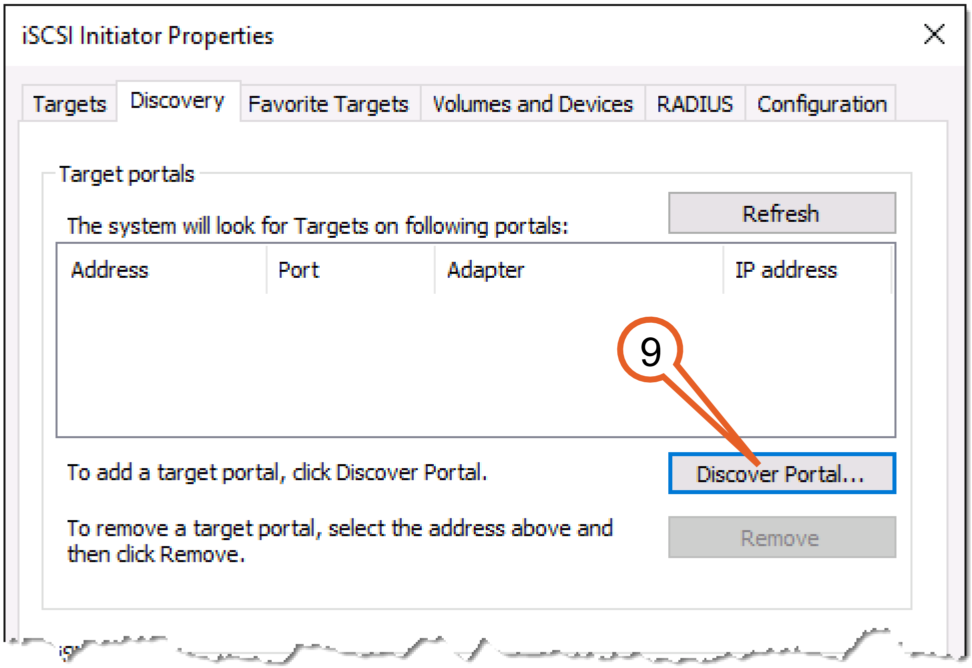


Figure 3-9:

The “Discover Target Portal” dialog opens. Here you specify the first of the IP addresses that the ONTAP Create LUN wizard assigned to your iSCSI LIFs when you created the svmluns SVM. Recall that the wizard assigned your LIF’s IP addresses in the range 192.168.0.133-192.168.0.136.

10. Set the “IP Address or DNS name” textbox to **`192.168.0.133`**, the first address in the range for your LIFs.
11. Click **OK**.

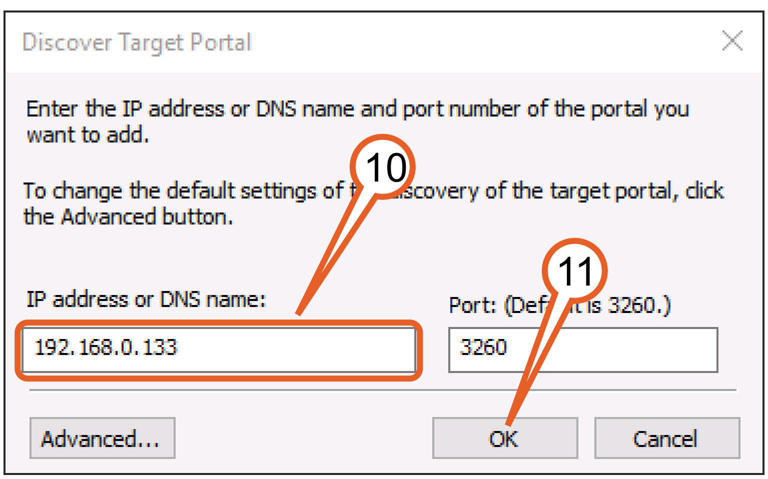


Figure 3-10:

The “Discover Target Portal” dialog closes, and focus returns to the “iSCSI Initiator Properties” dialog.

12. The “Target Portals” list now contains an entry for the IP address you entered in the previous step.

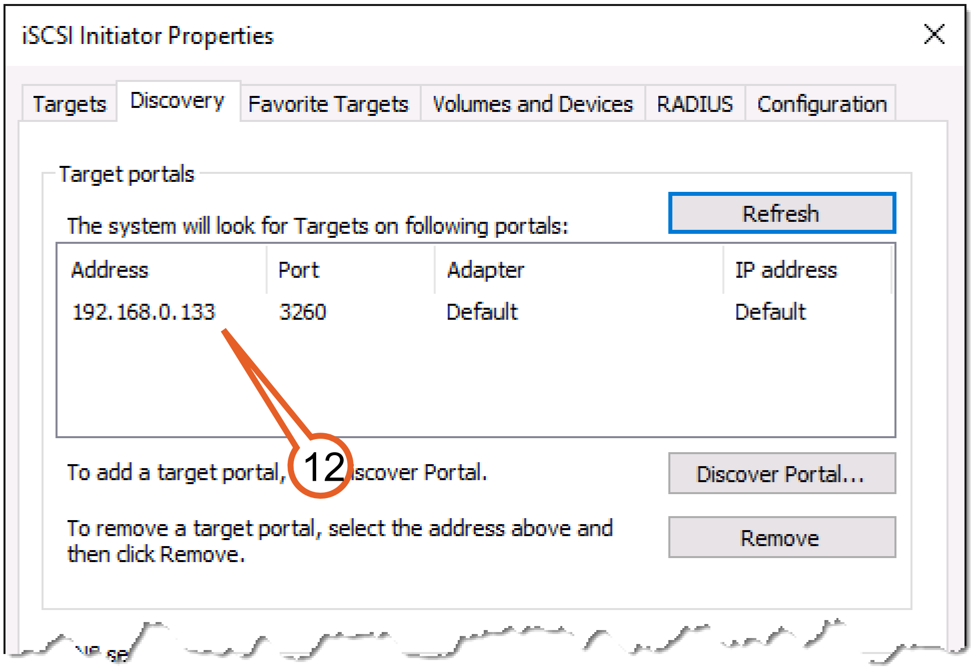


Figure 3-11:

13. Click on the **Targets** tab.
14. In the “Discovered targets”, list select the only listed target, **iqn.1992-08.com.netapp:...**. Observe that the target's status is Inactive, because although you have discovered it you have not yet connected to it. Also note that the “Name” of the discovered target in your lab will have a different value than what you see in this guide; that name string is uniquely generated for each instance of the lab.

Note: Make a mental note of that string value as you will see it a lot as you continue to configure iSCSI in later steps of this procedure.

15. Click **Connect**.

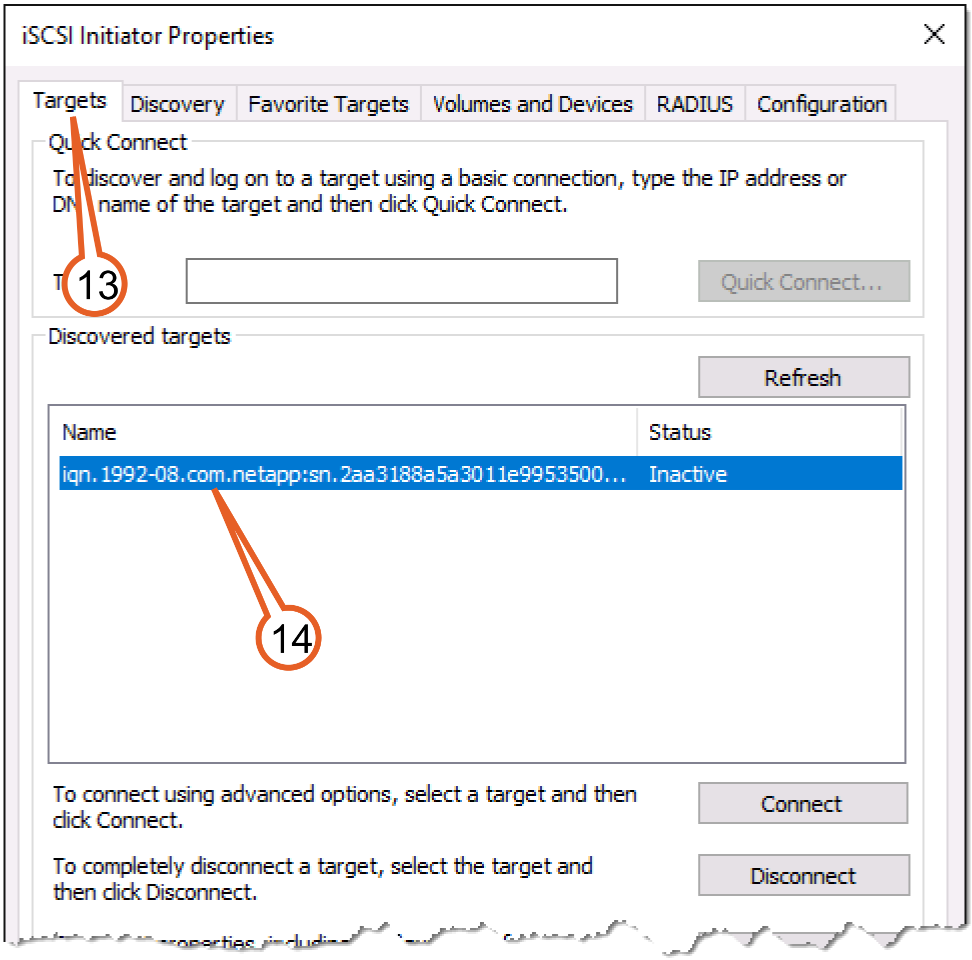


Figure 3-12:

The “Connect to Target” dialog box opens.

16. Click the **Enable multi-path** checkbox.
17. Click **Advanced…**.

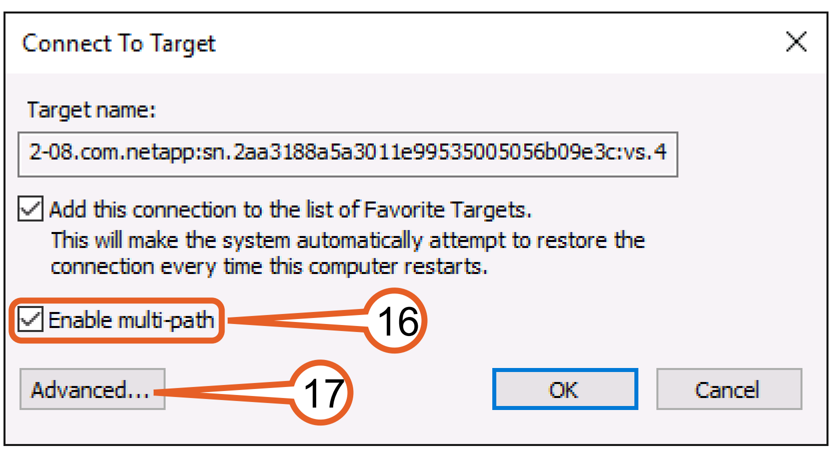


Figure 3-13:

The “Advanced Settings” dialog opens.

18. In the “Target portal IP” dropdown menu select the entry containing the IP address you specified when you discovered the target portal, which should be **192.168.0.133**. The listed values are IP Address and Port number combinations, and the specific value you want to select is **192.168.0.133 / 3260**.
19. When finished, click **OK**.

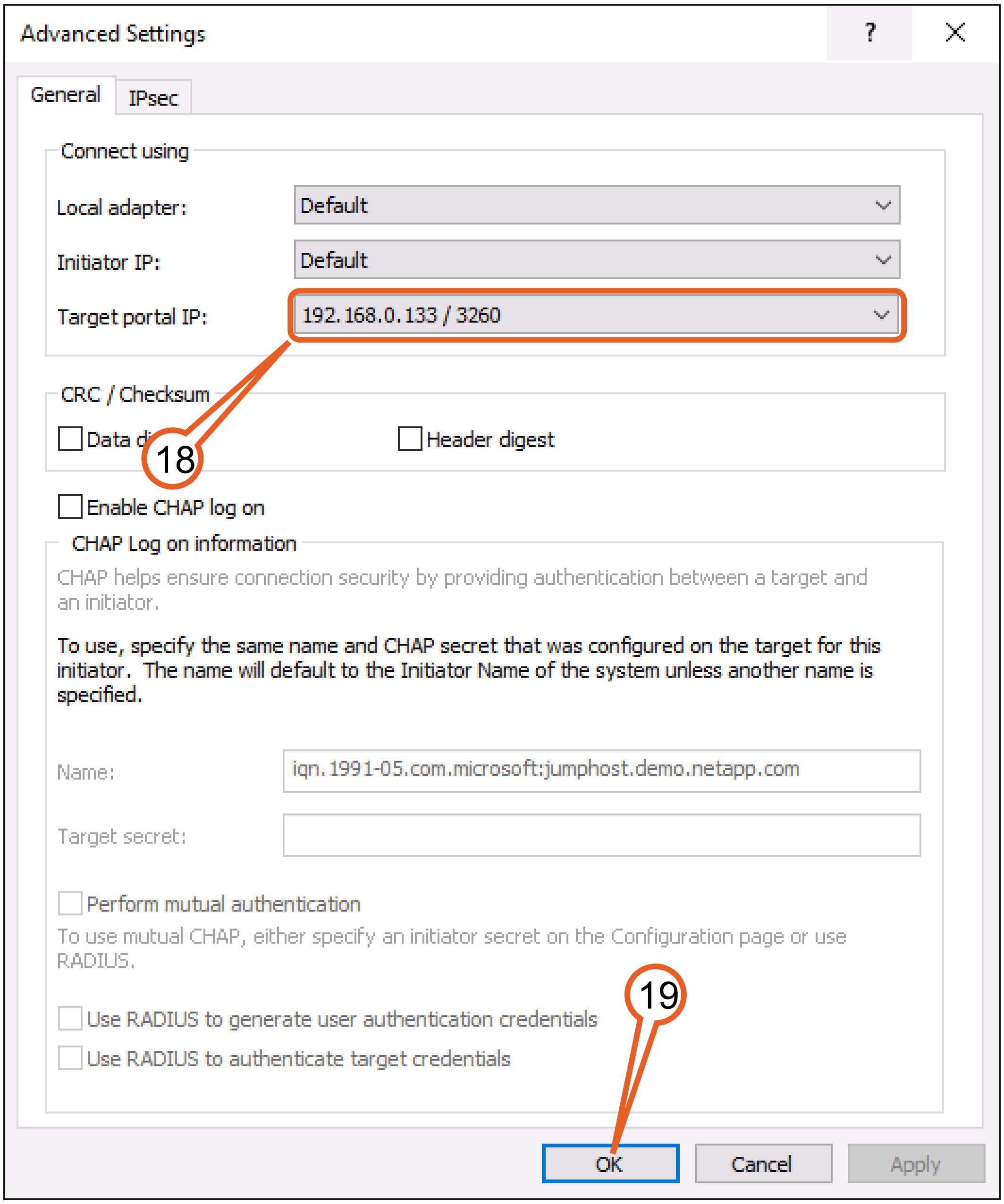


Figure 3-14:

The “Advanced Setting” dialog closes, and focus returns to the “Connect to Target” dialog.

20. Click **OK**.

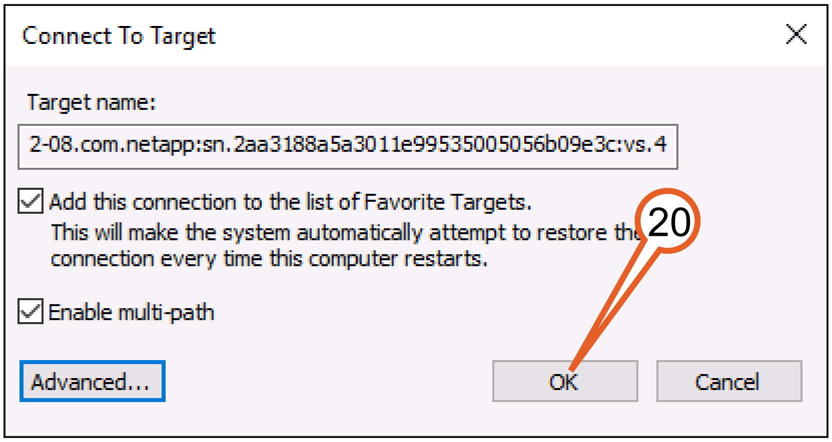


Figure 3-15:

The “Connect to Target” window closes, and focus returns to the “iSCSI Initiator Properties” window.

21. Notice that the status of the listed discovered target has changed from “Inactive” to “Connected”.

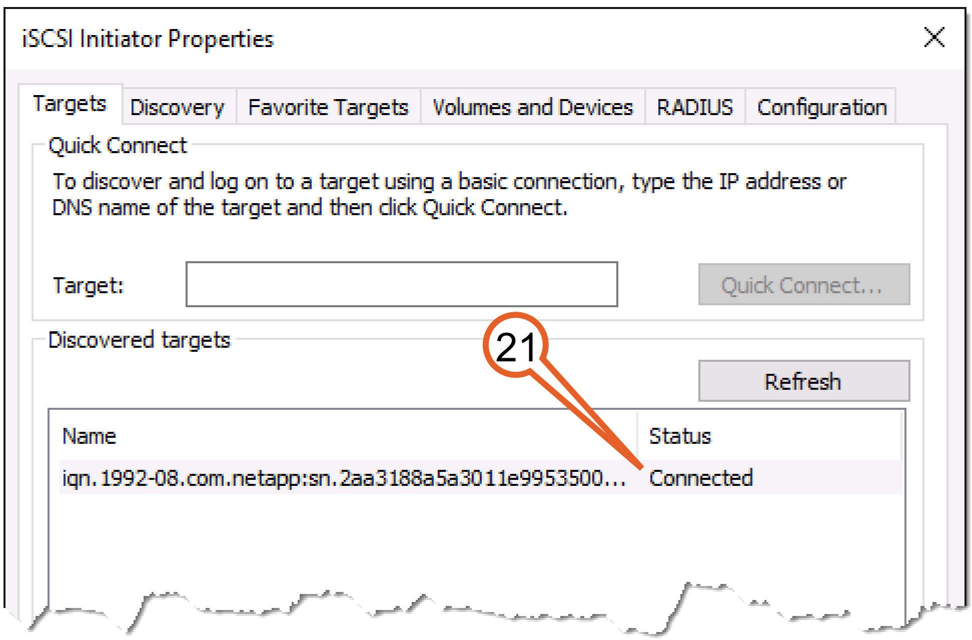


Figure 3-16:

Up to this point you have added a single path to your iSCSI LUN. The 192.168.0.133 address you selected corresponds to the cluster1-01_iscsi_lif_1 SAN LIF you created for the “svmluns” SVM in the preceding exercise. You created three additional SAN LIFs during that same exercise, and now you are going to add each of their IP addresses as alternate paths to this same LUN. To begin this procedure you must edit the properties of your existing iSCSI connection.

22. Still on the “Targets” tab, select the discovered target entry for your existing connection.

23. Click **Properties**.

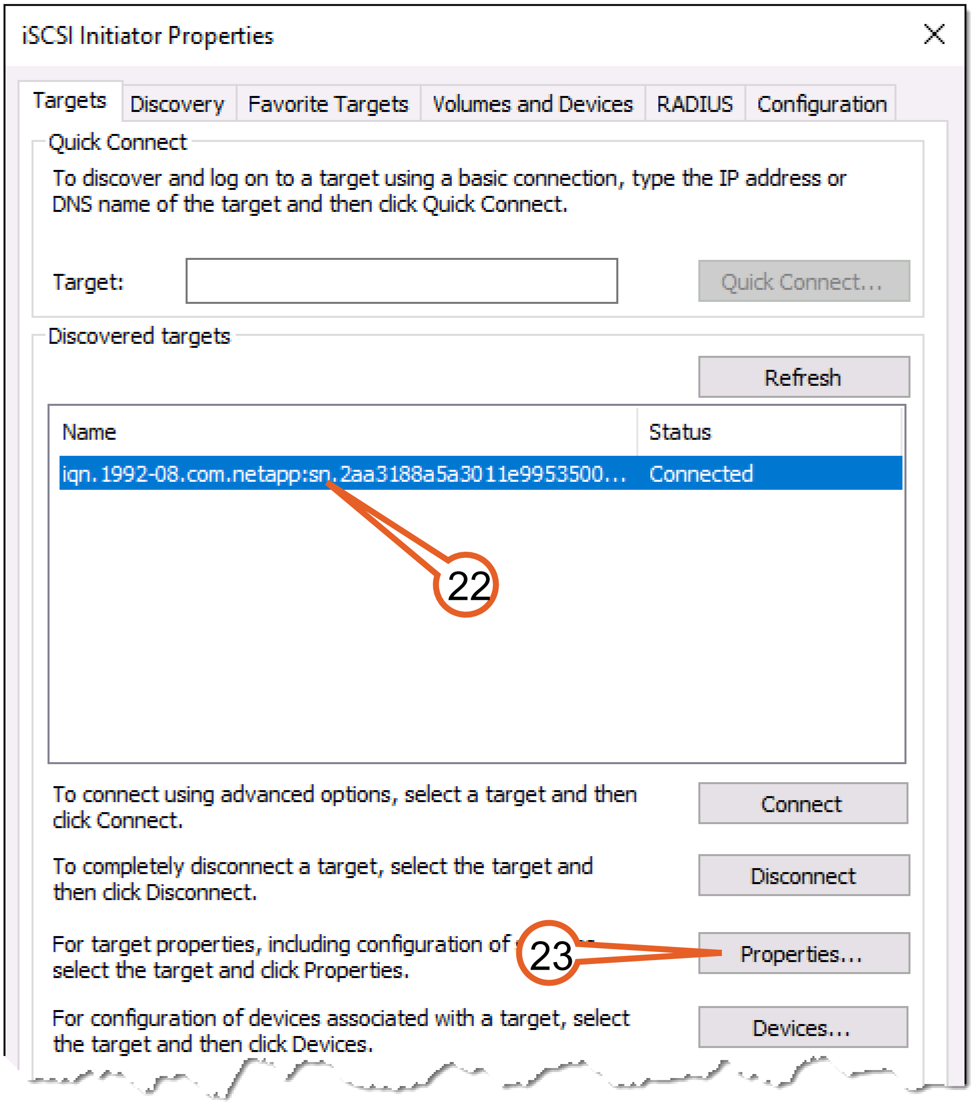


Figure 3-17:

The “Properties” dialog opens. From this window you start to connect alternate paths for your newly connected LUN. You will repeat this procedure 3 times, once for each of the remaining LIFs that are present on the svmluns SVM.

LIF IP Address
192.168.0.134
192.168.0.135
192.168.0.136

24. The Identifier list contains an entry for every path you have specified so far, so it can serve as a visual indicator of your progress for specifying all your paths. The first time you enter this window you will see one entry for the LIF you used to first connect to this LUN. When you are done you will see four entries in this window.

25. Click **Add Session**.

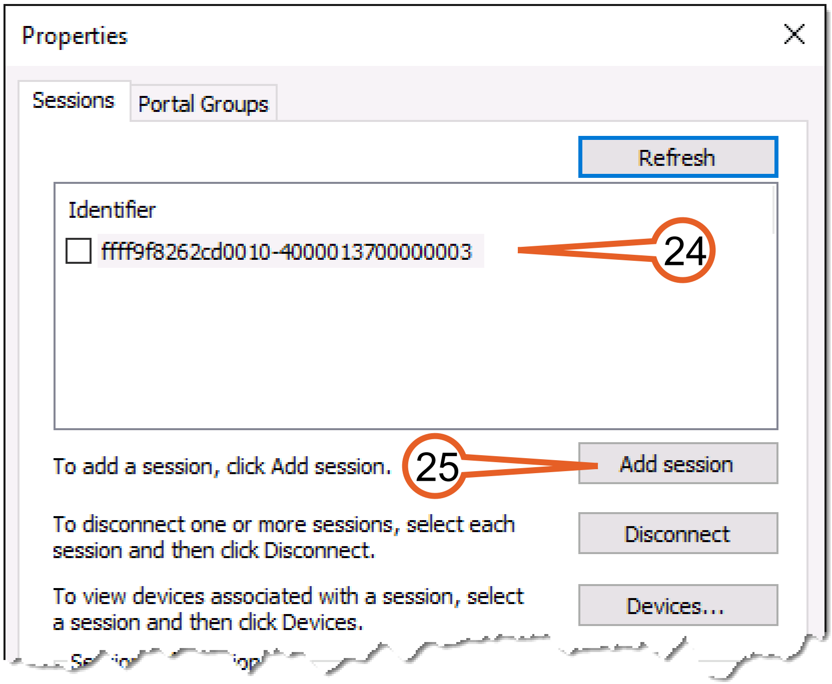


Figure 3-18:

The “Connect to Target” dialog opens.

26. Check the **Enable muti-path** checkbox.
27. Click **Advanced.…**

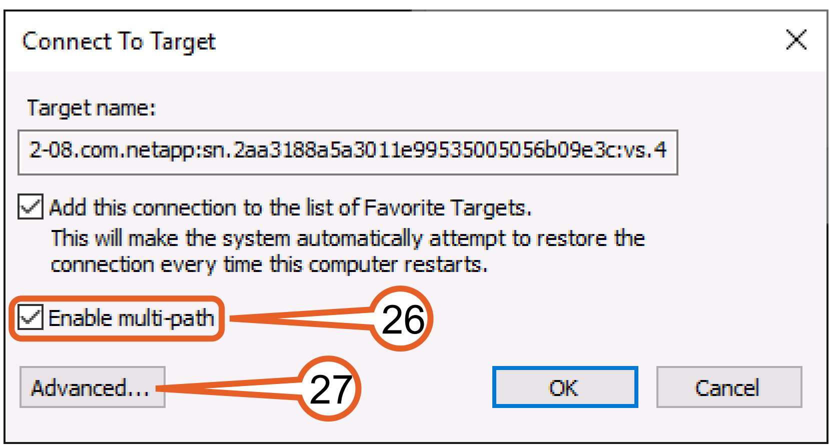


Figure 3-19:

The “Advanced Setting” dialog opens.

28. Select the “Target port IP” entry that contains the IP address of the specific LIF you are adding as an alternate path. The following screenshot shows the 192.168.0.134 address, but the value you specify depends of which specific path you are configuring.

29. When finished, click **OK**.

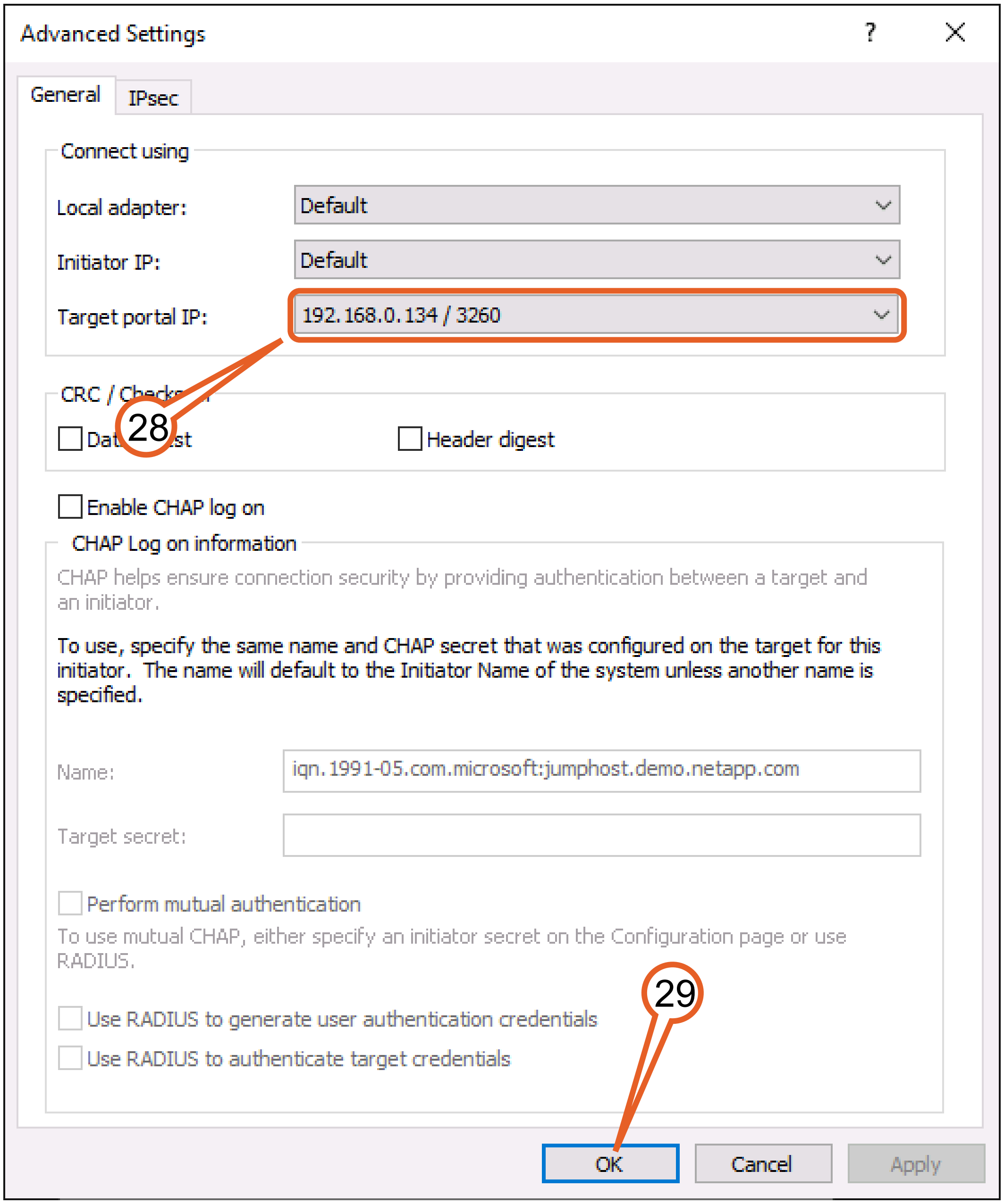


Figure 3-20:

The “Advanced Settings” dialog closes, and focus returns to the “Connect to Target” dialog.

30. Click **OK**.

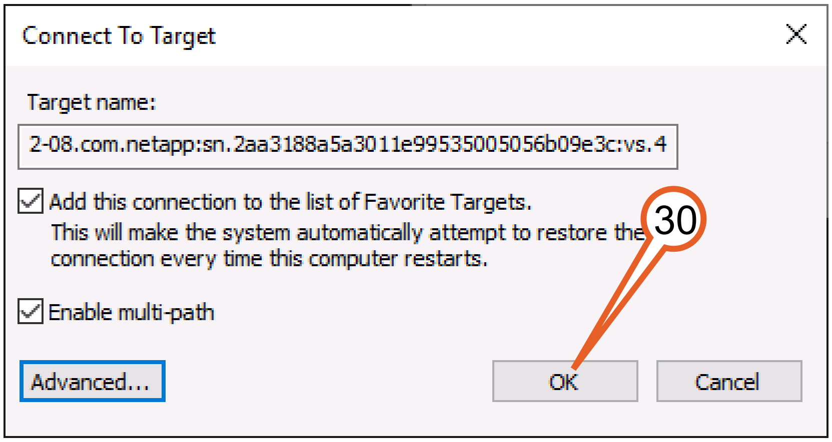


Figure 3-21:

The “Connect to Target” dialog closes, and focus returns to the “Properties” dialog where there are now two entries shown in the identifier list.

Repeat steps 24 - 30 for each of the last two remaining LIF IP addresses. When you have finished adding all the additional paths, the “Identifier” list in the Properties window should contain 4 entries.

31. There are 4 entries in the “Identifier” list when you are finished, indicating that there are 4 sessions, one for each path. Note that it is okay if the identifier values in your lab differ from those in the screenshot.

32. Click **OK**.

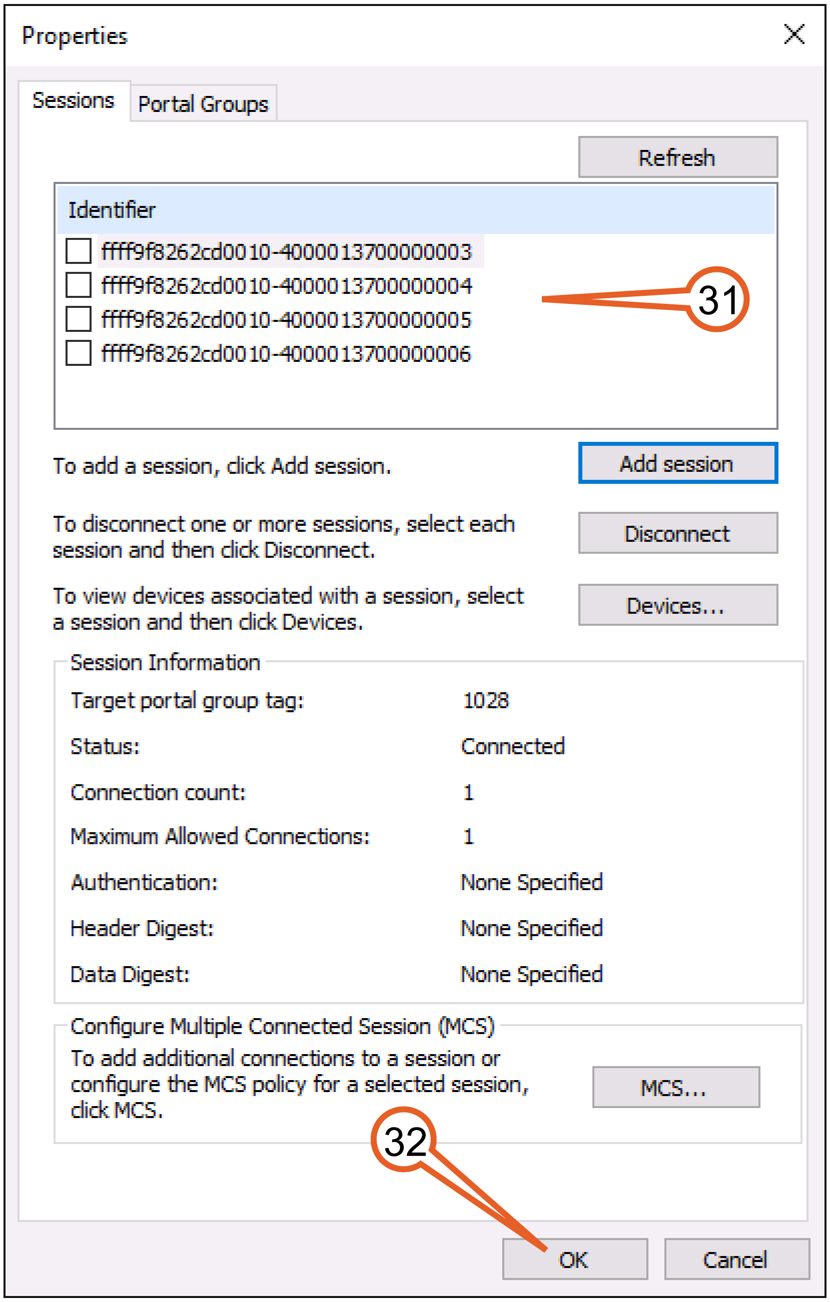


Figure 3-22:

The “Properties” dialog closes, and focus returns to the “iSCSI Properties” dialog.

33. Click **OK**.

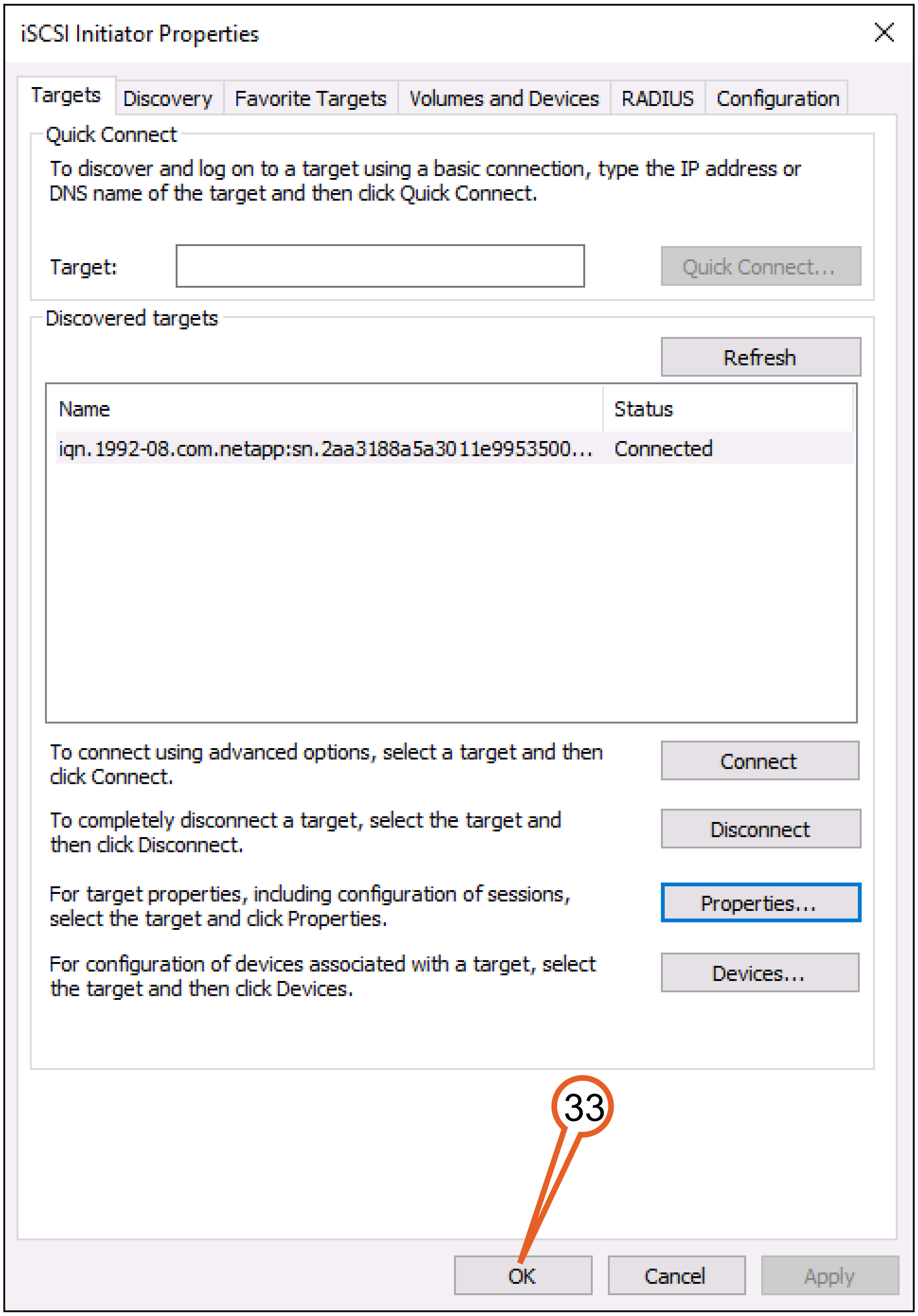


Figure 3-23:

The “iSCSI Properties” window closes, and focus returns to the desktop of Jumphost. If the “Administrative Tools” window is not still open on your desktop, open it again now.

The Jumphost should now be connected to the LUN using multi-pathing, so it is time to format your LUN and build a filesystem on it.

34. In “Administrative Tools”, double-click the **Computer Management** tool.

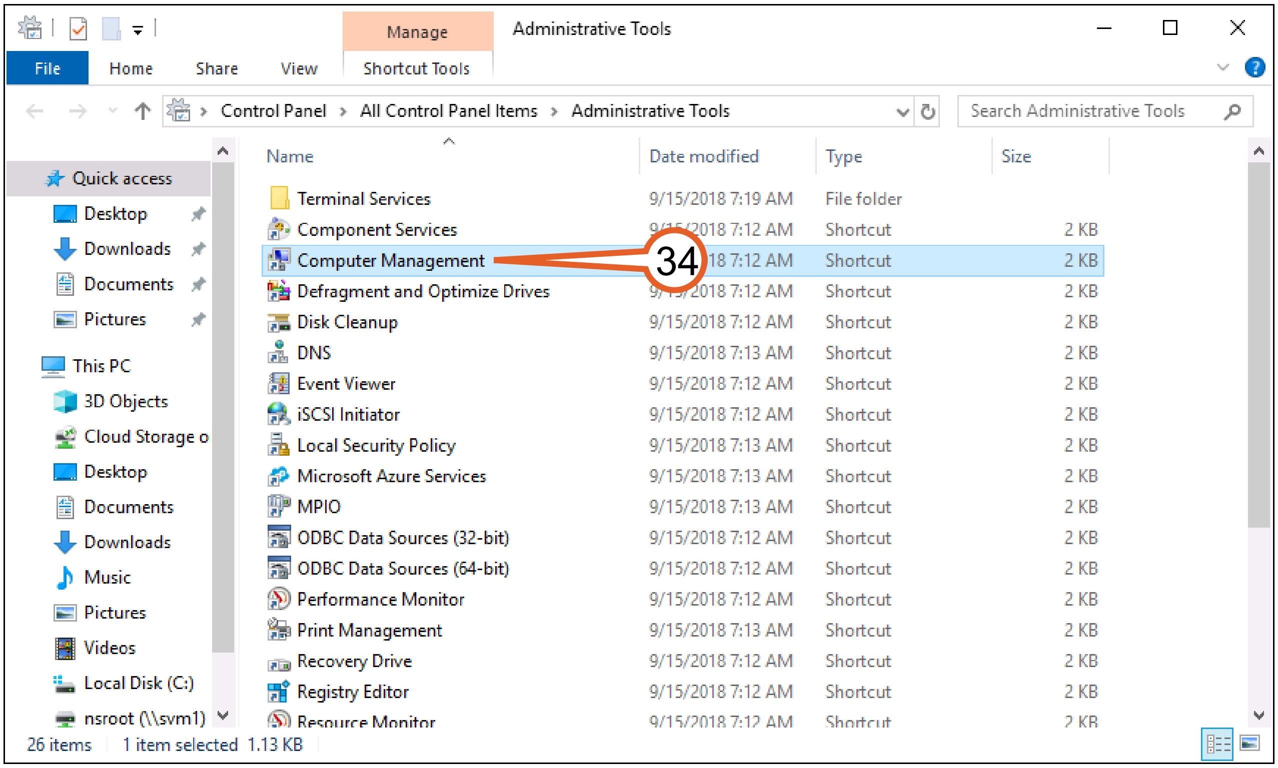


Figure 3-24:

The “Computer Management” window opens.

35. In the left pane of the “Computer Management” window, navigate to **Computer Management (Local)** > **Storage** > **Disk Management**.

36. In the middle pane, locate the entry for Disk 1 and observe that it is currently offline.

Tip: If you see more disks listed than Disk 0 (the boot disk) and Disk 1 (the LUN), MPIO has not correctly recognized that the multiple paths you set up are all for the same LUN. If this occurs, you need to quit **Computer Manager** and go back to the **iSCSI Initiator** tool to review your path configuration steps to find and correct any configuration errors. After that you can return to the Computer Management tool and try again.

37. Right-click on the listing for **Disk 1** and select **Online** from the context menu.

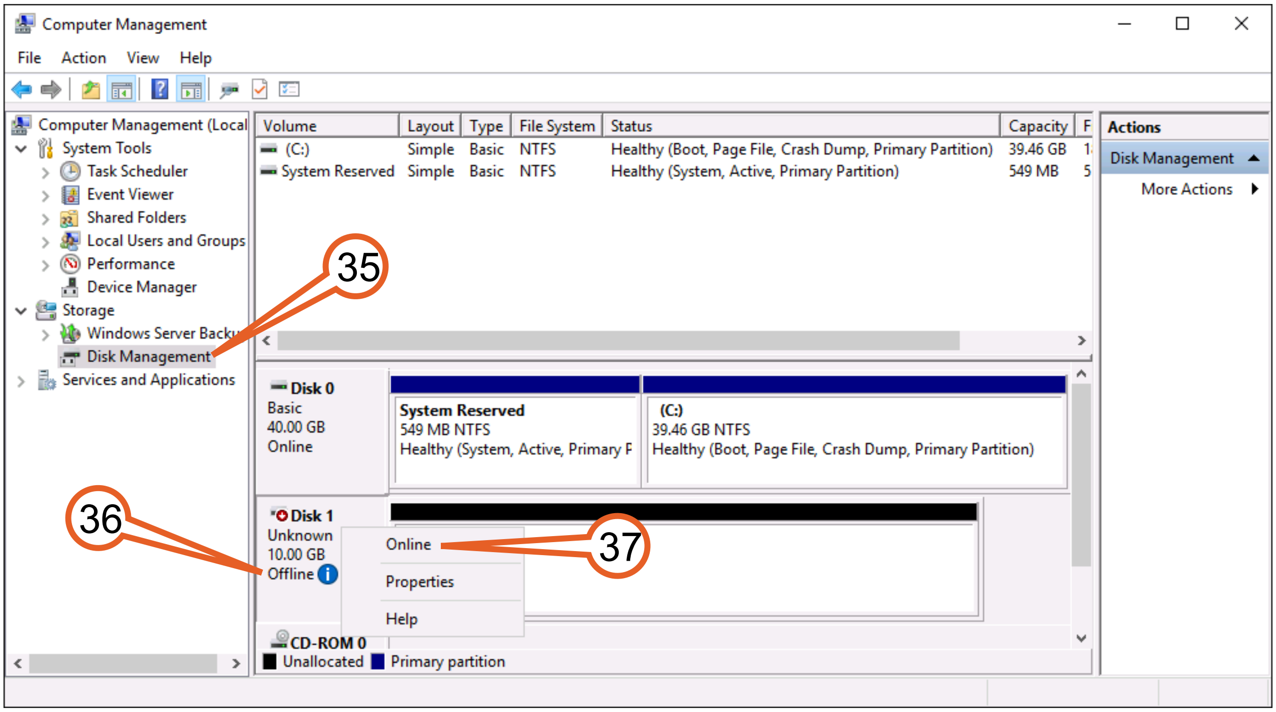


Figure 3-25:

38. The entry for Disk 1 now shows as "Unknown" and "Not Initialized".
39. Right-click inside the entry for **Disk 1** and select **Initialize Disk** from the context menu.

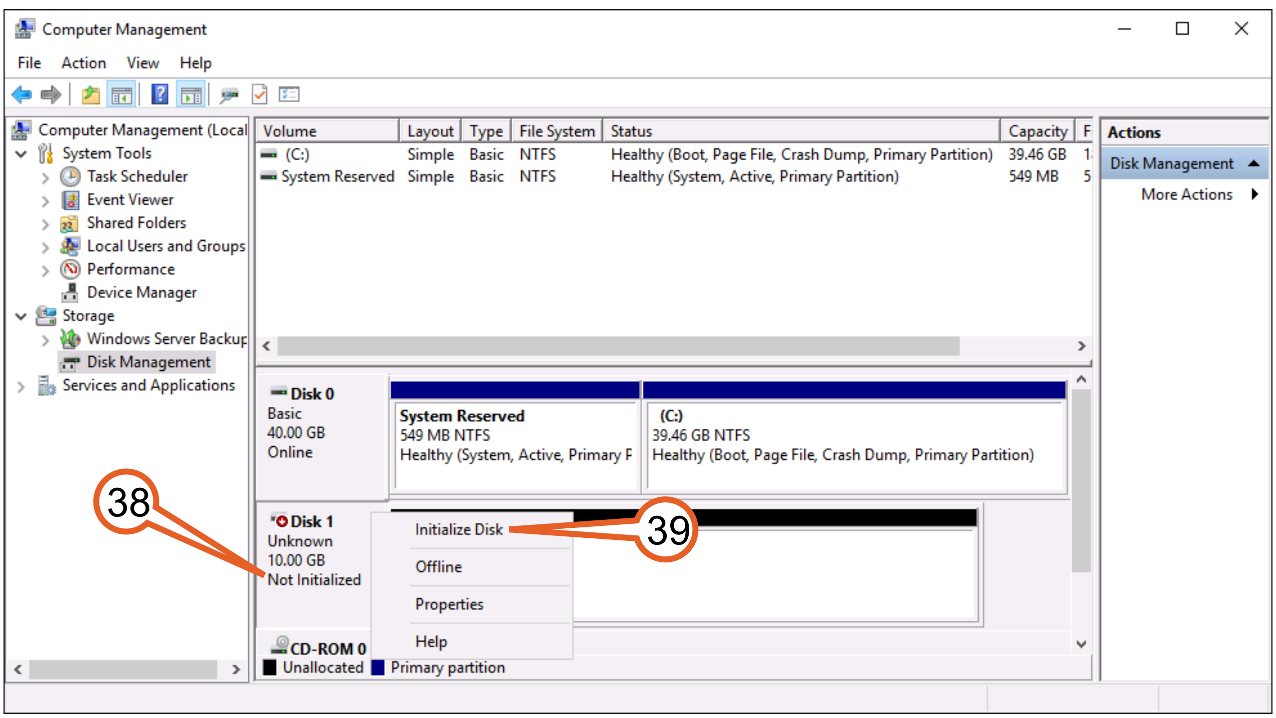


Figure 3-26:

The "Initialize Disk" dialog opens.

40. When you launch Disk Management, an “Initialize Disk” dialog will open informing you that you must initialize a new disk before Logical Disk Manager can access it.

Click **OK** to initialize the disk.

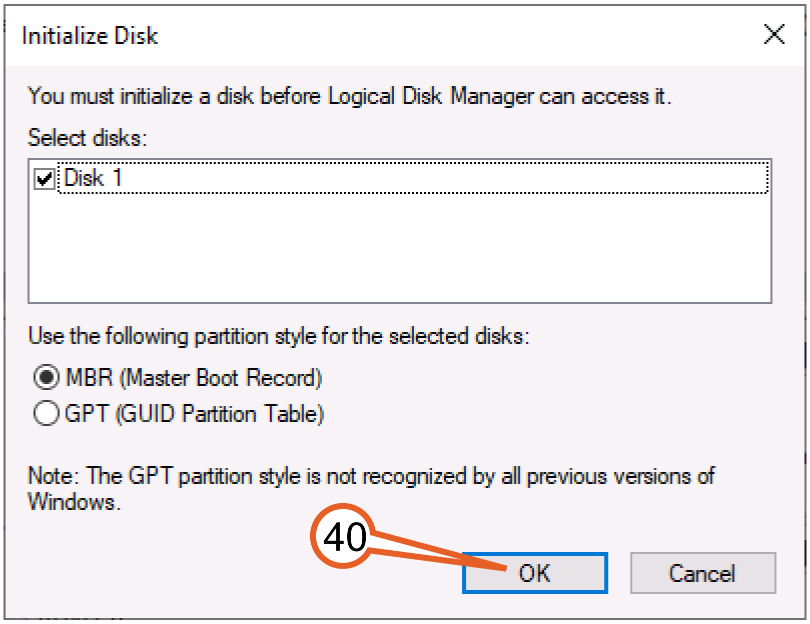


Figure 3-27:

The “Initialize Disk” window closes, and focus returns to the “Disk Management” view in the Computer Management window.

41. The disk now reports a type of “Basic”, and a status of “online”.

42. In the partition area, the disk reports as “Unallocated”.

43. Right-click inside the disk's partition area (where it says **Unallocated**) and select **New Simple Volume…** from the context menu.

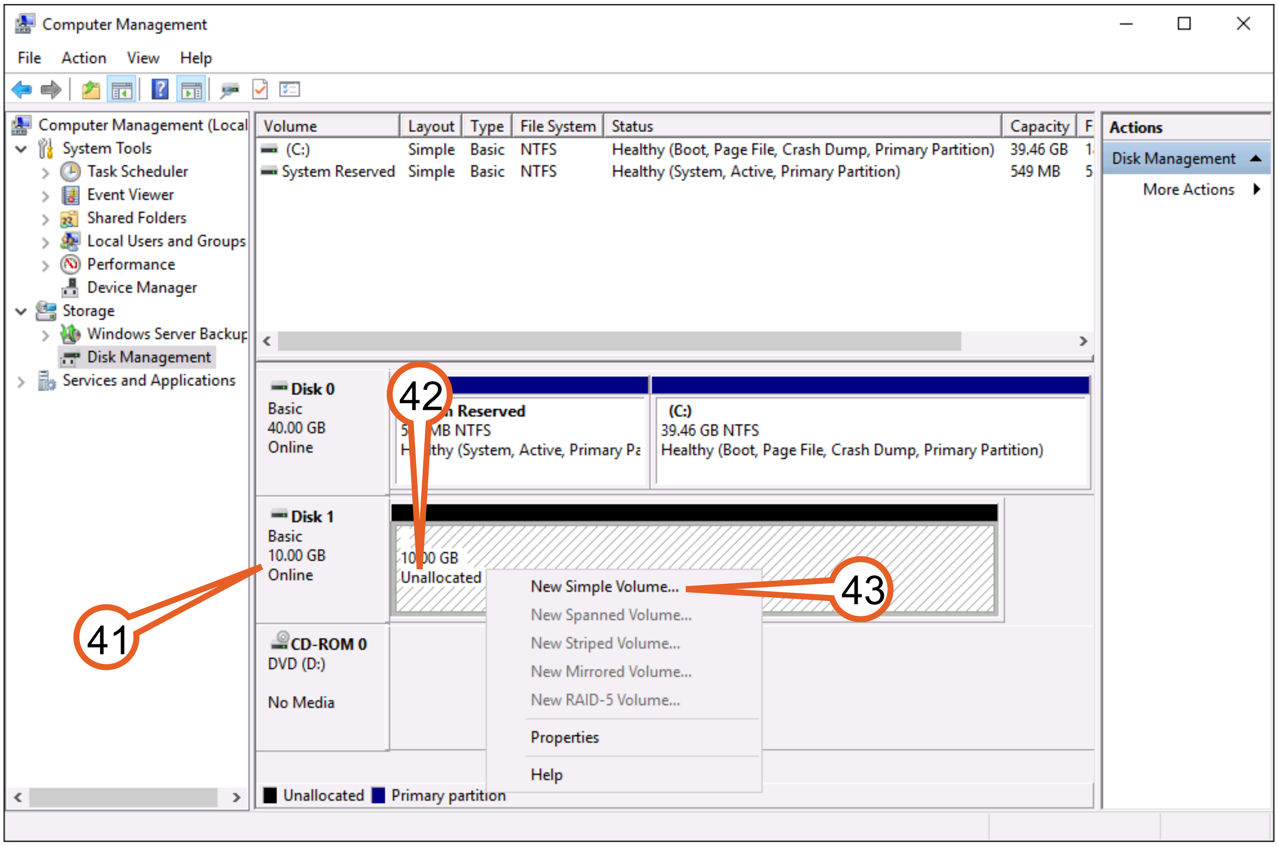


Figure 3-28:

The "New Simple Volume Wizard" window opens.

44. Click the **Next** button to advance the wizard.

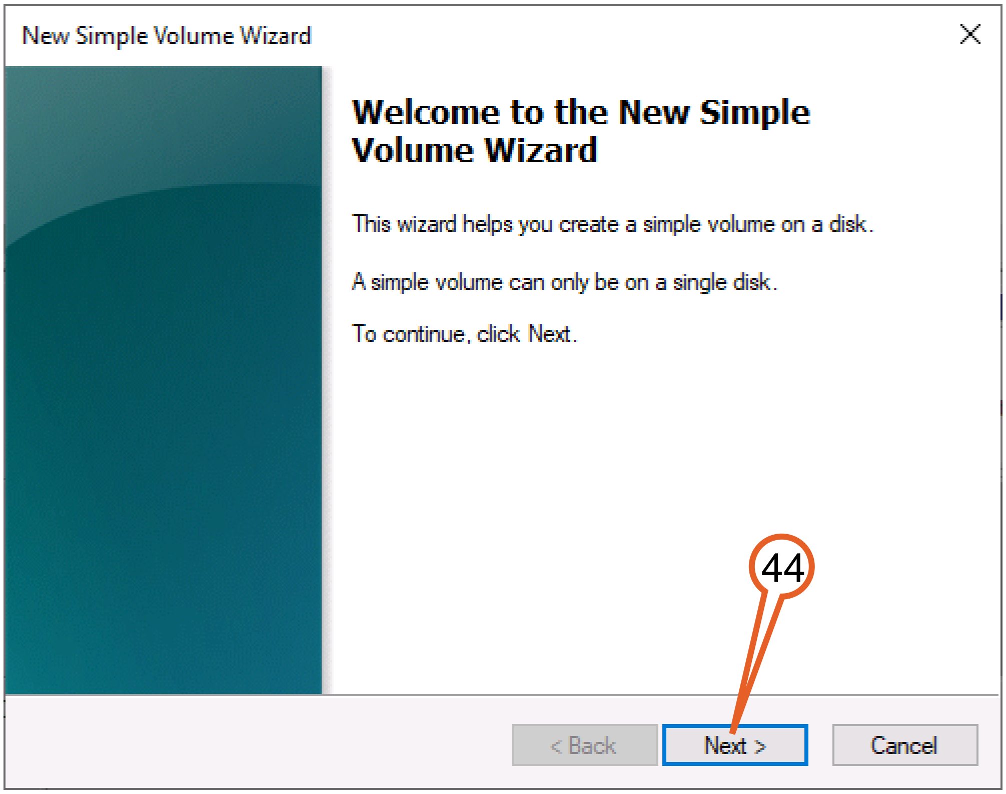


Figure 3-29:

The wizard advances to the “Specify Volume Size” step.

45. The wizard defaults to allocating all of the space in the volume, so click the **Next** button.

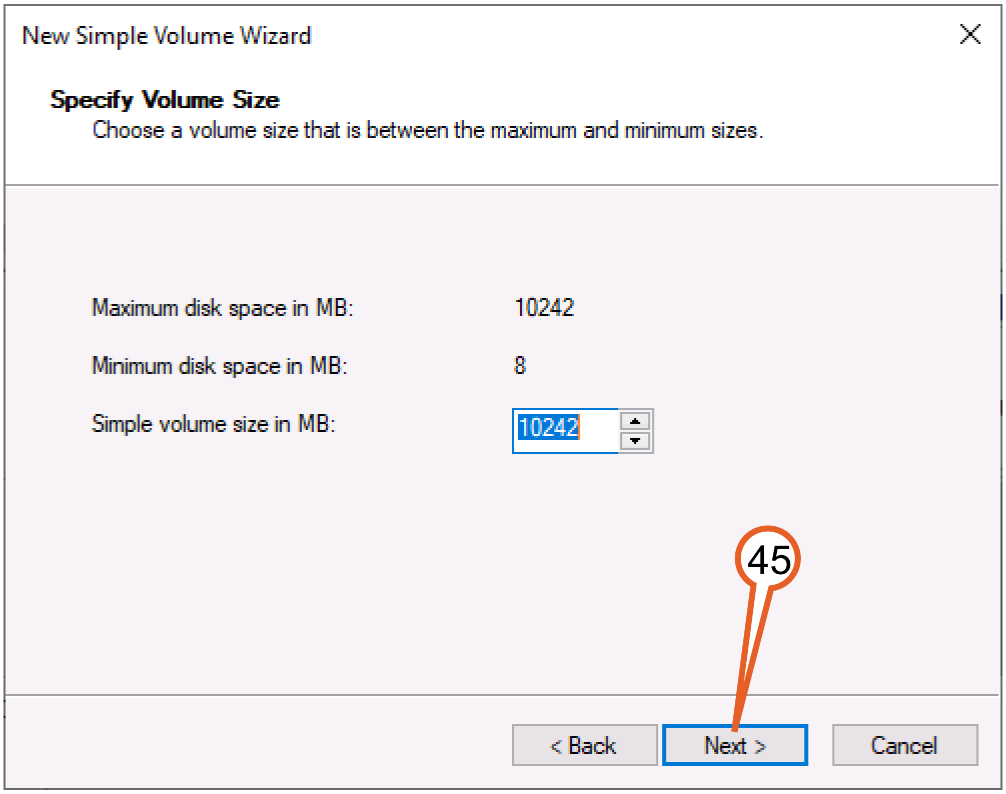


Figure 3-30:

The wizard advances to the “Assign Drive Letter or Path” step.

46. The wizard automatically selects the next available drive letter, which should be **E**.

47. Click **Next**.

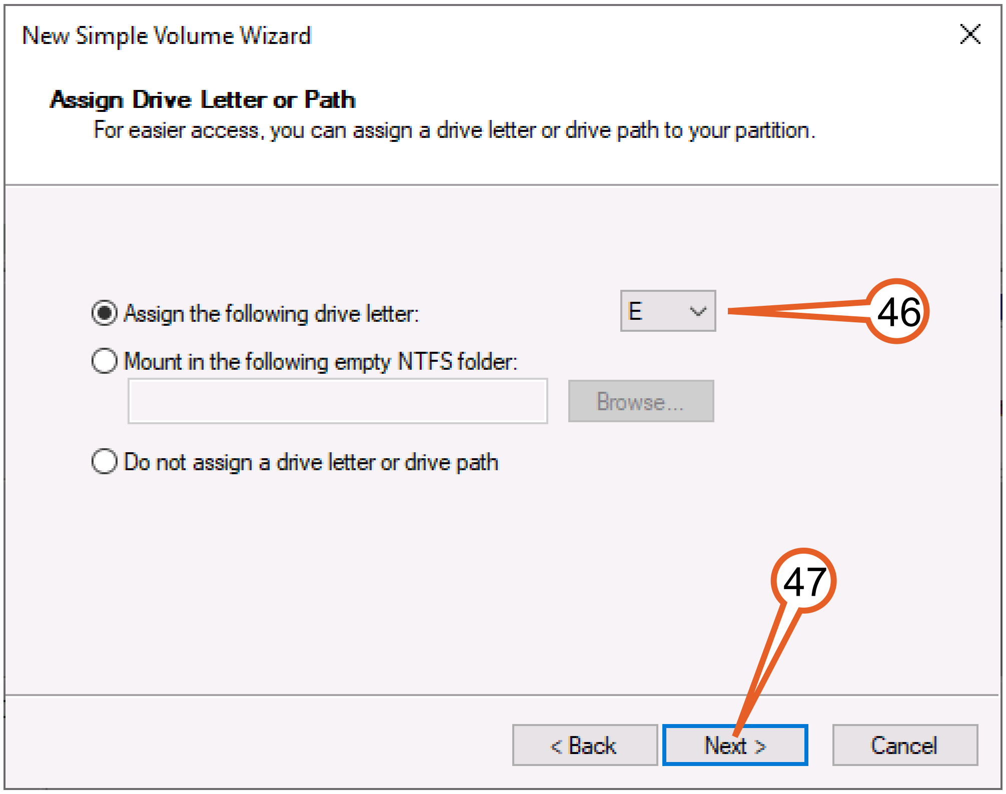


Figure 3-31:

The wizard advances to the “Format Partition” step.

48. Set the “Volume Label” field to **`WINLUN`**.

49. Click **Next**.

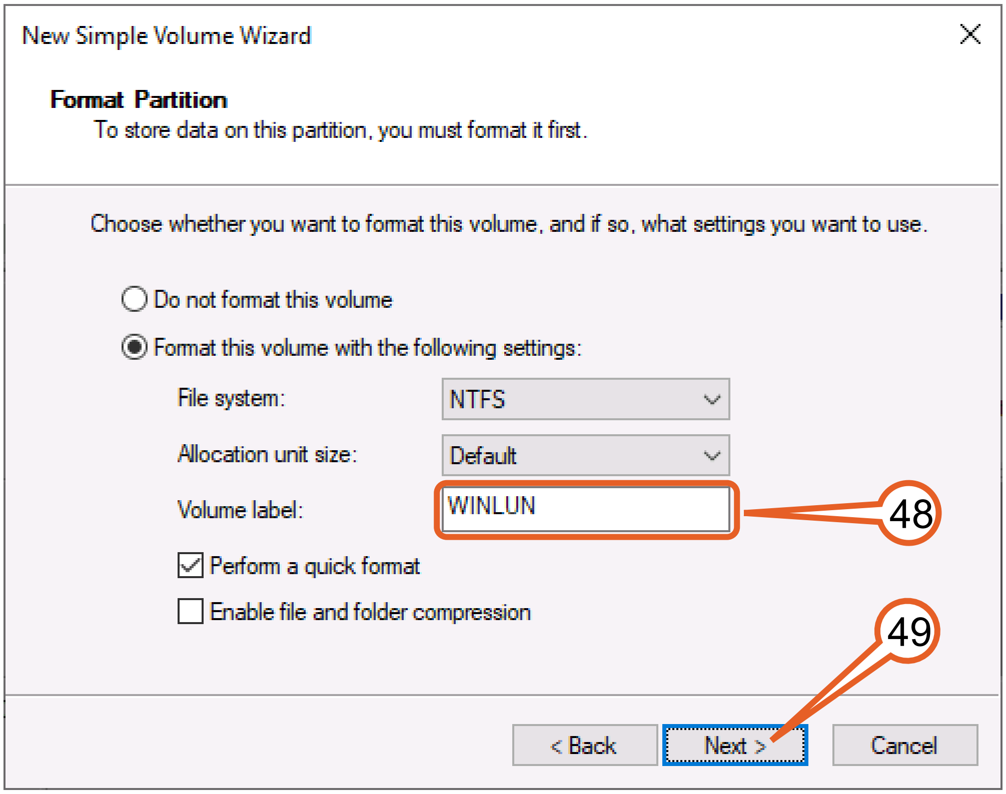


Figure 3-32:

The wizard advances to the “Completing the New Simple Volume Wizard” step.

50. Click **Finish**.

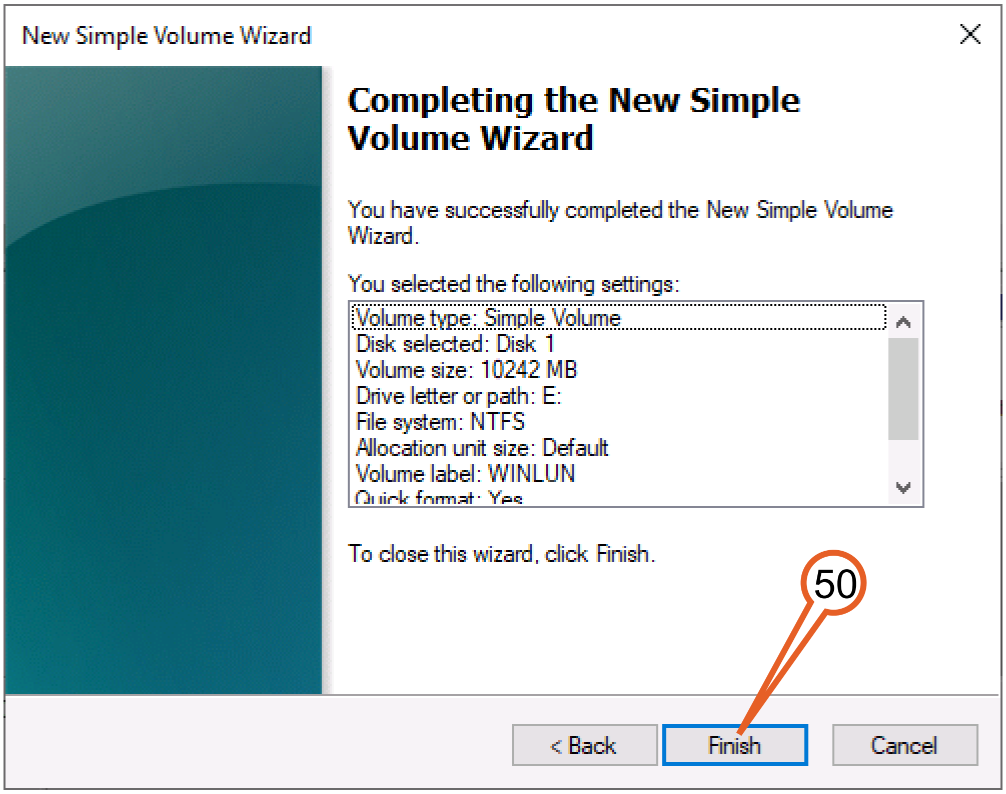
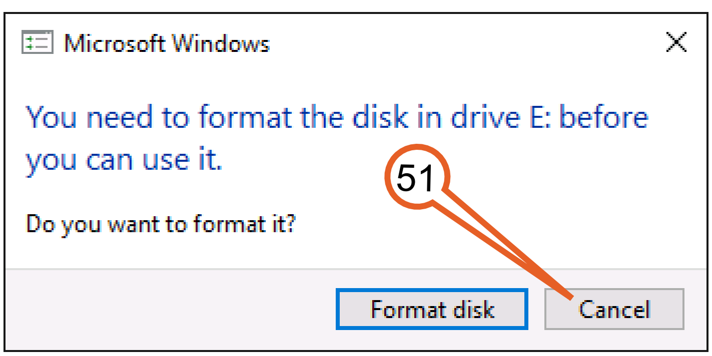


Figure 3-33:

The “New Simple Volume Wizard” window closes, and focus returns to the “Disk Management” view of the Computer Management window.

51. Windows may open a pop-up dialog (which may appear behind other windows that are already open) indicating that you need to format the new disk in drive E:. This is an erroneous warning due to a race condition in Windows. You already formatted the volume during the New Simple Volume wizard, so click **Cancel** to dismiss the dialog. If you do not see this dialog now, you will likely see it later in step 64.

Figure 3-34:



52. The new WINLUN volume displays as “Healthy” in the disk list at the bottom of the Computer Management window, indicating that the new LUN is mounted and ready to use.

53. Before you complete this section of the lab, take a look at the MPIO configuration for this LUN, right-click inside the partition box for the **WINLUN** volume, and select **Properties** from the context menu.

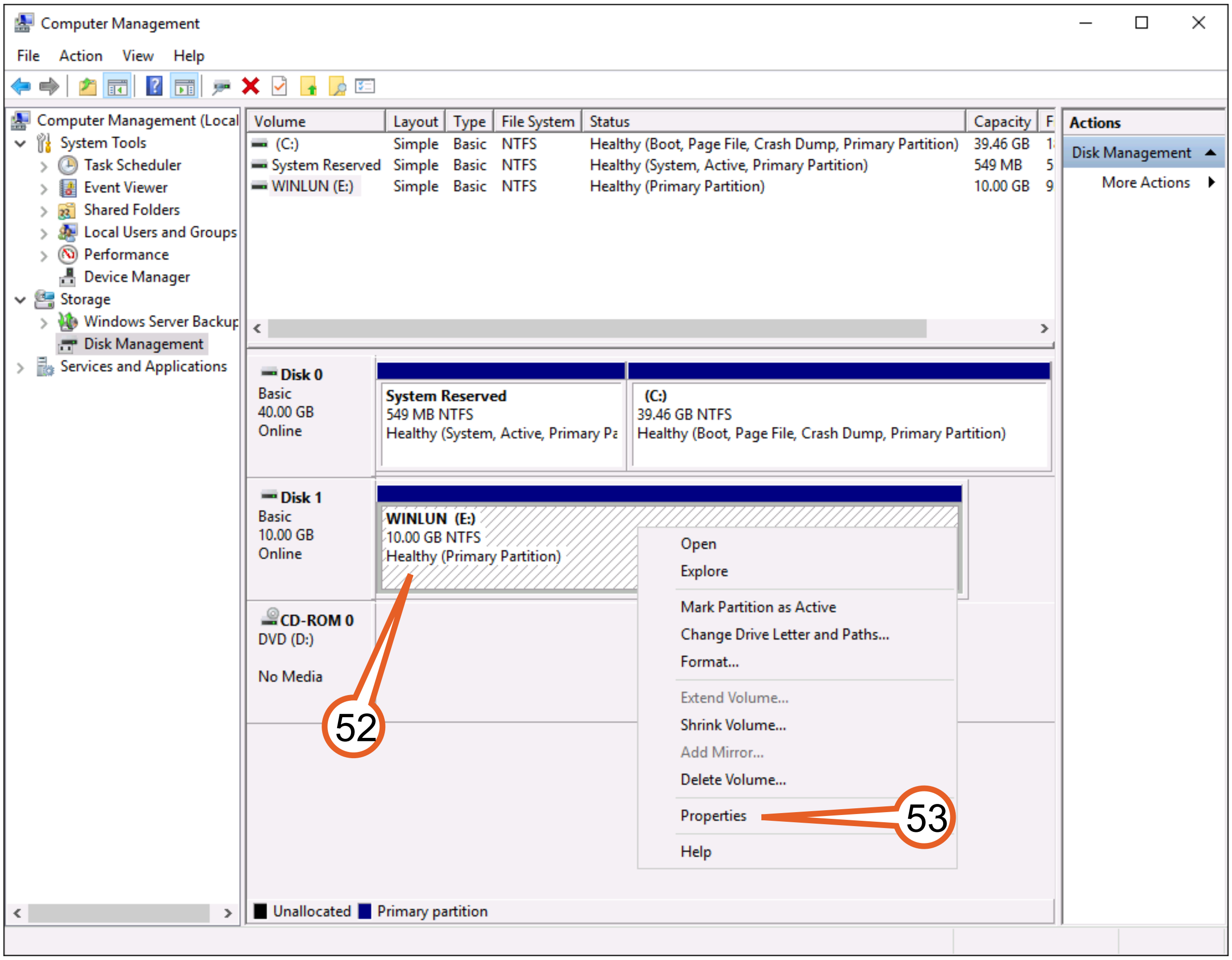


Figure 3-35:

The “WINLUN (E:) Properties” window opens.

54. Click the **Hardware** tab.

55. In the “All disk drives” list, select the **NETAPP LUN C-Mode Multi-Path Disk** entry.

56. Click **Properties**.

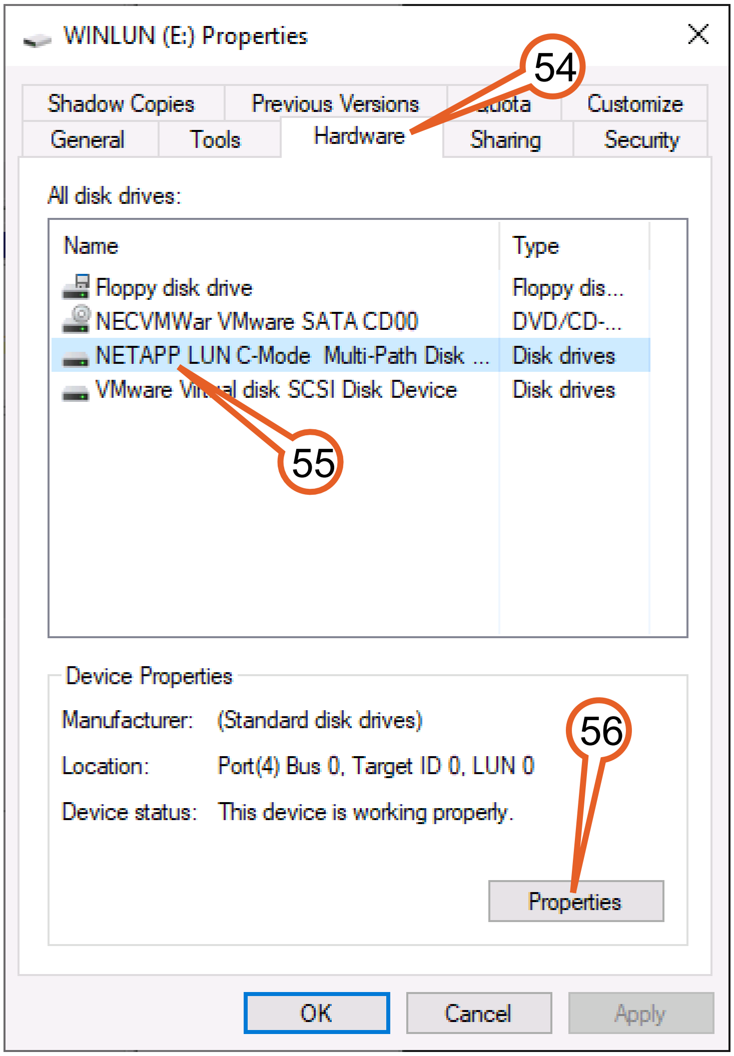


Figure 3-36:

The "NETAPP LUN C-Mode Multi-Path Disk Device Properties" window opens.

57. Click the **MPIO** tab.

58. Notice that you are using the Microsoft DSM for multi-path access.

59. The MPIO policy is set to "Round Robin with Subset". A number of different multi-pathing policies are available, but the configuration shown here sends LUN I/O down alternating active/optimized paths. You can click the **More information about MPIO policies** link at the bottom of the dialog window for details about all the available policies.

60. The top two paths show both a "Path State" and "TPG State" as "Active/Optimized". These paths are connected to the node cluster1-01, and the Least Queue Depth policy makes active use of both paths to this node. Conversely, the bottom two paths show a "Path State" of "Unavailable", and a "TPG State" of "Active/Unoptimized". These paths are connected to the node cluster1-02, and only enter a Path State of "Active/Optimized" if the node cluster1-01 becomes unavailable, or if the volume hosting the LUN migrates over to the node cluster1-02.

61. When you finish reviewing the information in this dialog, click **OK** to exit. If you changed any of the values in this dialog click **Cancel** to discard those changes.

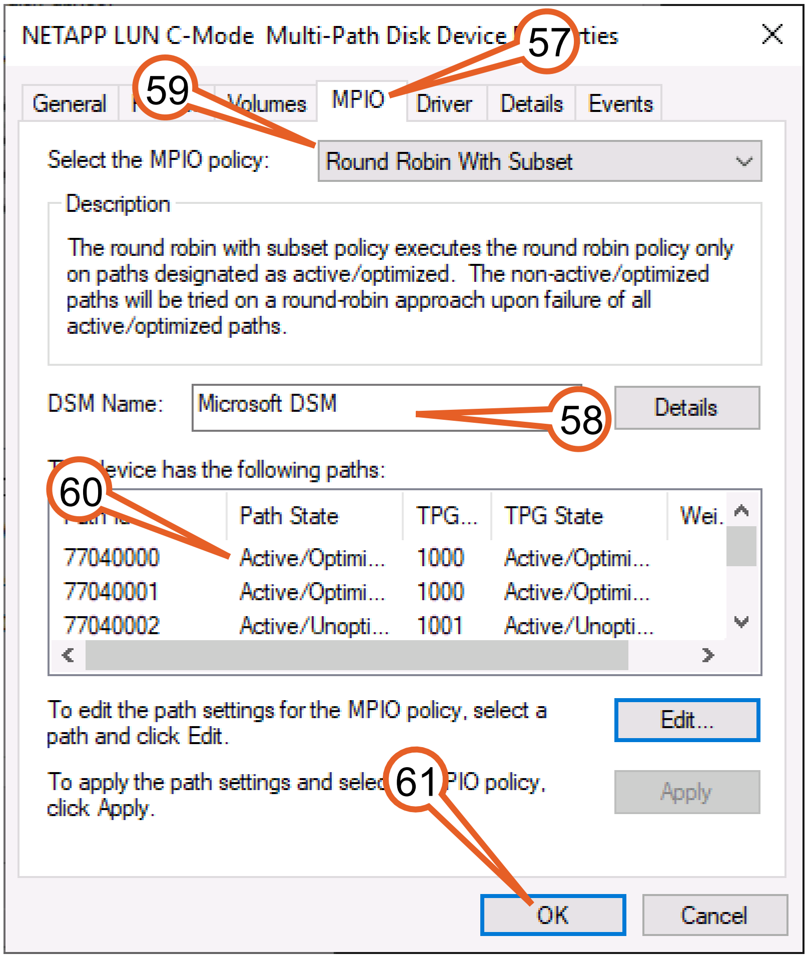


Figure 3-37:

The “NETAPP LUN C-Mode Multi-Path Disk Device Properties” window closes, and focus returns to the “WINLUN (E:) Properties” window.

62. Click **OK**.

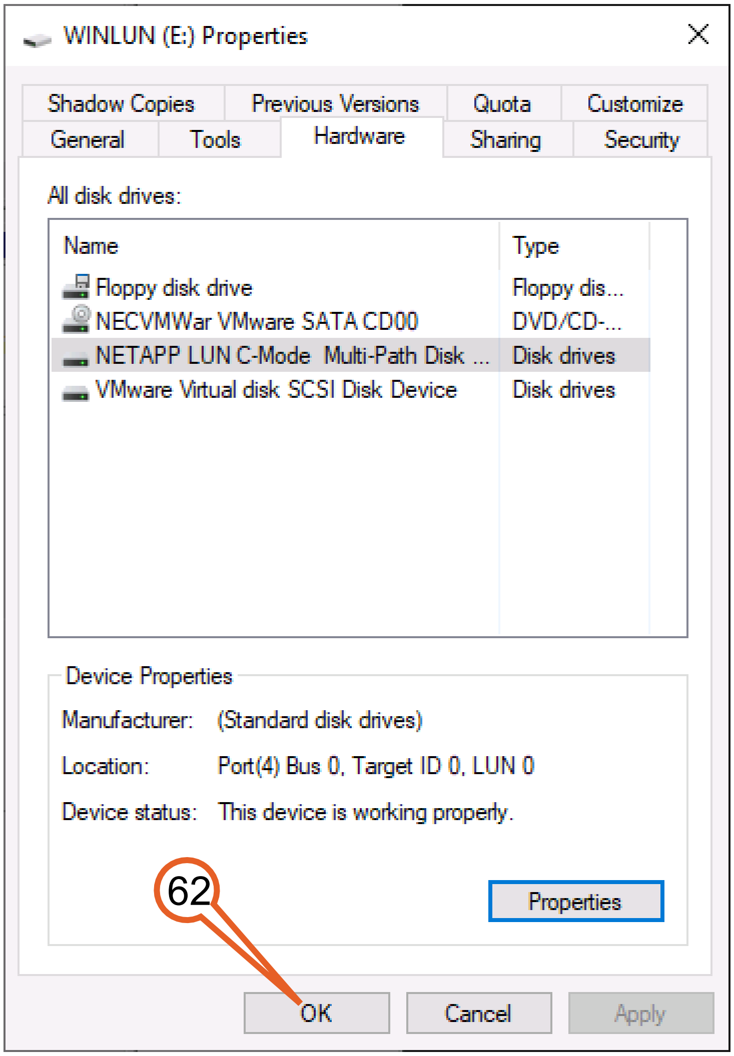


Figure 3-38:

The “WINLUN (E:) Properties” window closes.

63. Close the **Computer Management** window.

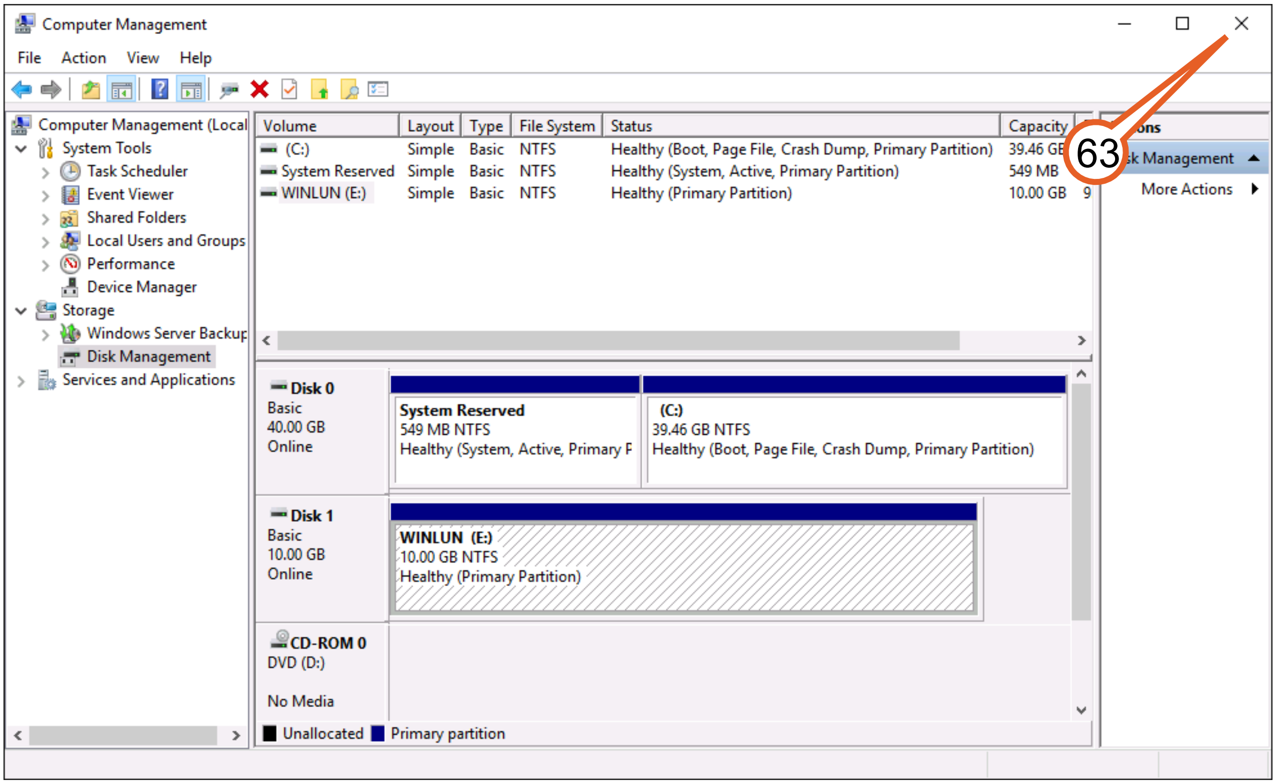


Figure 3-39:

64. If you did not see the pop-up dialog in step 51 from Microsoft Windows stating that you must format the disk in drive E: before you can use it, you may see it now. This window might have been obscured by one of the other windows on the desktop. As you may recall, you did format the LUN during the “New Simple Volume Wizard", meaning this dialog represents an erroneous disk format message. Click **Cancel** to ignore the format request.

Note: Do not close the Administrative tools window yet, as you need to use it again shortly.

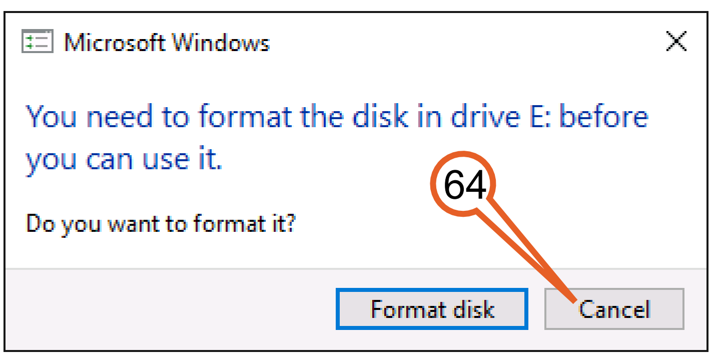


Figure 3-40:

Finally, verify that Windows has detected that the new LUN supports space reclamation. Remember that only Windows 2012 and newer OSs support this feature, and you must minimally have NetApp Windows Unified Host Utilities v6.0.2 (or later) installed. Jumphost meets this criteria.

65. In the “Administrative Tools” window, double-click **Defragment and Optimize drives**.

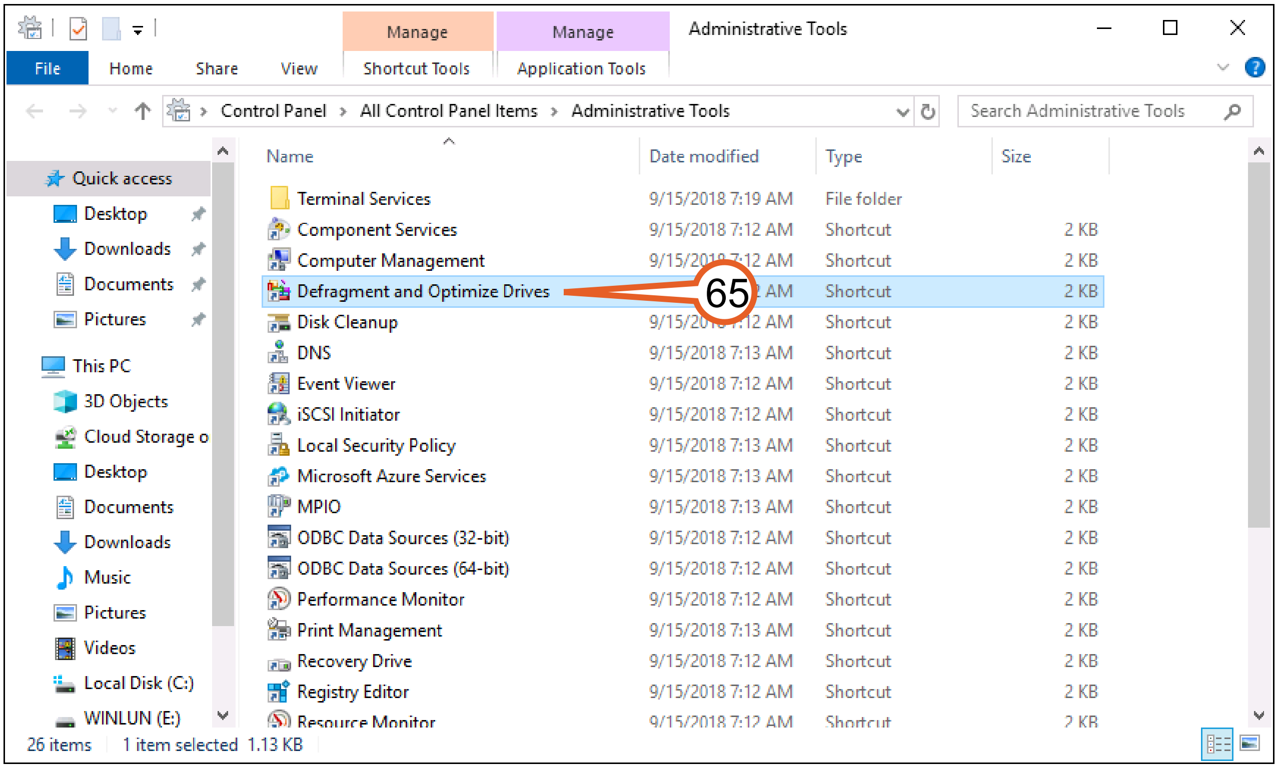


Figure 3-41:

The “Optimize Drives” window opens .

66. Find the **WINLUN (E:)** entry in the drive list and look at its “Media type” value. If that value is “Thin provisioned drive”, then Windows has recognized that this drive supports space reclamation. If that value is “Hard disk drive”, then it does not.

67. Click **Close**.

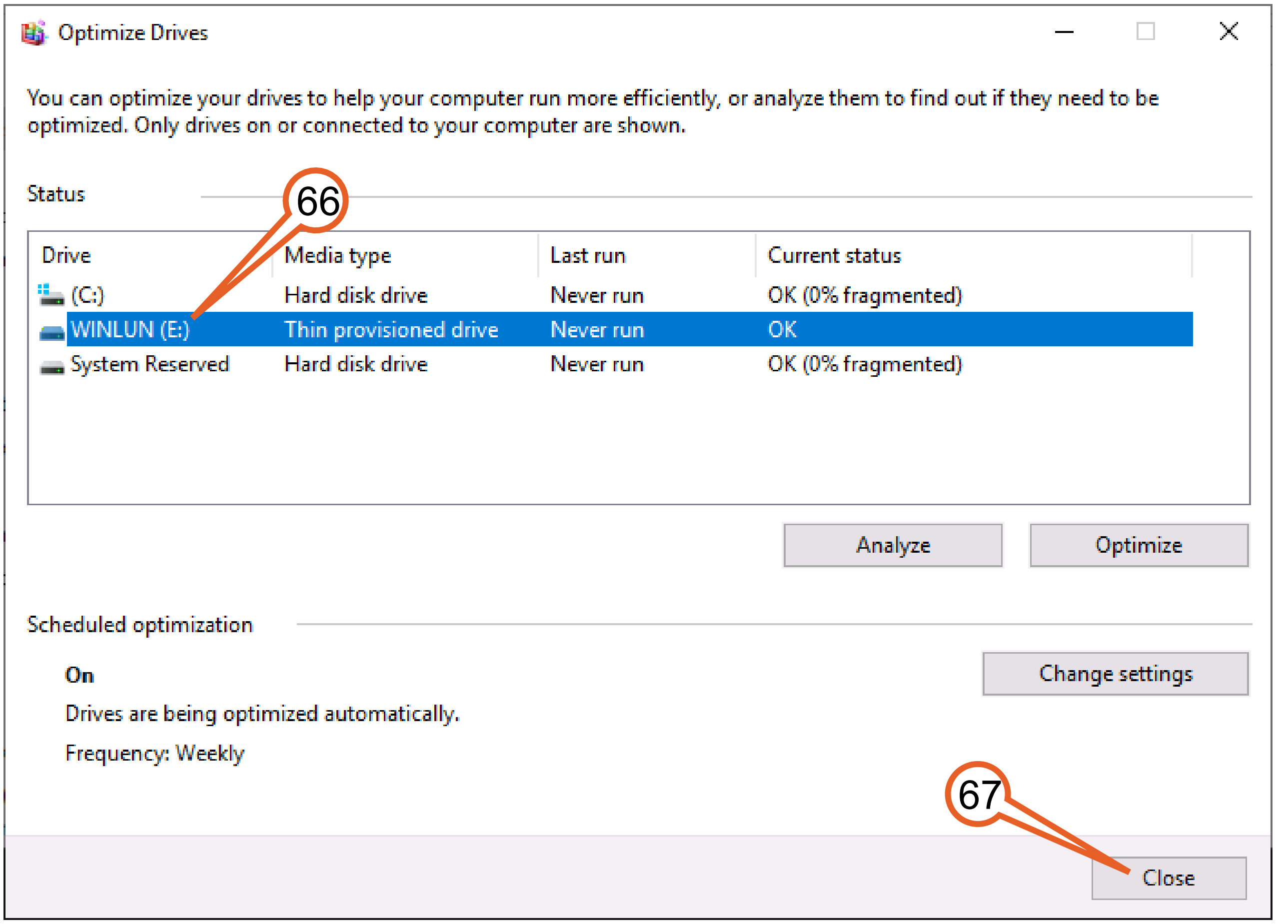


Figure 3-42:

The “Optimize Drives” window closes.

68. Close the **Administrative Tools** window.

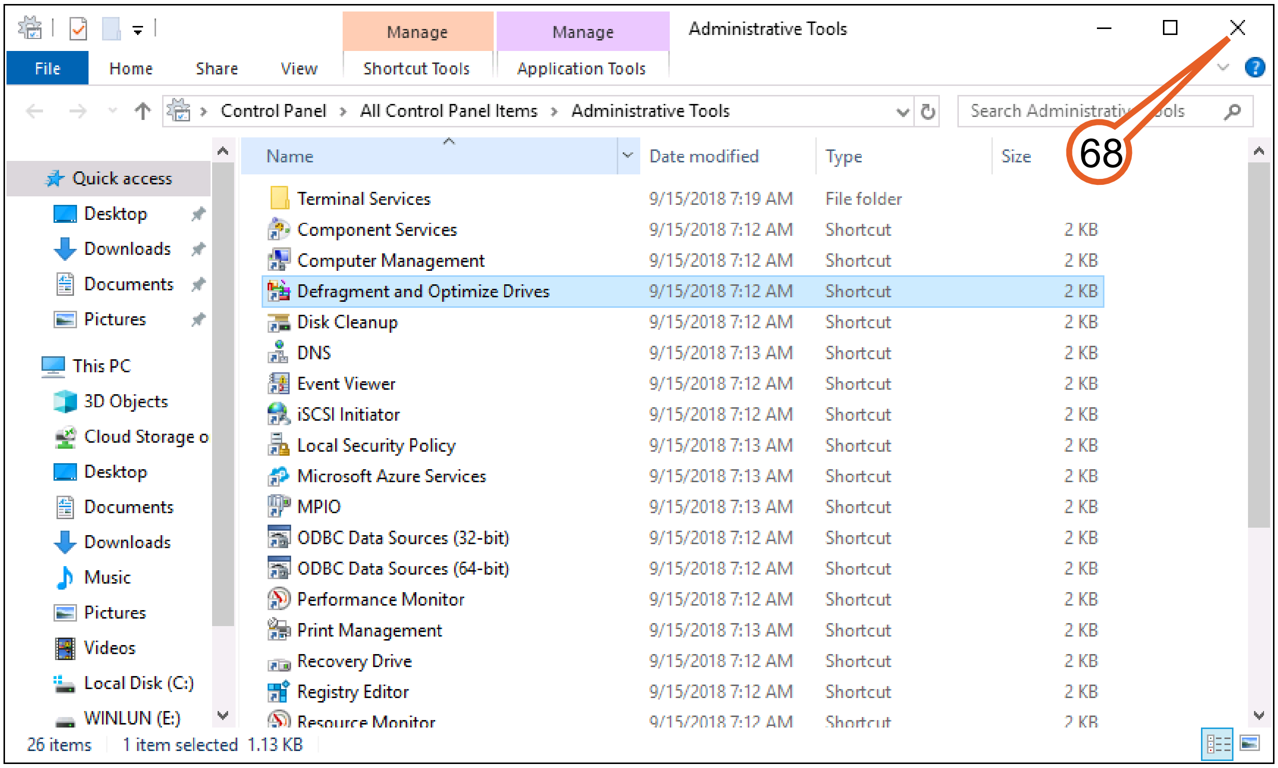


Figure 3-43:

Feel free to open Windows Explorer on Jumphost, and verify that you can create a file on the E: drive.

This completes this exercise.

3.2.3 Create, Map, and Mount a Linux LUN

In an earlier section you created a new SVM, and configured it for iSCSI. In the following sub-sections you will perform the remaining steps needed to configure and use a LUN under Linux:

- Gather the iSCSI Initiator Name of the Linux client.
- Create a thin provisioned Linux volume, create a thin provisioned Linux LUN named “linux.lun” within that volume, and map the LUN to the Linux client.
- Mount the LUN on the Linux client.

The examples in these exercises assume that you have completed the Create, Map, and Mount a Windows LUN” exercise, but completing that exercise is not a preprequisite for this exercise.

3.2.3.1 LUN Concepts

In ONTAP, LUNs are simply large files that reside within a FlexVol. A single FlexVol can support multiple LUNs, and it is quite common for a client that is using multiple LUNs to have all of those LUNs co-located on the same FlexVol. This facilitates taking application-consistent snapshots, and generally makes LUN management related to the client easier.

While it is possible to configure your SVM so that a volume containing a LUN is also accessible through CIFS or NFS, there is usually no advantage to be gained by doing so. Additionally, if you do not carefully manage export rules and file permissions, you can inadvertently expose the LUN file to unintended NAS clients. While it is perfectly acceptable to serve NAS and SAN clients off of the same SVM, best practice is to not mix NAS files and LUNs within the same volume, or to expose LUN volumes over NAS protocols.

An initiator group, or igroup, defines a list of the Fibre Channel WWPNs or iSCSI node names of the SAN clients that are permitted to access (or even see) a set of LUNs. Mapping a LUN to an igroup applies the igroup's access restrictions to the LUN.

Network access to an iSCSI (or FC) LUN occurs over a LIF. Unlike NAS LIFs, that can migrate online from one node to another in a cluster, SAN LIFs cannot be moved while online. Instead, Asymmetric Logical Unit Access (ALUA) provides redundant paths and automatic path selection as part of any ONTAP SAN solution. When there is an I/O interruption on a given LIF, the client simply (and transparently) retries by switching the I/O to another LIF.

Clients utilize multipathing software in conjunction with ALUA to support multiple paths to a LUN, and for path selection. Windows and Linux both include native multipathing software.

NetApp recommends configuring two SAN LIFs on each node that participate in a SAN configuration in order to provide path redundancy. The optimal path for accessing a LUN is usually to connect directly to the node that is hosting the LUN (i.e., the node that is directly connected to the SSDs/HDDs comprising the aggregate on which the volume hosting the LUN resides). In an HA configuration, that includes the node owning the LUN and its HA partner. LIFs on other nodes can also be used to access the LUN, but those paths may suffer a slight performance penalty as data will have to transfer between the involved nodes over the cluster network.

Selective LUN Mapping (SLM) is an ONTAP feature that reduces the number of paths from the host to the LUN. With SLM, when a new LUN map is created, the LUN is accessible only through paths on the node owning the LUN and its HA partner. This is very helpful in reducing complexity for large clusters (with many nodes) that may present many potential paths to a given LUN.

Important: Beginning with ONTAP 9.7, NetApp has started shipping a 2-node AFF All SAN Array option available for factory-ordered and shipped AFF systems. The new AFF All SAN Array (ASA) offers mission-critical workloads (such as databases), a symmetric active-active pathing in a 2-node configuration to offer an even higher level of continuous availability in a simple setup for SAN workloads. While a standard AFF offers a more scalable multi-protocol solution with active-passive (non-optimized) pathing, the AFF ASA SAN system provide a higher level of uninterrupted access for SAN protocols during a planned or unplanned storage failover event.

3.2.3.2 Gather the Linux Client iSCSI Initiator Name

You need to determine the Linux client's iSCSI initiator name so that you can set up an appropriate initiator group to control access to the LUN.

You should already have a PuTTY connection open to the Linux host rhel1. If you do not, then open one now using the instructions found in the "Accessing the Command Line" section at the beginning of this lab guide. The username will be **`root`**, and the password will be **`Netapp1!`**.

1. Change to the directory that hosts the iscsi configuration files.

```
[root@rhel1 ~]# cd /etc/iscsi
[root@rhel1 iscsi]# ls
initiatorname.iscsi  iscsid.conf
[root@rhel1 iscsi]#
```

2. Display the name of the iscsi initiator.

```
[root@rhel1 iscsi] cat initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:rhel1.demo.netapp.com
[root@rhel1 iscsi]#
```

Important: The initiator name for rhel1 is **iqn.1994-05.com.redhat:rhel1.demo.netapp.com**.

3.2.3.3 Create and Map a Linux LUN

In this activity, you create a new thin provisioned Linux LUN on the SVM "svmluns" under the volume "linluns", and also create an initiator igroup for the LUN so that only the Linux host rhel1 can access it. An initiator group, or igroup, defines a list of the Fibre Channel WWPNs or iSCSI node names for the hosts that are permitted to see the associated LUNs.

1. If you do not currently have a PuTTY session open to cluster1 then open one now following the instructions from 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 thin provisioned volume “linluns” that will host the Linux LUN you will create in a later step.

```
cluster1::> volume create -vserver svmluns -volume linluns -aggregate aggr1_cluster1_01
 -size 10.31GB -percent-snapshot-space 0 -snapshot-policy none -space-guarantee none
 -autosize-mode grow -nvfail on -tiering-policy none
[Job 271] Job is queued: Create linluns.
[Job 271] Job succeeded: Successful
cluster1::>
```

3. Display the volume list.

```
cluster1::> volume show
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
cluster1-01
          vol0         aggr0_cluster1_01
                                    online     RW      23.01GB    20.04GB    8%
cluster1-02
          vol0         aggr0_cluster1_02
                                    online     RW      23.01GB    20.28GB    7%
svm1      eng_users    aggr1_cluster1_01
                                    online     RW         10GB     9.50GB    5%
svm1      engineering  aggr1_cluster1_01
                                    online     RW         10GB     9.50GB    5%
svm1      svm1_root    aggr1_cluster1_01
                                    online     RW         20MB    18.85MB    5%
svmluns   linluns      aggr1_cluster1_01
                                    online     RW      10.31GB    10.31GB    0%
svmluns   svmluns_root aggr1_cluster1_01
                                    online     RW         20MB    18.86MB    5%
svmluns   winluns      aggr1_cluster1_01
                                    online     RW      10.31GB    10.28GB    0%
8 entries were displayed.
cluster1::>
```

4. Display a list of the LUNs on the cluster.

```
cluster1::> lun show
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------- ------- -------- -------- --------
svmluns   /vol/winluns/windows.lun        online  mapped   windows_2008
                                                                     10.00GB
cluster1::>
```

5. Create the thin provisioned Linux LUN “linux.lun” on the volume “linluns”.

```
cluster1::> lun create -vserver svmluns -volume linluns -lun linux.lun -size 10GB
 -ostype linux -space-reserve disabled
Created a LUN of size 10g  (10742215680)
cluster1::>
```

6. Add a comment to the LUN “linux.lun”.

```
cluster1::> lun modify -vserver svmluns -volume linluns -lun linux.lun
 -comment "Linux LUN"
cluster1::>
```

7. Display the list of LUNs.

```
cluster1::> lun show
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------- ------- -------- -------- --------
svmluns   /vol/linluns/linux.lun          online  unmapped linux        10GB
svmluns   /vol/winluns/windows.lun        online  mapped   windows_2008
                                                                     10.00GB
2 entries were displayed.
cluster1::>
```

8. Display a list of the cluster's igroups.

```
cluster1::> igroup show
Vserver   Igroup       Protocol OS Type  Initiators
--------- ------------ -------- -------- ------------------------------------
svmluns   winigrp      iscsi    windows  iqn.1991-05.com.microsoft:Jumphost.
                                         demo.netapp.com
cluster1::>
```

9. Create a new igroup named “linigrp” that grants rhel1 access to the LUN “linux.lun”.

```
cluster1::> igroup create -vserver svmluns -igroup linigrp -protocol iscsi
 -ostype linux -initiator iqn.1994-05.com.redhat:rhel1.demo.netapp.com
cluster1::>
```

10. Display a list of the igroups.

```
cluster1::> igroup show
Vserver   Igroup       Protocol OS Type  Initiators
--------- ------------ -------- -------- ------------------------------------
svmluns   linigrp      iscsi    linux    iqn.1994-05.com.redhat:rhel1.demo.
                                         netapp.com
svmluns   winigrp      iscsi    windows  iqn.1991-05.com.microsoft:Jumphost.
                                         demo.netapp.com
2 entries were displayed.
cluster1::>
```

11. Map the LUN “linux.lun” to the igroup “linigrp”.

```
cluster1::> lun map -vserver svmluns -volume linluns -lun linux.lun -igroup linigrp
cluster1::>
```

12. Display a list of the LUNs.

```
cluster1::> lun show
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------- ------- -------- -------- --------
svmluns   /vol/linluns/linux.lun          online  mapped   linux        10GB
svmluns   /vol/winluns/windows.lun        online  mapped   windows_2008
                                                                     10.00GB
2 entries were displayed.
cluster1::>
```

13. Display a list of the LUN mappings.

```
cluster1::> lun mapped show
Vserver    Path                                      Igroup   LUN ID  Protocol
---------- ----------------------------------------  -------  ------  --------
svmluns    /vol/linluns/linux.lun                    linigrp       0  iscsi
svmluns    /vol/winluns/windows.lun                  winigrp       0  iscsi
2 entries were displayed.
cluster1::>
```

14. Display just the “linux.lun” LUN.

```
cluster1::> lun show -lun linux.lun
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------- ------- -------- -------- --------
svmluns   /vol/linluns/linux.lun          online  mapped   linux        10GB
cluster1::>
```

15. Display LUN mappings for just “linux.lun”.

```
cluster1::> lun mapped show -lun linux.lun
Vserver    Path                                      Igroup   LUN ID  Protocol
---------- ----------------------------------------  -------  ------  --------
svmluns    /vol/linluns/linux.lun                    linigrp       0  iscsi
cluster1::>
```

16. Display detailed LUN mapping information for “linux.lun”.

```
cluster1::> lun show -lun linux.lun -instance

                  Vserver Name: svmluns
                      LUN Path: /vol/linluns/linux.lun
```

```
                    Volume Name: linluns
                     Qtree Name: ""
                       LUN Name: linux.lun
                       LUN Size: 10GB
                        OS Type: linux
              Space Reservation: disabled
                  Serial Number: wOj6b]QPl/dU
            Serial Number (Hex): 774f6a36625d51506c2f6455
                        Comment: Linux LUN
     Space Reservations Honored: false
               Space Allocation: disabled
                          State: online
                       LUN UUID: c6035ef1-356a-4d69-8344-626e06cac077
                         Mapped: mapped
Physical Size of Logical Block: 512B
               Device Legacy ID: -
               Device Binary ID: -
                 Device Text ID: -
                      Read Only: false
          Fenced Due to Restore: false
                      Used Size: 0
            Maximum Resize Size: 128.0GB
                  Creation Time: 12/18/2020 00:16:45
                          Class: regular
           Node Hosting the LUN: cluster1-01
               QoS Policy Group: -
      QoS Adaptive Policy Group: -
            Caching Policy Name: -
                          Clone: false
       Clone Autodelete Enabled: false
            Inconsistent Import: false
                    Application: -

cluster1::>
```

ONTAP supports a space reclamation feature that allows it to reclaim space from a thin provisioned LUN when the client deletes data from the LUN, and also allows ONTAP to notify the client when the LUN cannot accept writes due to lack of space on the volume. This feature is supported by VMware ESX 5.0 and later, Red Hat Enterprise Linux 6.2 and later, and Microsoft Windows 2012 and later. The RHEL clients used in this lab are running version 7.5, so you will enable the space reclamation feature for your Linux LUN.

17. Display the space reclamation setting for the “linux.lun” LUN.

```
cluster1::> lun show -vserver svmluns -path /vol/linluns/linux.lun -fields space-allocation
vserver path                   space-allocation
------- ---------------------- ----------------
svmluns /vol/linluns/linux.lun disabled
cluster1::>
```

18. Configure the LUN “linux.lun” to support space reclamation.

```
cluster1::> lun modify -vserver svmluns -path /vol/linluns/linux.lun -space-allocation
 enabled
cluster1::>
```

19. Display the new space reclamation setting for the LUN “linux.lun”.

```
 lun show -vserver svmluns -path /vol/linluns/linux.lun -fields space-allocation
vserver path                   space-allocation
------- ---------------------- ----------------
svmluns /vol/linluns/linux.lun enabled
cluster1::>
```

3.2.3.4 Mount the LUN on a Linux Client

In this activity you use the Linux command line to configure the host rhel1 to connect to the Linux LUN /vol/linluns/linux.lun you created in the preceding section.

This section assumes that you know how to use the Linux command line. If you are not familiar with these concepts, we recommend that you skip this section of the lab.

1. If you do not currently have a PuTTY session open to rhel1, open one now and log in as user **root** with the password **"Netapp1!"**.
2. The NetApp Linux Unified Host Utilities kit has been pre-installed on both Red Hat Linux hosts in this lab, and the iSCSI initiator name has already been configured for each host. Confirm that is the case:

```
[root@rhel1 ~]# cd
[root@rhel1 ~]# rpm -qa | grep netapp
netapp_linux_unified_host_utilities-7-1.x86_64
[root@rhel1 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:rhel1.demo.netapp.com
[root@rhel1 ~]#
```

3. In the /etc/iscsi/iscsid.conf file the node.session.timeo.replacement_timeout value is set to **5** to better support timely path failover, and the node.startup value is set to **automatic** so that the system will automatically log in to the iSCSI node at startup.

```
[root@rhel1 ~]# grep replacement_time /etc/iscsi/iscsid.conf
#node.session.timeo.replacement_timeout = 120
node.session.timeo.replacement_timeout = 5
[root@rhel1 ~]# grep node.startup /etc/iscsi/iscsid.conf
# node.startup = automatic
node.startup = automatic
[root@rhel1 ~]#
```

4. You will find that the Red Hat Linux hosts in the lab have pre-installed DM-Multipath packages and a /etc/multipath.conf file pre-configured to support multi-pathing, so that the RHEL host can access the LUN using all of the SAN LIFs you created for the svmluns SVM.

```
[root@rhel1 ~]# rpm -q device-mapper
device-mapper-1.02.146-4.el7.x86_64
[root@rhel1 ~]# rpm -q device-mapper-multipath
device-mapper-multipath-0.4.9-119.el7.x86_64
[root@rhel1 ~]# cat /etc/multipath.conf
# For a complete list of the default configuration values, see
# /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.defaults
# For a list of configuration options with descriptions, see
# /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.annotated
#
# REMEMBER: After updating multipath.conf, you must run
#
# service multipathd reload
#
# for the changes to take effect in multipathd

# NetApp recommended defaults
defaults {
 flush_on_last_del yes
 max_fds     max
 queue_without_daemon    no
 user_friendly_names     no
 dev_loss_tmo            infinity
 fast_io_fail_tmo        5
}

blacklist {
 devnode "^sda"
 devnode "^hd[a-z]"
 devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
 devnode "^ccis.*"
}

devices {
        # NetApp iSCSI LUNs
 device {
  vendor                    "NETAPP"
  product                   "LUN"
  path_grouping_policy      group_by_prio
  features                  "3 queue_if_no_path pg_init_retries 50"
  prio                      "alua"
  path_checker              tur
  failback                  immediate
  path_selector             "round-robin 0"
  hardware_handler          "1 alua"
  rr_weight                 uniform
  rr_min_io                 128
```

```
  getuid_callout             "/lib/udev/scsi_id -g -u -d /dev/%n"
 }
}
[root@rhel1 ~]#
```

5. You now need to start the iSCSI software service on rhel1, and configure it to start automatically at boot time. Note that a force-start is only necessary the very first time you start the iscsid service on host.

```
[root@rhel1 ~]# systemctl status iscsid.service
# iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:iscsid(8)
           man:iscsiadm(8)
[root@rhel1 ~]# systemctl start iscsid
[root@rhel1 ~]# systemctl enable iscsid
Created symlink from /etc/systemd/system/multi-user.target.wants/iscsid.service to /usr/lib/
systemd/system/iscsid.service.
[root@rhel1 ~]#  systemctl status iscsid.service
#m iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled; vendor preset: disabled)
   Active: active (running) since Apr 2019-04-09 17:46:56 UTC; 12s ago
     Docs: man:iscsid(8)
           man:iscsiadm(8)
 Main PID: 30908 (iscsid)
   CGroup: /system.slice/iscsid.service
           ##30907 /usr/sbin/iscsid
           ##30908 /usr/sbin/iscsid

Apr 09 17:46:56 rhel1 systemd[1]: Starting Open-iSCSI...
Apr 09 17:46:56 rhel1 iscsid[30906]:
Apr 09 17:46:56 rhel1 systemd[1]: Failed to read PID from file /var/run/isc...nt
Apr 09 17:46:56 rhel1 systemd[1]: Started Open-iSCSI.
Apr 09 17:46:56 rhel1 iscsid[30907]: iSCSI daemon with pid=30908 started!
Hint: Some lines were ellipsized, use -l to show in full.
[root@rhel1 ~]#
```

6. Next discover the available targets using the `iscsiadm` command. Note that the exact values used for the node paths may differ in your lab from what is shown in this example, and that after running this command there will still not be active iSCSI sessions because you have not yet created the necessary device files.

```
[root@rhel1 ~]# iscsiadm --mode discovery --op update --type sendtargets
--portal 192.168.0.133
192.168.0.133:3260,1028 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5
192.168.0.136:3260,1031 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5
192.168.0.135:3260,1030 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5
192.168.0.134:3260,1029 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5
[root@rhel1 ~]# iscsiadm --mode session
iscsiadm: No active sessions.
[root@rhel1 ~]#
```

7. Create the devices necessary to support the discovered nodes, after which the sessions become active.

```
[root@rhel1 ~]# iscsiadm --mode node -l all
Logging in to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.133,3260]
 (multiple)
Logging in to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.136,3260]
 (multiple)
Logging in to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.135,3260]
 (multiple)
Logging in to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.134,3260]
 (multiple)
Login to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.133,3260]
 successful.
Login to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.136,3260]
 successful.
Login to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.135,3260]
 successful.
```

```
Login to [iface: default, target:
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5, portal: 192.168.0.134,3260]
 successful.
[root@rhel1 ~]# iscsiadm --mode session
tcp: [1] 192.168.0.133:3260,1028
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5 (non-flash)
tcp: [2] 192.168.0.136:3260,1031
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5 (non-flash)
tcp: [3] 192.168.0.135:3260,1030
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5 (non-flash)
tcp: [4] 192.168.0.134:3260,1029
 iqn.1992-08.com.netapp:sn.85f24836beaf11e8ba95005056b048e1:vs.5 (non-flash)
[root@rhel1 ~]#
```

8. At this point the Linux client sees the LUN over all four paths, but it does not yet understand that all four paths represent the same LUN.

```
[root@rhel1 ~]# sanlun lun show
controller(7mode/E-Series)/                                  device          host
     lun
vserver(cDOT/FlashRay)        lun-pathname                   filename        adapter
 protocol   size    product
-----------------------------------------------------------------------------------------------------
svmluns                       /vol/linluns/linux.lun         /dev/sdd        host5      iSCSI
     10g     cDOT
svmluns                       /vol/linluns/linux.lun         /dev/sde        host6      iSCSI
     10g     cDOT
svmluns                       /vol/linluns/linux.lun         /dev/sdc        host4      iSCSI
     10g     cDOT
svmluns                       /vol/linluns/linux.lun         /dev/sdb        host3      iSCSI
     10g     cDOT
[root@rhel1 ~]#
```

9. Verify that the multipathd service is enabled and running.

```
[root@rhel1 ~]# systemctl status multipathd
# multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled; vendor preset:
 enabled)
   Active: active (running) since Tue 2020-12-15 06:008:24 UTC; 2 days ago
 Main PID: 535 (multipathd)
   CGroup: /system.slice/multipathd.service
           ##535 /sbin/multipathd

Dec 18 00:26:23 rhel1 multipathd[535]: sde: add path (uevent)
Dec 18 00:26:23 rhel1 multipathd[535]: 3600a0980774f6a36625d4d4f7775564d: load table [0
 20971520 multipath 4 qu...:64 1]
Dec 18 00:26:23 rhel1 multipathd[535]: 3600a0980774f6a36625d4d4f7775564d: event checker
 started
Dec 18 00:26:23 rhel1 multipathd[535]: sde [8:64]: path added to devmap
 3600a0980774f6a36625d4d4f7775564dDec 18 00:26:23 rhel1 multipathd[535]: sdb: add path
 (uevent)
Dec 18 00:26:23 rhel1 multipathd[535]: sdc: add path (uevent)
Dec 18 00:26:23 rhel1 multipathd[535]: sdd: add path (uevent)
Dec 18 00:26:23 rhel1 multipathd[535]: 3600a0980774f6a36625d4d4f7775564d: load table [0
 20971520 multipath 4 qu...:48 1]
Warning: Journal has been rotated since unit was started. Log output is incomplete or
 unavailable.
Hint: Some lines were ellipsized, use -l to show in full.
[root@rhel1 ~]#
```

Tip: If multipathd was not running, then you could enable and start it by issuing the `systemctl enable multipathd` and `systemctl start multipathd` commands.

10. The `multipath` command displays the configuration of DM-Multipath, and the `multipath -ll` command displays a list of the multipath devices. DM-Multipath maintains a device file under /dev/mapper that you use to access the multipathed LUN (in order to create a filesystem on it and to mount it). The first line of output from the `multipath -ll` command lists the name of that device file (in this example “3600a0980774f6a36625d4d4f7775564d”). The autogenerated name for this device file will likely differ in your copy of the lab. Also pay attention to the output of the `sanlun lun show -p` command which shows information about the ONTAP path of the LUN, the LUN's size, its device file name under /dev/mapper, the multipath policy, and also information about the various device paths themselves.

```
[root@rhel1 ~]# multipath -ll
Dec 18 00:26:45 | /etc/multipath.conf line 43, invalid keyword: getuid_callout
```

```
3600a0980774f6a36625d4d4f7775564d dm-3 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle'
 hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 6:0:0:0 sde 8:64 active ready running
| `- 3:0:0:0 sdb 8:16 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 4:0:0:0 sdc 8:32 active ready running
  `- 5:0:0:0 sdd 8:48 active ready running
[root@rhel1 ~]# ls -l /dev/mapper
total 0
lrwxrwxrwx 1 root root       7 Dec 18 00:26 3600a0980774f6a36625d4d4f7775564d -> ../dm-3
crw------- 1 root root 10, 236 Dec 15 06:08 control
lrwxrwxrwx 1 root root       7 Dec 15 06:08 docker-253:1-998953-pool -> ../dm-2
lrwxrwxrwx 1 root root       7 Dec 15 06:08 rhel-root -> ../dm-1
lrwxrwxrwx 1 root root       7 Dec 15 06:08 rhel-swap -> ../dm-0
[root@rhel1 ~]# sanlun lun show -p

                    ONTAP Path: svmluns:/vol/linluns/linux.lun
                           LUN: 0
                      LUN Size: 10g
                       Product: cDOT
                   Host Device: 3600a0980774f6a36625d4d4f7775564d
              Multipath Policy: round-robin 0
            Multipath Provider: Native
--------- ---------- ------- ------------ ----------------------------------------------
host      vserver
path      path       /dev/   host         vserver
state     type       node    adapter      LIF
--------- ---------- ------- ------------ ----------------------------------------------
up        primary    sde     host6        cluster1-01_iscsi_lif_2
up        primary    sdb     host3        cluster1-01_iscsi_lif_1
up        secondary  sdc     host4        cluster1-02_iscsi_lif_2
up        secondary  sdd     host5        cluster1-02_iscsi_lif_1
[root@rhel1 ~]#
```

Tip: You can see even more detail about the configuration of multipath and the LUN as a whole by issuing the `multipath -v3 -d -ll` or `iscsiadm -m session -P 3` commands. Because the output of these commands is rather lengthy, it is omitted here, but you are welcome to run these commands in your lab.

11. The LUN is now fully configured for multipath access, so the only steps that remain before you can use the LUN is to create a filesystem and mount it.

Important: When you run the following commands in your lab you will need to substitute in the /dev/mapper/… string that identifies the specific LUN in your lab. You can get that string from the output of ls -l /dev/mapper. You can use bash tab completion when entering the multipath file name to save yourself some tedious typing.

```
[root@rhel1 ~]# mkfs.ext4 /dev/mapper/3600a0980774f6a36625d4d4f7775564d
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=16 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables:  done
Writing inode tables:  done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information:  done

[root@rhel1 ~]# mkdir /linuxlun
[root@rhel1 ~]# mount -t ext4 -o discard /dev/mapper/3600a0980774f6a36625d4d4f7775564d
 /linuxlun
[root@rhel1 ~]# ls /linuxlun
```

```
lost+found
[root@rhel1 ~]# df
Filesystem                                    1K-blocks    Used Available Use% Mounted on
/dev/mapper/rhel-root                          39265556 2811432  36454124   8% /
devtmpfs                                         930956       0    930956   0% /dev
tmpfs                                            941996       0    941996   0% /dev/shm
tmpfs                                            941996   98884    843112  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     960     18496   5% /svm1
/dev/mapper/3600a0980774f6a36625d4d4f7775564d  10190100   36888   9612540   1% /linuxlun
[root@rhel1 ~]# ls /linuxlun
lost+found
[root@rhel1 ~]# echo "hello from rhel1" > /linuxlun/test.txt
[root@rhel1 ~]# cat /linuxlun/test.txt
hello from rhel1
[root@rhel1 iscsi]# ls -l /linuxlun/test.txt
-rw-r--r-- 1 root root 17 Apr  9 18:00 /linuxlun/test.txt
[root@rhel1 ~]#
```

The discard option for **mount** allows the Red Hat host to utilize space reclamation for the LUN.

12. To have RHEL automatically mount the LUN's filesystem at boot time, issue the following command (modified to reflect the multipath device path being used in your instance of the lab) to add the mount information to the /etc/fstab file. Issue the following command as a single line.

```
[root@rhel1 ~]# echo '/dev/mapper/3600a0980774f6a36625d4d4f7775564d
 /linuxlun ext4 _netdev,discard,defaults 0 0' >> /etc/fstab
[root@rhel1 ~]#
```

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