Exercise 4: Configuring NVMe in a Storage VM

In this exercise, you use best practice tools to create a simple NVMe server in a storage VM.

Objectives

This exercise focuses on enabling you to do the following:

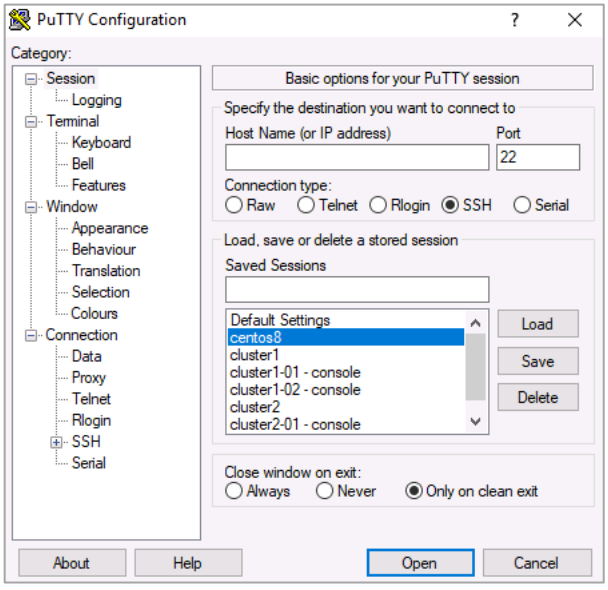
- Prepare the Linux host to use the NVMe protocol
- Use NetApp ONTAP System Manager to configure a storage VM for NVMe over TCP (NVMe/TCP)
- Create NVMe namespaces
- Access the NVMe-attached namespace on the Linux host

Lab Equipment

Use the following equipment to complete the exercise:

System	Host Name	IP Addresses	User Name	Password
Windows Server	jumphost	192.168.0.5	DEMO\Administrator	Netapp1!
ONTAP cluster-management LIF (cluster2)	cluster2	192.168.0.102	admin (case sensitive)	Netapp1!
CentOS Linux Server	centos8	192.168.0.21	root (case sensitive)	Netapp1!

Task 1: Prepare the Linux host for NVMe

Step	Action
1-1	Open a PuTTY session with your Linux host, which is named **centos8**. 
1-2	From PuTTY, log in to your Linux system: • User name: **root** • Password: **Netapp1!**

Step	Action

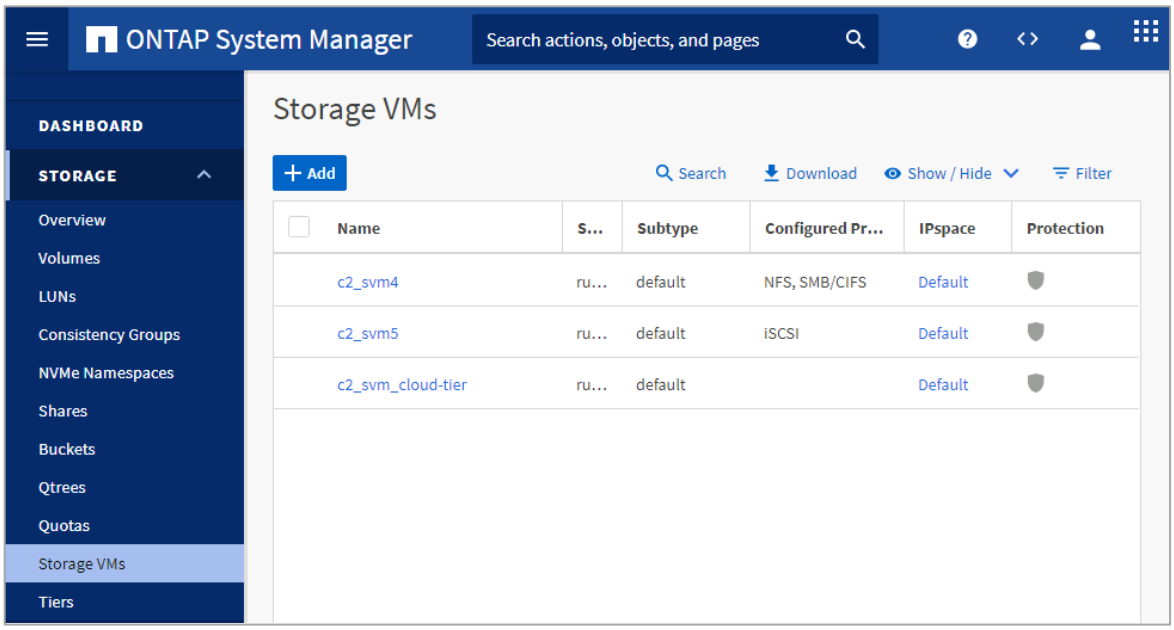
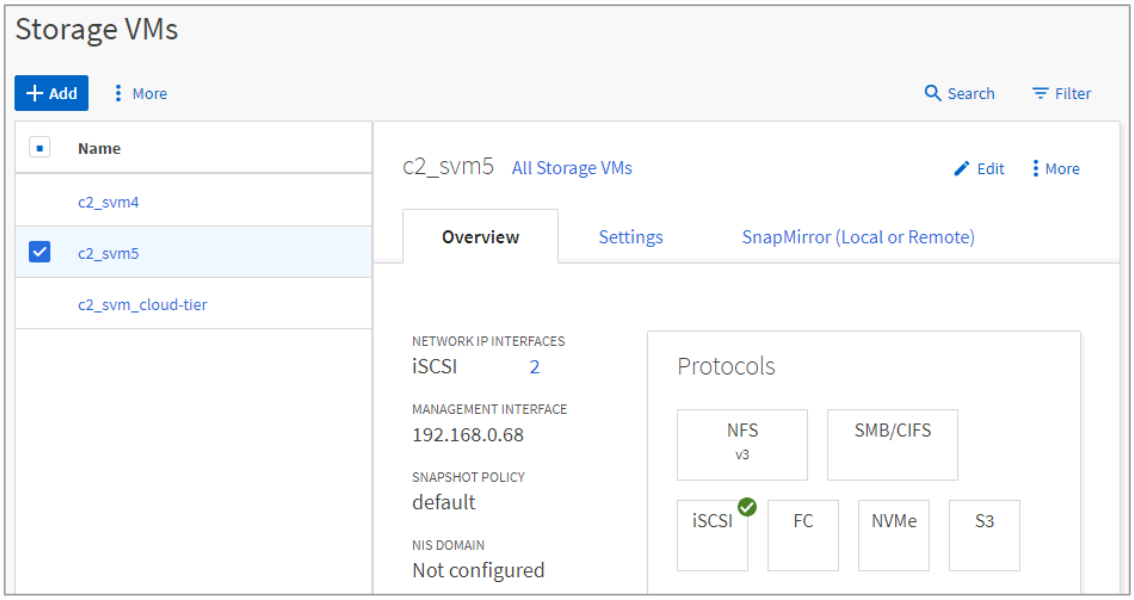
1-3	Load the core NVMe kernel module into memory.
	modprobe -v nvme
	Sample output:
	`[root@centos8 nas1]# modprobe -v nvme`
	`insmod /lib/modules/4.18.0-240.el8.x86_64/kernel/drivers/nvme/host/nvme-core.ko.xz`
	`insmod /lib/modules/4.18.0-240.el8.x86_64/kernel/drivers/nvme/host/nvme.ko.xz`

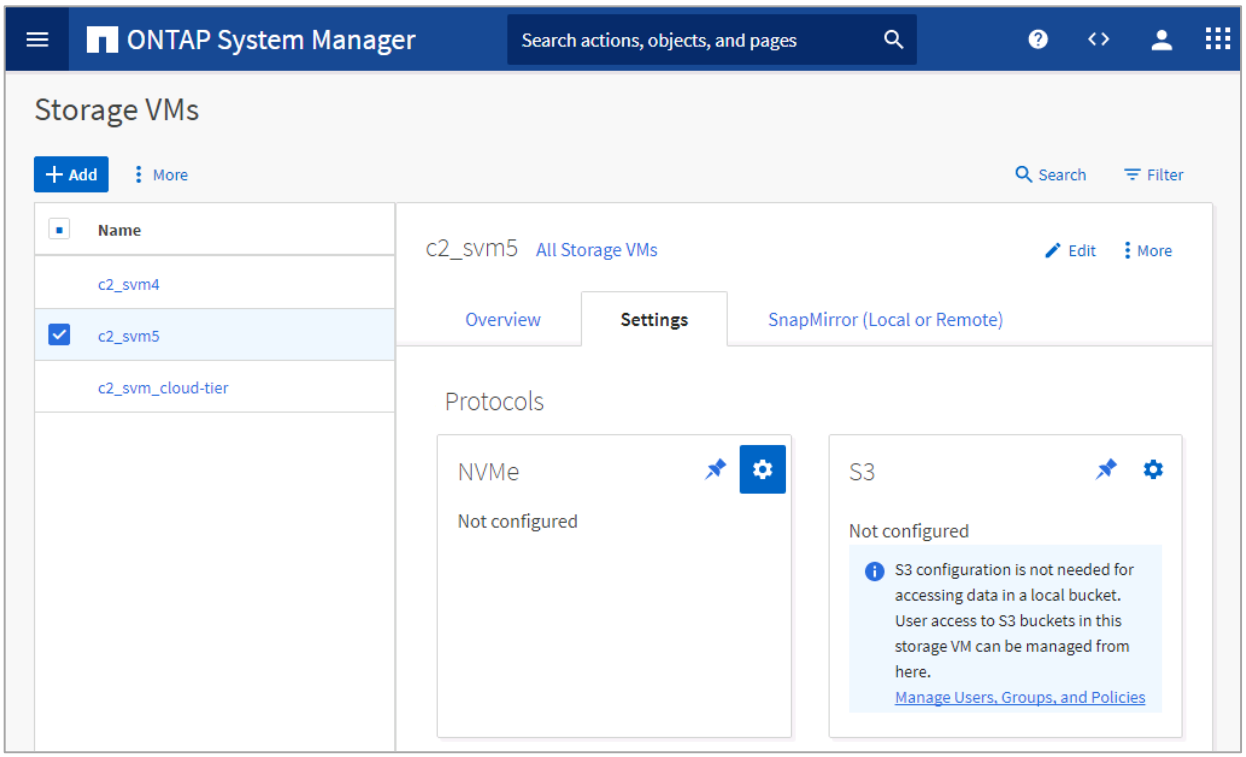
1-4	Load the NVMe over TCP kernel module into memory.
	modprobe -v nvme-tcp
	Sample output:
	`[root@centos8 nas1]# modprobe -v nvme-tcp`
	`insmod /lib/modules/4.18.0-240.el8.x86_64/kernel/drivers/nvme/host/nvme-fabrics.ko.xz`
	insmod /lib/modules/4.18.0-240.el8.x86_64/kernel/drivers/nvme/host/nvme-tcp.ko.xz

1-5	Display the host NVMe Qualified Name (NQN).
	nvme show-hostnqn
	Sample output:
	`[root@centos8 nas1]# nvme show-hostnqn`
	`nqn.2014-08.org.nvmexpress:uuid:2322432f-74c6-4bf7-9a45-7e5a23e0f215`

Task 2: Enable NVMe Over TCP on a Storage VM

Step	Action
2-1	Log in to NetApp ONTAP System Manager for cluster2.
2-2	From the System Manager menu, select **Storage > Storage VMs**.

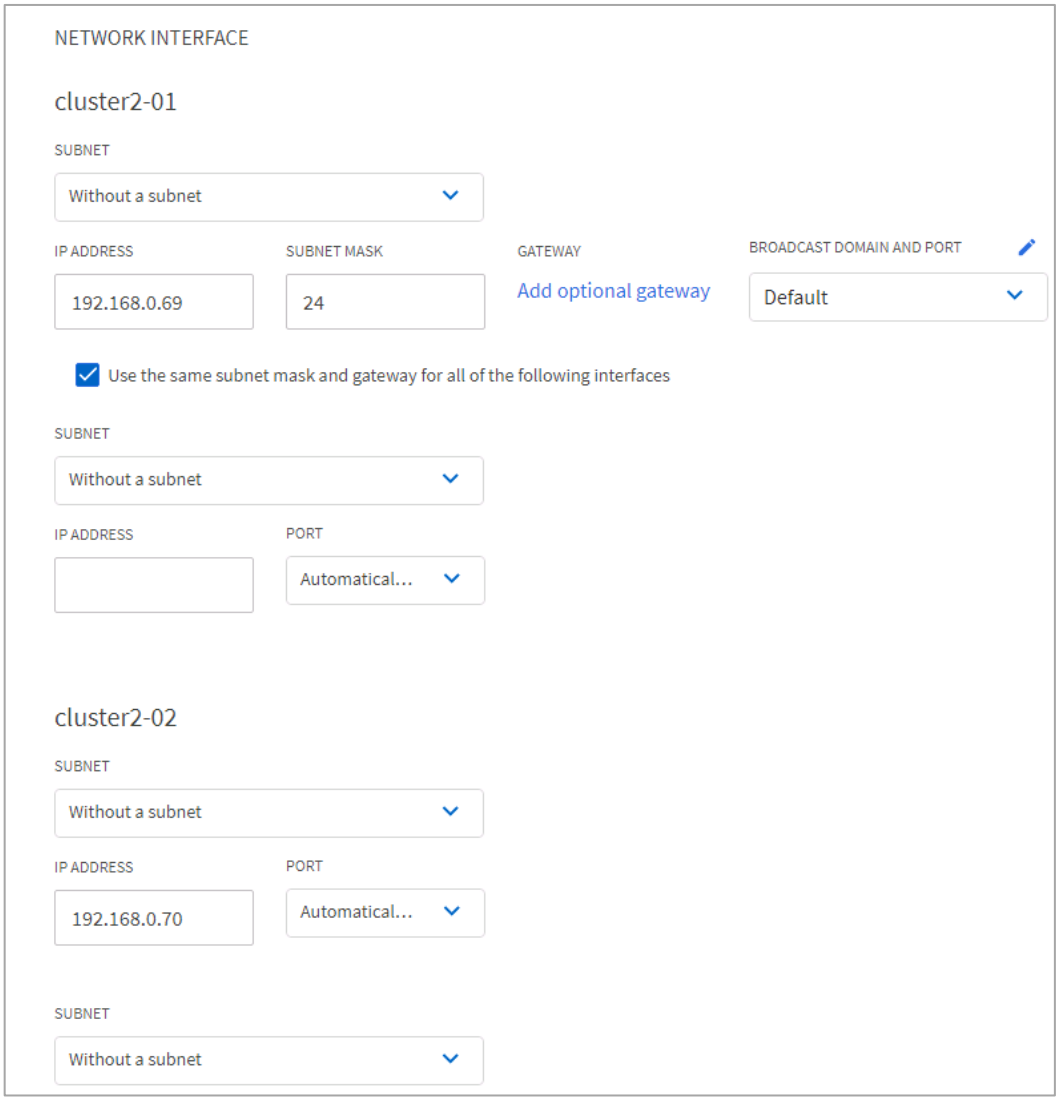
Step	Action
2-3	Click **c2_svm5**. 
2-4	On the c2_svm5 details page, click the **Settings** tab. 

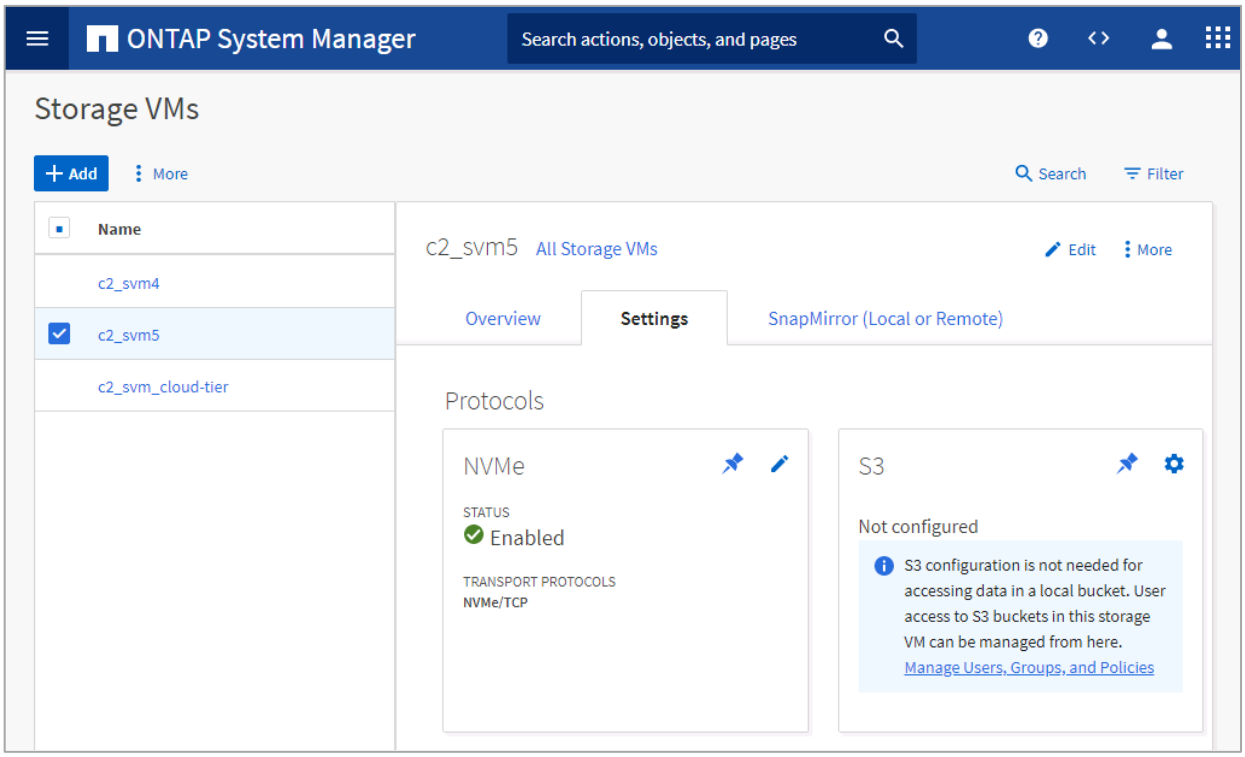
Step	Action
2-5	In the NVMe panel, click the gear icon. 
2-6	In the Configure NVMe window, select **Enable NVMe/TCP**. 

2-7

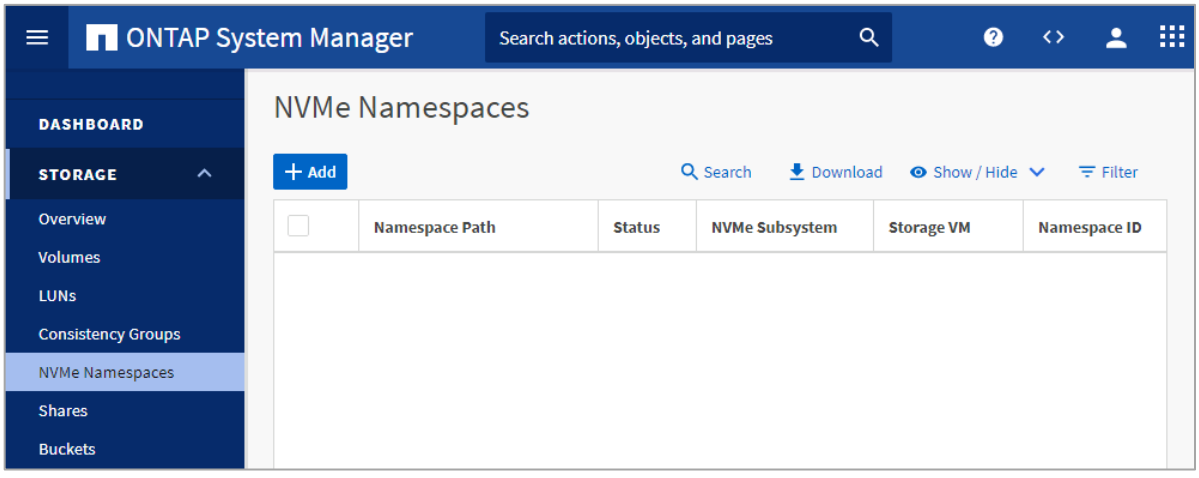
In the Network interface section, specify the following settings:

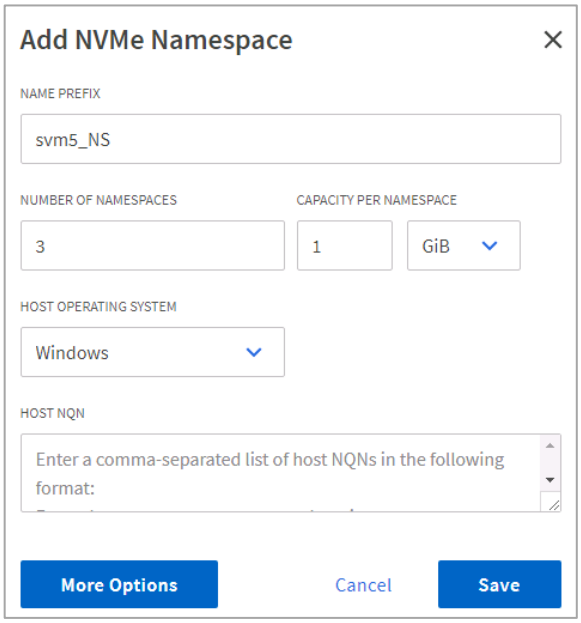
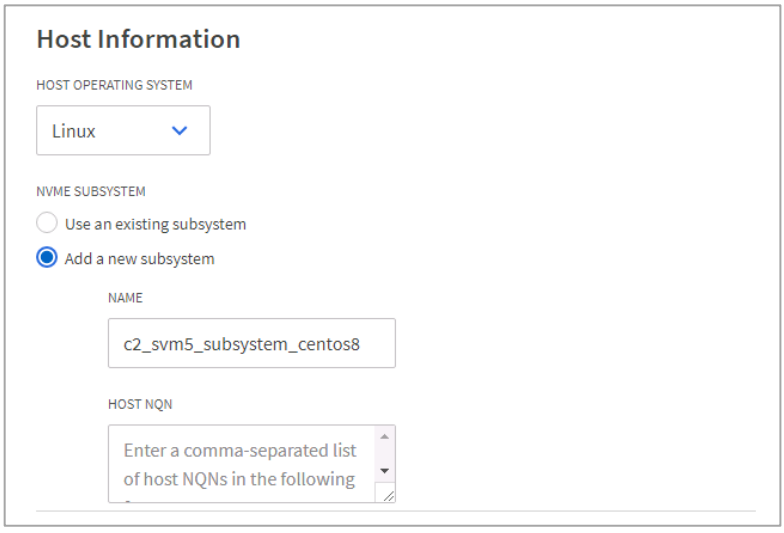
- Enable NVMe/TCP: **<selected>**
- Cluster2-01:
  - Subnet: **Without a subnet** (default)
  - IP Address: **192.168.0.69**
  - Subnet Mask: **24**
  - Gateway: **192.168.0.1** (default)
  - Broadcast Domain and Port: **Default**
  - Use the same subnet mask and gateway for all of the following interfaces **<selected>**
- Cluster2-02:
  - Subnet: **Without a subnet** (default)
  - IP Address: **192.168.0.70**
  - Port: **Automatically select a home port (recommended)** (default)



Step	Action
2-8	Review the configuration, and then click **Save**.
2-9	Verify that the NVMe protocol is enabled on the storage VM. 

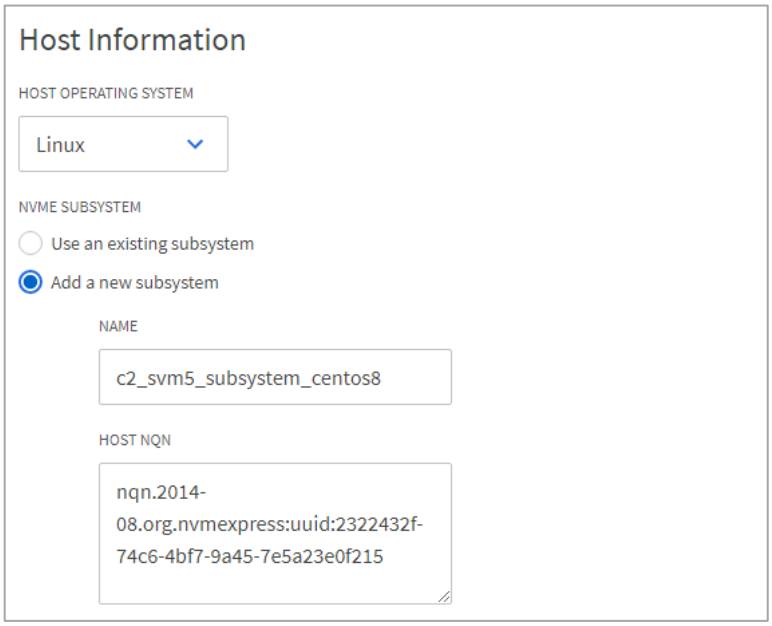
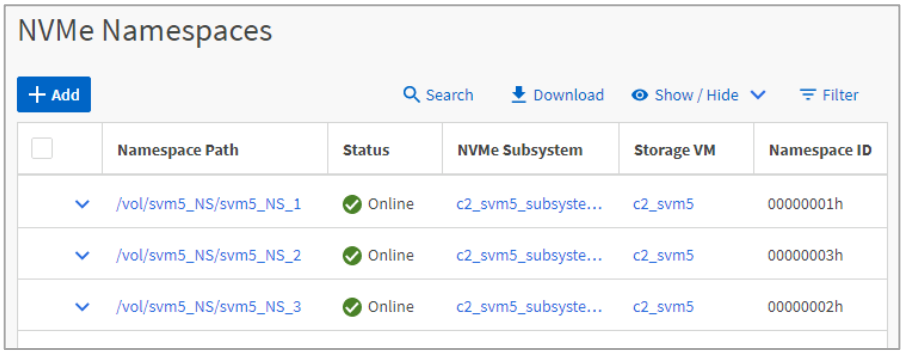
Task 3: Create NVMe Namespaces

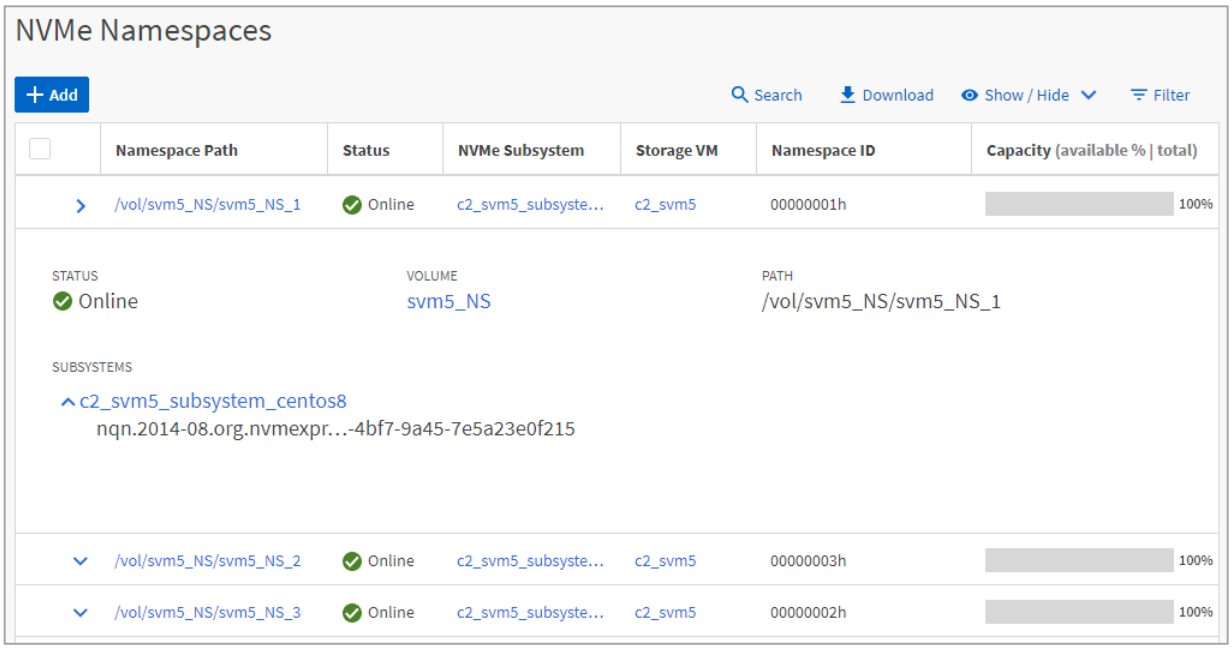
Step	Action
3-1	From the System Manager menu, select **Storage > NVMe Namespaces**, and then click **Add**. 

Step	Action
3-2	In the Add NVMe Namespaces dialog box, specify the following settings: • Name Prefix: **svm5_NS** • Number of Namespaces: **3** • Capacity: **1 GiB** 
3-3	Click **More Options**.
3-4	On the Add NVMe Namespaces page, scroll to the Host Information section, and then specify the following settings: • Host Operating System: **Linux** • Add a new subsystem: **<selected>** • Name: **c2_svm5_subsystem_centos8** 

Step	Action

3-5	Return to the PuTTY session with the Linux host, and copy the host NVMe Qualified Name value to your clipboard.
	`nvme show-hostnqn`
	Sample output:
	`[root@centos8 nas1]# nvme show-hostnqn`
	`nqn.2014-08.org.nvmexpress:uuid:2322432f-74c6-4bf7-9a45-7e5a23e0f215`

3-6	In the Host NQN field, paste the NQN of the client host to include the Linux host in the new NVMe subsystem. 
3-7	Click **Save**.
3-8	Verify the creation of the namespaces. 

Step	Action
3-9	Expand **svm5_NS_1** and observe the configuration settings. 

Task 4: Access the NVMe-Attached Namespace from the Linux Host

Step	Action
4-1	Return to the PuTTY session with the Linux host.

4-2	Display a list of the currently known NVMe namespaces, which is empty.
	`nvme list`
	
	Sample output:
	`[root@centos8 nas1]# nvme list`
	`Node SN Model`
	`Namespace Usage Format FW Rev`
	`---------- -------------------- ---------- --------`
	`---------- -------------------------- ---------------- --------`

Step	Action

4-3	Discover the NVMe enabled storage VM by using the IP address that you assigned to a LIF.
	nvme discover -t tcp -a 192.168.0.69 -s 4420
	
	Sample output:
	`[root@centos8 nas1]# nvme discover -t tcp -a 192.168.0.69 -s 4420`
	
	`Discovery Log Number of Records 1, Generation counter 4`
	`=====Discovery Log Entry 0======`
	`trtype:  tcp`
	`adrfam:  ipv4`
	`subtype: nvme subsystem`
	`treq:    not specified`
	`portid:  0`
	`trsvcid: 4420`
	`subnqn:  nqn.1992-08.com.netapp:sn.935d3f1d99c211ecb61e005056b08c1a:`
	`subsystem.c2_svm5_subsystem_centos8`
	`traddr:  192.168.0.69`
	`sectype: none`

4-4	Connect to the NVMe enabled storage VM by using the subsystem NQN that you obtained in the previous step. **nvme connect -t tcp -n nqn.1992-08.com.netapp:sn.935d3f1d99c211ecb61e00 5056b08c1a:subsystem.c2_svm5_subsystem_centos8 -a 192.168.0.69 -s 4420**

4-5	Display a list of the discovered NVMe namespaces, and observe the namespace device names.
	nvme list
	
	Sample output:
	`[root@centos8 nas1]# nvme list`
	`Node  SN  Model`
	`Namespace Usage  Format  FW Rev`
	`---------- ---------- ----------`
	`--------- ---------- ---------- --------`
	`/dev/nvme1n1  wpEzyNSZi-/TAAAAAAAB NetApp ONTAP Controller`
	`1  1.07  GB /  1.07  GB  4 KiB +  0 B  FFFFFFFF`
	`/dev/nvme1n2  wpEzyNSZi-/TAAAAAAAB NetApp ONTAP Controller`
	`2  1.07  GB /  1.07  GB  4 KiB +  0 B  FFFFFFFF`
	`/dev/nvme1n3  wpEzyNSZi-/TAAAAAAAB NetApp ONTAP Controller`
	`3  1.07  GB /  1.07  GB  4 KiB +  0 B  FFFFFFFF`

Step	Action

4-6	Build an ext2 type file system in the first NVMe namespace.
	mkfs -t ext2 /dev/nvme1n1
	
	Sample output:
	`[root@centos8 nas1]# mkfs -t ext2 /dev/nvme1n1`
	`mke2fs 1.45.6 (20-Mar-2020)`
	`Creating filesystem with 262144 4k blocks and 65536 inodes`
	`Filesystem UUID: 4c9343e4-567a-4318-8b56-b7477e781541`
	`Superblock backups stored on blocks:`
	`32768, 98304, 163840, 229376`
	
	`Allocating group tables: done`
	`Writing inode tables: done`
	`Writing superblocks and filesystem accounting information: done`

4-7	Build an ext3 type file system in the second NVMe namespace.
	mkfs -t ext3 /dev/nvme1n2
	
	Sample output:
	`[root@centos8 nas1]# mkfs -t ext3 /dev/nvme1n2`
	`mke2fs 1.45.6 (20-Mar-2020)`
	`Creating filesystem with 262144 4k blocks and 65536 inodes`
	`Filesystem UUID: 25d2467b-26df-40e4-bb35-65ef12fea717`
	`Superblock backups stored on blocks:`
	`32768, 98304, 163840, 229376`
	
	`Allocating group tables: done`
	`Writing inode tables: done`
	`Creating journal (8192 blocks): done`
	`Writing superblocks and filesystem accounting information: done`

4-8	Build an ext4 type file system in the third NVMe namespace.
	mkfs -t ext4 /dev/nvme1n3
	
	Sample output:
	`[root@centos8 nas1]# mkfs -t ext4 /dev/nvme1n3`
	`mke2fs 1.45.6 (20-Mar-2020)`
	`Creating filesystem with 262144 4k blocks and 65536 inodes`
	`Filesystem UUID: b9d326f5-d264-4680-84ee-f6064a2b9c21`
	`Superblock backups stored on blocks:`
	`32768, 98304, 163840, 229376`
	
	`Allocating group tables: done`
	`Writing inode tables: done`
	`Creating journal (8192 blocks): done`
	`Writing superblocks and filesystem accounting information: done`

4-9	Create mount point directories for the three NVMe namespaces. **mkdir /nvme_vol1_ext2** **mkdir /nvme_vol2_ext3** **mkdir /nvme_vol3_ext4**

Step	Action
4-10	Attach the three NVMe namespaces to the three mount point directories. **mount /dev/nvme1n1 /nvme_vol1_ext2** **mount /dev/nvme1n2 /nvme_vol2_ext3** **mount /dev/nvme1n3 /nvme_vol3_ext4**

4-11	Verify that the three NVMe namespaces are mounted and accessible.
	df -h /nvme*
	
	Sample output:
	`[root@centos8 nas1]# df -h /nvme*`
	`Filesystem      Size  Used Avail Use% Mounted on`
	`/dev/nvme1n1   1008M  1.3M  956M   1% /nvme_vol1_ext2`
	`/dev/nvme1n2    976M  1.3M  924M   1% /nvme_vol2_ext3`
	`/dev/nvme1n3    976M  2.6M  907M   1% /nvme_vol3_ext4`

End of exercise