NVMe over InfiniBand Setup

E-Series

NetApp
February 15, 2022

Table of Contents

NVMe over InfiniBand Setup

Verify Linux support and review restrictions

As a first step, you should verify that your Linux configuration is supported and also review the controller, host, and recovery restrictions.

Verify the Linux configuration is supported

To ensure reliable operation, you create an implementation plan and then use the NetApp Interoperability Matrix Tool (IMT) to verify that the entire configuration is supported.

Steps

1. Go to the NetApp Interoperability Matrix Tool.
2. Click on the **Solution Search** tile.
3. In the **Protocols › SAN Host** area, click the **Add** button next to **E-Series SAN Host**.
4. Click **View Refine Search Criteria**.

   The Refine Search Criteria section is displayed. In this section you may select the protocol that applies, as well as other criteria for the configuration such as Operating System, NetApp OS, and Host Multipath driver.

5. Select the criteria you know you want for your configuration, and then see what compatible configuration elements apply.
6. As necessary, make the updates for your operating system and protocol that are prescribed in the tool.

   Detailed information for your chosen configuration is accessible on the View Supported Configurations page by clicking the right page arrow.

Review NVMe over InfiniBand restrictions

Before using NVMe over InfiniBand, review the controller, host, and recovery restrictions. For an up-to-date listing of all restrictions, see the NetApp Interoperability Matrix Tool.

Controller restrictions

- NVMe over InfiniBand can be configured for EF300 (100GB controllers only), EF600, EF570, or E5700 controllers. The controllers must have 100GB or 200GB InfiniBand host ports.
- This protocol can be used only for EF300, EF600, EF570, or EF570 controllers. A minimum of 32 GB of physical memory is required to use this protocol on EF600, EF570, and E5700 controllers. For the EF300, a minimum of 16 GB of physical memory is required. If the minimum memory requirements for the controllers are not met during start of day operations, a message is displayed that helps you diagnose the problem.
- No simplex (single controller) configurations are supported.
- There is no support for mixed NVMe over InfiniBand and SCSI host interfaces.
- For EF300 controllers, no more than 64 NVMe hosts can be supported on the IB interface.

Host, host protocol, and host operating system restrictions

- The host must be running the latest compatible RHEL 7, SUSE Linux Enterprise Server 12 or 15 service pack operating system. See the NetApp Interoperability Matrix Tool for a complete list of the latest requirements.
- The only supported host channel adapters are from Mellanox. See the NetApp Interoperability Matrix Tool for more information.
- The only supported host interface card (HIC) is the 100G or 200G EDR IB HIC, which also supports iSER and SRP (but iSER and SRP are not supported simultaneously).

Storage and disaster recovery restrictions

- Asynchronous and synchronous mirroring are not supported.
- Thin provisioning (the creation of thin volumes) is not supported.

Configure IP addresses using DHCP

To configure communications between the management station and the storage array, use Dynamic Host Configuration Protocol (DHCP) to provide IP addresses.

What you'll need

A DHCP server installed and configured on the same subnet as the storage management ports.

About this task

Each storage array has either one controller (simplex) or two controllers (duplex), and each controller has two storage management ports. Each management port will be assigned an IP address.

The following instructions refer to a storage array with two controllers (a duplex configuration).

Steps

1. If you have not already done so, connect an Ethernet cable to the management station and to management port 1 on each controller (A and B).

   The DHCP server assigns an IP address to port 1 of each controller.

   > Do not use management port 2 on either controller. Port 2 is reserved for use by NetApp technical personnel.

   > If you disconnect and reconnect the Ethernet cable, or if the storage array is power-cycled, DHCP assigns IP addresses again. This process occurs until static IP addresses are configured. It is recommended that you avoid disconnecting the cable or power-cycling the array.

   If the storage array cannot get DHCP-assigned IP addresses within 30 seconds, the following default IP addresses are set:

   - Controller A, port 1: 169.254.128.101
   - Controller B, port 1: 169.254.128.102
   - Subnet mask: 255.255.0.0

2. Locate the MAC address label on the back of each controller, and then provide your network administrator with the MAC address for port 1 of each controller.

   Your network administrator needs the MAC addresses to determine the IP address for each controller. You will need the IP addresses to connect to your storage system through your browser.

Install SANtricity Storage Manager for SMcli (SANtricity software version 11.53 or earlier)

If you are using SANtricity software 11.53 or earlier, you can install the SANtricity Storage Manager software on your management station to help manage the array.

SANtricity Storage Manager includes the command line interface (CLI) for additional management tasks, and also the Host Context Agent for pushing host configuration information to the storage array controllers through the I/O path.

> If you are using SANtricity software 11.60 and newer, you do not need to follow these steps. The SANtricity Secure CLI (SMcli) is included in the SANtricity OS and downloadable through the SANtricity System Manager. For more information on how to download the SMcli through the SANtricity System Manager, refer to the *Download command line interface (CLI)* topic under the SANtricity System Manager Online Help.

What you'll need

- SANtricity software 11.53 or earlier.
- Correct administrator or superuser privileges.
- A system for the SANtricity Storage Manager client with the following minimum requirements:
  - **RAM**: 2 GB for Java Runtime Engine
  - **Disk space**: 5 GB
  - **OS/Architecture**: For guidance on determining the supported operating system versions and architectures, go to NetApp Support. From the **Downloads** tab, go to **Downloads › E-Series SANtricity Storage Manager**.

About this task

This task describes how to install SANtricity Storage Manager on both the Windows and Linux OS platforms, because both Windows and Linux are common management station platforms when Linux is used for the data host.

Steps

1. Download the SANtricity software release at NetApp Support. From the **Downloads** tab, go to **Downloads › E-Series SANtricity Storage Manager**.
2. Run the SANtricity installer.

Windows	Linux
Double-click the SMIA*.exe installation package to start the installation.	a. Go to the directory where the SMIA*.bin installation package is located. b. If the temp mount point does not have execute permissions, set the `IATEMPDIR` variable. Example: `IATEMPDIR=/root ./SMIA-LINUXX64-11.25.0A00.0002.bin` c. Run the `chmod +x SMIA*.bin` command to grant execute permission to the file. d. Run the `./SMIA*.bin` command to start the installer.

3. Use the installation wizard to install the software on the management station.

Access SANtricity System Manager and use the Setup wizard

To configure your storage array, you can use the Setup wizard in SANtricity System Manager.

SANtricity System Manager is a web-based interface embedded on each controller. To access the user interface, you point a browser to the controller's IP address. A setup wizard helps you get started with system configuration.

What you'll need

- Out-of-band management.
- A management station for accessing SANtricity System Manager that includes one of the following browsers:

Browser	Minimum version
Google Chrome	79
Microsoft Internet Explorer	11
Microsoft Edge	79
Mozilla Firefox	70
Safari	12

About this task

The wizard automatically relaunches when you open System Manager or refresh your browser and *at least one* of the following conditions is met:

- No pools and volume groups are detected.
- No workloads are detected.
- No notifications are configured.

Steps

1. From your browser, enter the following URL: `https://<DomainNameOrIPAddress>`

   `IPAddress` is the address for one of the storage array controllers.

   The first time SANtricity System Manager is opened on an array that has not been configured, the Set Administrator Password prompt appears. Role-based access management configures four local roles: admin, support, security, and monitor. The latter three roles have random passwords that cannot be guessed. After you set a password for the admin role, you can change all of the passwords using the admin credentials. For more information about the four local user roles, see the online help available in the SANtricity System Manager user interface.

2. Enter the System Manager password for the admin role in the Set Administrator Password and Confirm Password fields, and then click **Set Password**.

   The Setup wizard launches if there are no pools, volumes groups, workloads, or notifications configured.

3. Use the Setup wizard to perform the following tasks:
   - **Verify hardware (controllers and drives)** — Verify the number of controllers and drives in the storage array. Assign a name to the array.
   - **Verify hosts and operating systems** — Verify the host and operating system types that the storage array can access.
   - **Accept pools** — Accept the recommended pool configuration for the express installation method. A pool is a logical group of drives.
   - **Configure alerts** — Allow System Manager to receive automatic notifications when a problem occurs with the storage array.
   - **Enable AutoSupport** — Automatically monitor the health of your storage array and have dispatches sent to technical support.
4. If you have not already created a volume, create one by going to **Storage › Volumes › Create › Volume**.

   For more information, see the online help for SANtricity System Manager.

Configure subnet manager

A subnet manager must be running in your environment on your switch or on your hosts. If you are running it host-side, use the following procedure to set it up.

Steps

1. Install the `opensm` package on any hosts that will be running the subnet manager.
2. Use the `ibstat -p` command to find `GUID0` and `GUID1` of the HCA ports. For example:

```
# ibstat -p
 0x248a070300a80a80
 0x248a070300a80a81
```

3. Enable Subnet Manager on each port of the connected HCA on the host:

   SLES example

   - Add the following two lines to /etc/rc.d/rc.after. Substitute the values you found in step 2 for `GUID0` and `GUID1`. For `P0` and `P1`, use the subnet manager priorities, with 1 being the lowest and 15 the highest:

     ```
     opensm -B -g GUID0 -p P0 -f /var/log/opensm-ib0.log
     opensm -B -g GUID1 -p P1 -f /var/log/opensm-ib1.log
     ```

     An example of the command with value substitutions:

     ```
     # cat /etc/rc.d/rc.local
      opensm -B -g 0x248a070300a80a80 -p 15 -f /var/log/opensm-ib0.log
      opensm -B -g 0x248a070300a80a81 -p 1 -f /var/log/opensm-ib1.log
     ```

   RHEL example

   - Add the following two lines to /etc/rc.d/rc.local. Substitute the values you found in step 2 for `GUID0` and `GUID1`. For `P0` and `P1`, use the subnet manager priorities, with 1 being the lowest and 15 the highest:

     ```
     opensm -B -g GUID0 -p P0 -f /var/log/opensm-ib0.log
     opensm -B -g GUID1 -p P1 -f /var/log/opensm-ib1.log
     ```

     An example of the command with value substitutions:

     ```
     # cat /etc/rc.d/rc.local
      opensm -B -g 0x248a070300a80a80 -p 15 -f /var/log/opensm-ib0.log
      opensm -B -g 0x248a070300a80a81 -p 1 -f /var/log/opensm-ib1.log
     ```

Set up NVMe over InfiniBand on the host side

Configuring an NVMe initiator in an InfiniBand environment includes installing and configuring the infiniband, nvme-cli, and rdma packages, configuring initiator IP addresses, and setting up the NVMe-oF layer on the host.

What you'll need

You must be running the latest compatible RHEL 7, RHEL 8, SUSE Linux Enterprise Server 12 or 15 service

pack operating system. See the NetApp Interoperability Matrix Tool for a complete list of the latest requirements.

Steps

1. Install the rdma, nvme-cli, and infiniband packages:

   SLES 12 or SLES 15

   ```
   # zypper install infiniband-diags
   # zypper install rdma-core
   # zypper install nvme-cli
   ```

   RHEL 7 or RHEL 8

   ```
   # yum install infiniband-diags
   # yum install rdma-core
   # yum install nvme-cli
   ```

2. Enable `ipoib`. Edit the /etc/rdma/rdma.conf file and modify the entry for loading `ipoib`:

   ```
   IPOIB_LOAD=yes
   ```

3. Check that both ib port links are up and the State = Active:

   ```
    # ibstat
   ```

```
CA 'mlx4_0'
        CA type: MT4099
        Number of ports: 2
        Firmware version: 2.40.7000
        Hardware version: 1
        Node GUID: 0x0002c90300317850
        System image GUID: 0x0002c90300317853
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 40
                Base lid: 4
                LMC: 0
                SM lid: 4
                Capability mask: 0x0259486a
                Port GUID: 0x0002c90300317851
                Link layer: InfiniBand
        Port 2:
                State: Active
                Physical state: LinkUp
                Rate: 56
                Base lid: 5
                LMC: 0
                SM lid: 4
                Capability mask: 0x0259486a
                Port GUID: 0x0002c90300317852
                Link layer: InfiniBand
```

4. Set up IPv4 IP addresses on the ib ports.

SLES 12 or SLES 15

Create the file /etc/sysconfig/network/ifcfg-ib0 with the following contents.

```
BOOTPROTO='static'
BROADCAST=
ETHTOOL_OPTIONS=
IPADDR='10.10.10.100/24'
IPOIB_MODE='connected'
MTU='65520'
NAME=
NETWORK=
REMOTE_IPADDR=
STARTMODE='auto'
```

Then, create the file /etc/sysconfig/network/ifcfg-ib1.

```
BOOTPROTO='static'
BROADCAST=
ETHTOOL_OPTIONS=
IPADDR='11.11.11.100/24'
IPOIB_MODE='connected'
MTU='65520'
NAME=
NETWORK=
REMOTE_IPADDR=
STARTMODE='auto'
```

RHEL 7 or RHEL 8

Create the file /etc/sysconfig/network-scripts/ifcfg-ib0 with the following contents.

```
CONNECTED_MODE=no
TYPE=InfiniBand
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
IPADDR='10.10.10.100/24'
DEFROUTE=no
IPV4=FAILURE_FATAL=yes
IPV6INIT=no
NAME=ib0
ONBOOT=yes
```

Then, create the file /etc/sysconfig/network-scripts/ifcfg-ib1:

```
CONNECTED_MODE=no
TYPE=InfiniBand
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
IPADDR='11.11.11.100/24'
DEFROUTE=no
IPV4=FAILURE_FATAL=yes
IPV6INIT=no
NAME=ib1
ONBOOT=yes
```

5. Enable the `ib` interface:

```
# ifup ib0
# ifup ib1
```

6. Verify the IP addresses you will use to connect to the array. Run this command for both `ib0` and `ib1`:

```
# ip addr show ib0
# ip addr show ib1
```

As shown in the example below, the IP address for `ib0` is `10.10.10.255`.

```
10: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc pfifo_fast
state UP group default qlen 256
    link/infiniband
80:00:02:08:fe:80:00:00:00:00:00:00:02:c9:03:00:31:78:51 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
    inet 10.10.10.255 brd 10.10.10.255 scope global ib0
       valid_lft forever preferred_lft forever
    inet6 fe80::202:c903:31:7851/64 scope link
       valid_lft forever preferred_lft forever
```

As shown in the example below, the IP address for `ib1` is `11.11.11.255`.

```
10: ib1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc pfifo_fast
state UP group default qlen 256
    link/infiniband
80:00:02:08:fe:80:00:00:00:00:00:00:02:c9:03:00:31:78:51 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
    inet 11.11.11.255 brd 11.11.11.255 scope global ib0
       valid_lft forever preferred_lft forever
    inet6 fe80::202:c903:31:7851/64 scope link
       valid_lft forever preferred_lft forever
```

7. Set up the NVMe-oF layer on the host. Create the following files under /etc/modules-load.d/ to load the `nvme-rdma` kernel module and make sure the kernel module will always be on, even after a reboot:

```
# cat /etc/modules-load.d/nvme-rdma.conf
  nvme-rdma
```

To verify the `nvme-rdma` kernel module is loaded, run this command:

```
# lsmod | grep nvme
nvme_rdma               36864  0
nvme_fabrics            24576  1 nvme_rdma
nvme_core              114688  5 nvme_rdma,nvme_fabrics
rdma_cm                114688  7
rpcrdma,ib_srpt,ib_srp,nvme_rdma,ib_iser,ib_isert,rdma_ucm
ib_core                393216  15
rdma_cm,ib_ipoib,rpcrdma,ib_srpt,ib_srp,nvme_rdma,iw_cm,ib_iser,ib_umad,
ib_isert,rdma_ucm,ib_uverbs,mlx5_ib,qedr,ib_cm
t10_pi                  16384  2 sd_mod,nvme_core
```

Configure storage array NVMe over InfiniBand connections

If your controller includes an NVMe over InfiniBand port, you can configure the IP address of each port using SANtricity System Manager.

Steps

1. From the System Manager interface, select **Hardware**.
2. If the graphic shows the drives, click **Show back of shelf**.

   The graphic changes to show the controllers instead of the drives.

3. Click the controller with the NVMe over InfiniBand ports you want to configure.

   The controller's context menu appears.

4. Select **Configure NVMe over InfiniBand ports**.

   > The Configure NVMe over InfiniBand ports option appears only if System Manager detects NVMe over InfiniBand ports on the controller.

   The **Configure NVMe over InfiniBand Ports** dialog box opens.

5. In the drop-down list, select the HIC port you want to configure, and then enter the IP address of the port.
6. Click **Configure**.
7. Repeat steps 5 and 6 for the other HIC ports that will be used.

Discover and connect to the storage from the host

Before making definitions of each host in SANtricity System Manager, you must discover the target controller ports from the host, and then establish NVMe connections.

Steps

1. Discover available subsystems on the NVMe-oF target for all paths using the following command:

```
nvme discover -t rdma -a target_ip_address
```

In this command, `target_ip_address` is the IP address of the target port.

> The `nvme discover` command discovers all controller ports in the subsystem, regardless of host access.

```
# nvme discover  -t rdma -a 10.10.10.100
 Discovery Log Number of Records 2, Generation counter 0
 =====Discovery Log Entry 0======
 trtype:  rdma
 adrfam:  ipv4
 subtype: nvme subsystem
 treq:    not specified
 portid:  0
 trsvcid: 4420
 subnqn:  nqn.1992-08.com.netapp:5700.600a098000af41580000000058ed54be
 traddr: 10.10.10.100
 rdma_prtype: infiniband
 rdma_qptype: connected
 rdma_cms:    rdma-cm
 rdma_pkey: 0x0000
=====Discovery Log Entry 1======
 trtype:  rdma
 adrfam:  ipv4
 subtype: nvme subsystem
 treq:    not specified
 portid:  1
 trsvcid: 4420
 subnqn:  nqn.1992-08.com.netapp:5700.600a098000af41580000000058ed54be
 traddr: 11.11.11.100
 rdma_prtype: infiniband
 rdma_qptype: connected
 rdma_cms:    rdma-cm
 rdma_pkey: 0x0000
```

2. Repeat step 1 for any other connections.
3. Connect to the discovered subsystem on the first path using the command: `nvme connect -t rdma -n discovered_sub_nqn -a target_ip_address -Q queue_depth_setting -l controller_loss_timeout_period`

> The above command does not persist through reboot. The `NVMe connect` command will need to be executed after each reboot to re-establish the NVMe connections.

The NVMe connections do not persist through system reboot or extended periods of the controller being unavailable.

Connections are not established for any discovered port inaccessible by the host.

If you specify a port number using this command, the connection fails. The default port is the only port set up for connections.

The recommended queue depth setting is 1024. Override the default setting of 128 with 1024 using the `-Q 1024`command line option, as shown in the following example.

The recommended controller loss timeout period in seconds is 60 minutes (3600 seconds). Override the default setting of 600 seconds with 3600 seconds using the `-l 3600` command line option, as shown in the following example:

```
# nvme connect -t rdma -a 10.10.10.100 -n  nqn.1992-
08.com.netapp:5700.600a098000af41580000000058ed54be -Q 1024 -l 3600
```

4. Use the `nvme list` command to see a list of the NVMe devices currently connected. In the example below, it is `nvme0n1`.

```
 # nvme list

 Node           SN             Model              Namespace
 ------------------------------------------------------------
 /dev/nvme0n1   021648023161   NetApp E-Series       1
```

```
Usage                  Format           FW Rev
-----------------------------------------------------------------
5.37 GB /5.37 GB       512 B + 0 B      0842XXXX
```

5. Connect to the discovered subsystem on the second path:

```
  # nvme connect -t rdma -a 11.11.11.100 -n  nqn.1992-
08.com.netapp:5700.600a098000af41580000000058ed54be -Q 1024 -l 3600
```

6. Use the Linux `lsblk` and `grep` commands to show additional information about each block device:

```
# lsblk | grep nvme

nvme0n1     259:0    0     5G  0 disk
nvme1n1     259:0    0     5G  0 disk
```

7. Use the `nvme list` command to see a new list of the NVMe devices currently connected. In the example below, it is `nvme0n1` and `nvme0n1`.

```
# nvme list
Node           SN             Model                    Namespace
-----------------------------------------------------------------
/dev/nvme0n1   021648023161   NetApp E-Series           1
/dev/nvme1n1   021648023161   NetApp E-Series           1
```

```
Usage                  Format           FW Rev
-----------------------------------------------------------------
5.37 GB /5.37 GB          512 B + 0 B      0842XXXX
5.37 GB /5.37 GB          512 B + 0 B      0842XXXX
```

Define a host

Using SANtricity System Manager, you define the hosts that send data to the storage array. Defining a host is one of the steps required to let the storage array know which hosts are attached to it and to allow I/O access to the volumes.

About this task

Keep these guidelines in mind when you define a host:

- You must define the host identifier ports that are associated with the host.
- Make sure that you provide the same name as the host's assigned system name.
- This operation does not succeed if the name you choose is already in use.
- The length of the name cannot exceed 30 characters.

Steps

1. Select **Storage › Hosts**.
2. Click **Create › Host**.

   The Create Host dialog box appears.

3. Select the settings for the host as appropriate.

Setting	Description
Name	Type a name for the new host.
Host operating system type	Select one of the following options from the drop-down list: • **Linux** for SANtricity 11.60 and newer • **Linux DM-MP (Kernel 3.10 or later)** for pre-SANtricity 11.60
Host interface type	Select the host interface type that you want to use.
Host ports	Do one of the following: • **Select I/O Interface** If the host ports have logged in, you can select host port identifiers from the list. This is the recommended method. • **Manual add** If the host ports have not logged in, look at /etc/nvme/hostnqn on the host to find the hostnqn identifiers and associate them with the host definition. You can manually enter the host port identifiers or copy/paste them from the /etc/nvme/hostnqn file (one at a time) into the **Host ports** field. You must add one host port identifier at a time to associate it with the host, but you can continue to select as many identifiers that are associated with the host. Each identifier is displayed in the **Host ports** field. If necessary, you also can remove an identifier by selecting the **X** next to it.

4. Click **Create**.

Result

After the host is successfully created, SANtricity System Manager creates a default name for each host port configured for the host.

The default alias is `<Hostname_Port Number>`. For example, the default alias for the first port created for `host IPT is IPT_1`.

Assign a volume

You must assign a volume (namespace) to a host or host cluster so it can be used for I/O operations. This assignment grants a host or host cluster access to one or more namespaces in a storage array.

About this task

Keep these guidelines in mind when you assign volumes:

- You can assign a volume to only one host or host cluster at a time.
- Assigned volumes are shared between controllers in the storage array.
- The same namespace ID (NSID) cannot be used twice by a host or a host cluster to access a volume. You must use a unique NSID.

Assigning a volume fails under these conditions:

- All volumes are assigned.
- The volume is already assigned to another host or host cluster.

The ability to assign a volume is unavailable under these conditions:

- No valid hosts or host clusters exist.
- All volume assignments have been defined.

All unassigned volumes are displayed, but functions for hosts with or without Data Assurance (DA) apply as follows:

- For a DA-capable host, you can select volumes that are either DA-enabled or not DA-enabled.
- For a host that is not DA-capable, if you select a volume that is DA-enabled, a warning states that the system must automatically turn off DA on the volume before assigning the volume to the host.

Steps

1. Select **Storage › Hosts**.
2. Select the host or host cluster to which you want to assign volumes, and then click **Assign Volumes**.

   A dialog box appears that lists all the volumes that can be assigned. You can sort any of the columns or type something in the **Filter** box to make it easier to find particular volumes.

3. Select the checkbox next to each volume that you want to assign or select the checkbox in the table header to select all volumes.
4. Click **Assign** to complete the operation.

Result

After successfully assigning a volume or volumes to a host or a host cluster, the system performs the following actions:

- The assigned volume receives the next available NSID. The host uses the NSID to access the volume.
- The user-supplied volume name appears in volume listings associated to the host.

Display the volumes visible to the host

You can use the SMdevices tool to view volumes currently visible on the host. This tool is part of the nvme-cli package, and can be used as an alternative to the `nvme list` command.

To view information about each NVMe path to an E-Series volume, use the `nvme netapp smdevices [-o <format>]` command. The output `<format>` can be normal (the default if -o is not used), column, or json.

```
# nvme netapp smdevices
/dev/nvme1n1, Array Name ICTM0706SYS04, Volume Name NVMe2, NSID 1, Volume
ID 000015bd5903df4a00a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n2, Array Name ICTM0706SYS04, Volume Name NVMe3, NSID 2, Volume
ID 000015c05903e24000a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n3, Array Name ICTM0706SYS04, Volume Name NVMe4, NSID 4, Volume
ID 00001bb0593a46f400a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n4, Array Name ICTM0706SYS04, Volume Name NVMe6, NSID 6, Volume
ID 00001696593b424b00a0980000af4112, Controller A, Access State unknown,
2.15GB
/dev/nvme2n1, Array Name ICTM0706SYS04, Volume Name NVMe2, NSID 1, Volume
ID 000015bd5903df4a00a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n2, Array Name ICTM0706SYS04, Volume Name NVMe3, NSID 2, Volume
ID 000015c05903e24000a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n3, Array Name ICTM0706SYS04, Volume Name NVMe4, NSID 4, Volume
ID 00001bb0593a46f400a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n4, Array Name ICTM0706SYS04, Volume Name NVMe6, NSID 6, Volume
ID 00001696593b424b00a0980000af4112, Controller B, Access State unknown,
2.15GB
```

Set up failover

To provide a redundant path to the storage array, you can configure the host to run failover.

What you'll need

You must install the required packages on your system.

- For Red Hat (RHEL) hosts, verify the packages are installed by running `rpm -q device-mapper-multipath`

- For SLES hosts, verify the packages are installed by running `rpm -q multipath-tools`

> Refer to NetApp Interoperability Matrix Tool to ensure any required updates are installed as multipathing may not work correctly with the GA versions of SLES or RHEL.

About this task

RHEL 7 and SLES 12 use Device Mapper Multipath (DMMP) for multipathing when using NVMe over Infiniband. RHEL 8 and SLES 15 use a built in Native NVMe Failover. Depending on which OS you are running, some additional configuration of multipath is required to get it running properly.

Enable Device Mapper Multipath (DMMP) for RHEL 7 or SLES 12

By default, DM-MP is disabled in RHEL and SLES. Complete the following steps to enable DM-MP components on the host.

Steps

1. Add the NVMe E-Series device entry to the devices section of the /etc/multipath.conf file, as shown in the following example:

```
devices {
        device {
                vendor "NVME"
                product "NetApp E-Series*"
                path_grouping_policy group_by_prio
                failback immediate
                no_path_retry 30
        }
}
```

2. Configure `multipathd` to start at system boot.

```
# systemctl enable multipathd
```

3. Start `multipathd` if it is not currently running.

```
# systemctl start multipathd
```

4. Verify the status of `multipathd` to make sure it is active and running:

```
# systemctl status multipathd
```

Setting up RHEL 8 with Native NVMe Multipathing

Native NVMe Multipathing is disabled by default in RHEL 8 and must be enabled using the steps below.

1. Setup `modprobe` rule to turn on Native NVMe Multipathing.

```
# echo "options nvme_core multipath=y" >> /etc/modprobe.d/50-
nvme_core.conf
```

2. Remake `initramfs` with new `modprobe` parameter.

```
# dracut -f
```

3. Reboot server to bring it up with the Native NVMe Multipathing enabled.

```
# reboot
```

4. Verify Native NVMe Multipathing has been enabled after the host boots back up.

```
# cat /sys/module/nvme_core/parameters/multipath
```

   a. If the command output is `N`, then Native NVMe Multipathing is still disabled.

   b. If the command output is `Y`, then Native NVMe Multipathing is enabled and any NVMe devices you discover will use it.

For SLES 15, Native NVMe Multipathing is enabled by default and no additional configuration is required.

Access NVMe volumes for virtual device targets

You can configure the I/O directed to the device target based on which OS (and by extension multipathing method) you are using.

For RHEL 7 and SLES 12, I/O is directed to virtual device targets by the Linux host. DM-MP manages the physical paths underlying these virtual targets.

Virtual devices are I/O targets

Make sure you are running I/O only to the virtual devices created by DM-MP and not to the physical device paths. If you are running I/O to the physical paths, DM-MP cannot manage a failover event and the I/O fails.

You can access these block devices through the `dm` device or the `symlink` in /dev/mapper. For example:

```
/dev/dm-1
/dev/mapper/eui.00001bc7593b7f5f00a0980000af4462
```

Example output

The following example output from the `nvme list` command shows the host node name and its correlation with the namespace ID.

```
NODE         SN           MODEL            NAMESPACE

/dev/nvme1n1 021648023072 NetApp E-Series 10
/dev/nvme1n2 021648023072 NetApp E-Series 11
/dev/nvme1n3 021648023072 NetApp E-Series 12
/dev/nvme1n4 021648023072 NetApp E-Series 13
/dev/nvme2n1 021648023151 NetApp E-Series 10
/dev/nvme2n2 021648023151 NetApp E-Series 11
/dev/nvme2n3 021648023151 NetApp E-Series 12
/dev/nvme2n4 021648023151 NetApp E-Series 13
```

Column	Description
`Node`	The node name includes two parts: • The notation `nvme1` represents controller A and `nvme2` represents controller B. • The notation `n1`, `n2`, and so on represent the namespace identifier from the host perspective. These identifiers are repeated in the table, once for controller A and once for controller B.
`Namespace`	The Namespace column lists the namespace ID (NSID), which is the identifier from the storage array perspective.

In the following `multipath -ll` output, the optimized paths are shown with a `prio` value of 50, while the non-optimized paths are shown with a `prio` value of 10.

The Linux operating system routes I/O to the path group that is shown as `status=active`, while the path groups listed as `status=enabled` are available for failover.

```
eui.00001bc7593b7f500a0980000af4462 dm-0 NVME,NetApp E-Series
size=15G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| `- #:#:#:# nvme1n1 259:5 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  `- #:#:#:# nvme2n1 259:9  active ready running

eui.00001bc7593b7f5f00a0980000af4462 dm-0 NVME,NetApp E-Series
size=15G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=0 status=enabled
| `- #:#:#:# nvme1n1 259:5 failed faulty running
`-+- policy='service-time 0' prio=10 status=active
  `- #:#:#:# nvme2n1 259:9  active ready running
```

Line item	Description
`policy='service-time 0' prio=50 status=active`	This line and the following line show that `nvme1n1`, which is the namespace with an NSID of 10, is optimized on the path with a `prio` value of 50 and a `status` value of `active`. This namespace is owned by controller A.
`policy='service-time 0' prio=10 status=enabled`	This line shows the failover path for namespace 10, with a `prio` value of 10 and a `status` value of `enabled`. I/O is not being directed to the namespace on this path at the moment. This namespace is owned by controller B.
`policy='service-time 0' prio=0 status=enabled`	This example shows `multipath -ll`output from a different point in time, while controller A is rebooting. The path to namespace 10 is shown as `failed faulty running` with a `prio` value of 0 and a `status` value of `enabled`.
`policy='service-time 0' prio=10 status=active`	Note that the `active` path refers to `nvme2`, so the I/O is being directed on this path to controller B.

Access NVMe volumes for physical NVMe device targets

You can configure the I/O directed to the device target based on which OS (and by extension multipathing method) you are using.

For RHEL 8 and SLES 15, I/O is directed to the physical NVMe device targets by the Linux host. A native NVMe multipathing solution manages the physical paths underlying the single apparent physical device displayed by the host.

It is best practice to use the links in /dev/disk/by-id/ rather than /dev/nvme0n1. For example:

```
# ls /dev/disk/by-id/ -l lrwxrwxrwx 1 root root 13 Oct 18 15:14
nvme-eui.0000320f5cad32cf00a0980000af4112 -> ../../nvme0n1
```

Physical NVMe devices are I/O targets

Run I/O to the physical nvme device path. There should only be one of these devices present for each namespace using the following format:

```
/dev/nvme[subsys#]n[id#]
```

All paths are virtualized using the native multipathing solution underneath this device.

You can view your paths by running:

```
# nvme list-subsys
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:5700.600a098000a522500000000589aa8a6
\
+- nvme0 rdma traddr=192.4.21.131 trsvcid=4420 live
+- nvme1 rdma traddr=192.4.22.141 trsvcid=4420 live
```

If you specify a namespace device when using the 'nvme list-subsys' command, it provides additional information about the paths to that namespace:

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:5700.600a098000af44620000000058d5dd96
\
 +- nvme0 rdma traddr=192.168.130.101 trsvcid=4420 live non-optimized
 +- nvme1 rdma traddr=192.168.131.101 trsvcid=4420 live non-optimized
 +- nvme2 rdma traddr=192.168.130.102 trsvcid=4420 live optimized
 +- nvme3 rdma traddr=192.168.131.102 trsvcid=4420 live optimized
```

There are also hooks into the multipath commands to allow you to view your path information for native failover through them as well:

```
#multipath -ll
```

To view the path information, the following must be set in /etc/multipath.conf:

```
defaults {
        enable_foreign nvme
}
```

Example output:

```
eui.0000a0335c05d57a00a0980000a5229d [nvme]:nvme0n9 NVMe,Netapp E-
Series,08520001
size=4194304 features='n/a' hwhandler='ANA' wp=rw
|-+- policy='n/a' prio=50 status=optimized
| `- 0:0:1 nvme0c0n1 0:0 n/a optimized    live
`-+- policy='n/a' prio-10 status=non-optimized
`- 0:1:1 nvme0c1n1 0:0 n/a non-optimized    live
```

Create filesystems (RHEL 7 and SLES 12)

For RHEL 7 and SLES 12, you create a file system on the namespace and mount the filesystem.

Steps

1. Run the `multipath -ll` command to get a list of `/dev/mapper/dm` devices.

   ```
   # multipath -ll
   ```

   The result of this command shows two devices, `dm-19` and `dm-16`:

```
eui.00001ffe5a94ff8500a0980000af4444 dm-19 NVME,NetApp E-Series
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- #:#:#:# nvme0n19 259:19  active ready running
| `- #:#:#:# nvme1n19 259:115 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- #:#:#:# nvme2n19 259:51  active ready running
  `- #:#:#:# nvme3n19 259:83  active ready running
eui.00001fd25a94fef000a0980000af4444 dm-16 NVME,NetApp E-Series
size=16G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- #:#:#:# nvme0n16 259:16  active ready running
| `- #:#:#:# nvme1n16 259:112 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- #:#:#:# nvme2n16 259:48  active ready running
  `- #:#:#:# nvme3n16 259:80  active ready running
```

2. Create a file system on the partition for each `/dev/mapper/eui-` device.

   The method for creating a file system varies depending on the file system chosen. This example shows creating an `ext4` file system.

```
# mkfs.ext4 /dev/mapper/dm-19
mke2fs 1.42.11 (09-Jul-2014)
Creating filesystem with 2620928 4k blocks and 655360 inodes
Filesystem UUID: 97f987e9-47b8-47f7-b434-bf3ebbe826d0
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
```

3. Create a folder to mount the new device.

```
# mkdir /mnt/ext4
```

4. Mount the device.

```
# mount /dev/mapper/eui.00001ffe5a94ff8500a0980000af4444 /mnt/ext4
```

Create filesystems (RHEL 8 and SLES 15)

For RHEL 8 and SLES 15, you create a filesystem on the native nvme device and mount the filesystem.

Steps

1. Run the `multipath -ll` command to get a list of `/dev/nvme` devices.

```
# multipath -ll
```

   The result of this command shows device `nvme0n6`.

```
eui.000082dd5c05d39300a0980000a52225 [nvme]:nvme0n6 NVMe,NetApp E-
Series,08520000
size=4194304 features='n/a' hwhandler='ANA' wp=rw
|-+- policy='n/a' prio=50 status=optimized
| `- 0:0:1 nvme0c0n1 0:0 n/a optimized     live
|-+- policy='n/a' prio=50 status=optimized
| `- 0:1:1 nvme0c1n1 0:0 n/a optimized     live
|-+- policy='n/a' prio=10 status=non-optimized
| `- 0:2:1 nvme0c2n1 0:0 n/a non-optimized live
`-+- policy='n/a' prio=10 status=non-optimized
  `- 0:3:1 nvme0c3n1 0:0 n/a non-optimized live
```

2. Create a file system on the partition for each `/dev/nvme0n#` device.

   The method for creating a file system varies depending on the file system chosen. This example shows creating an `ext4` file system.

```
# mkfs.ext4 /dev/disk/by-id/nvme-eui.000082dd5c05d39300a0980000a52225
mke2fs 1.42.11 (22-Oct-2019)
Creating filesystem with 2620928 4k blocks and 655360 inodes
Filesystem UUID: 97f987e9-47b8-47f7-b434-bf3ebbe826d0
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
```

3. Create a folder to mount the new device.

```
# mkdir /mnt/ext4
```

4. Mount the device.

```
# mount /dev/disk/by-id/nvme-eui.000082dd5c05d39300a0980000a52225
/mnt/ext4
```

Verify storage access on the host

Before using the namespace, you verify that the host can write data to the namespace and read it back.

What you'll need

An initialized namespace that is formatted with a file system.

Steps

1. On the host, copy one or more files to the mount point of the disk.
2. Copy the files back to a different folder on the original disk.
3. Run the `diff` command to compare the copied files to the originals.

After you finish

Remove the file and folder that you copied.

Record your NVMe over IB configuration

You can generate and print a PDF of this page, and then use the following worksheet to record NVMe over InfiniBand storage configuration information. You need this information to perform provisioning tasks.

Host identifiers

The software initiator NQN is determined during the task.

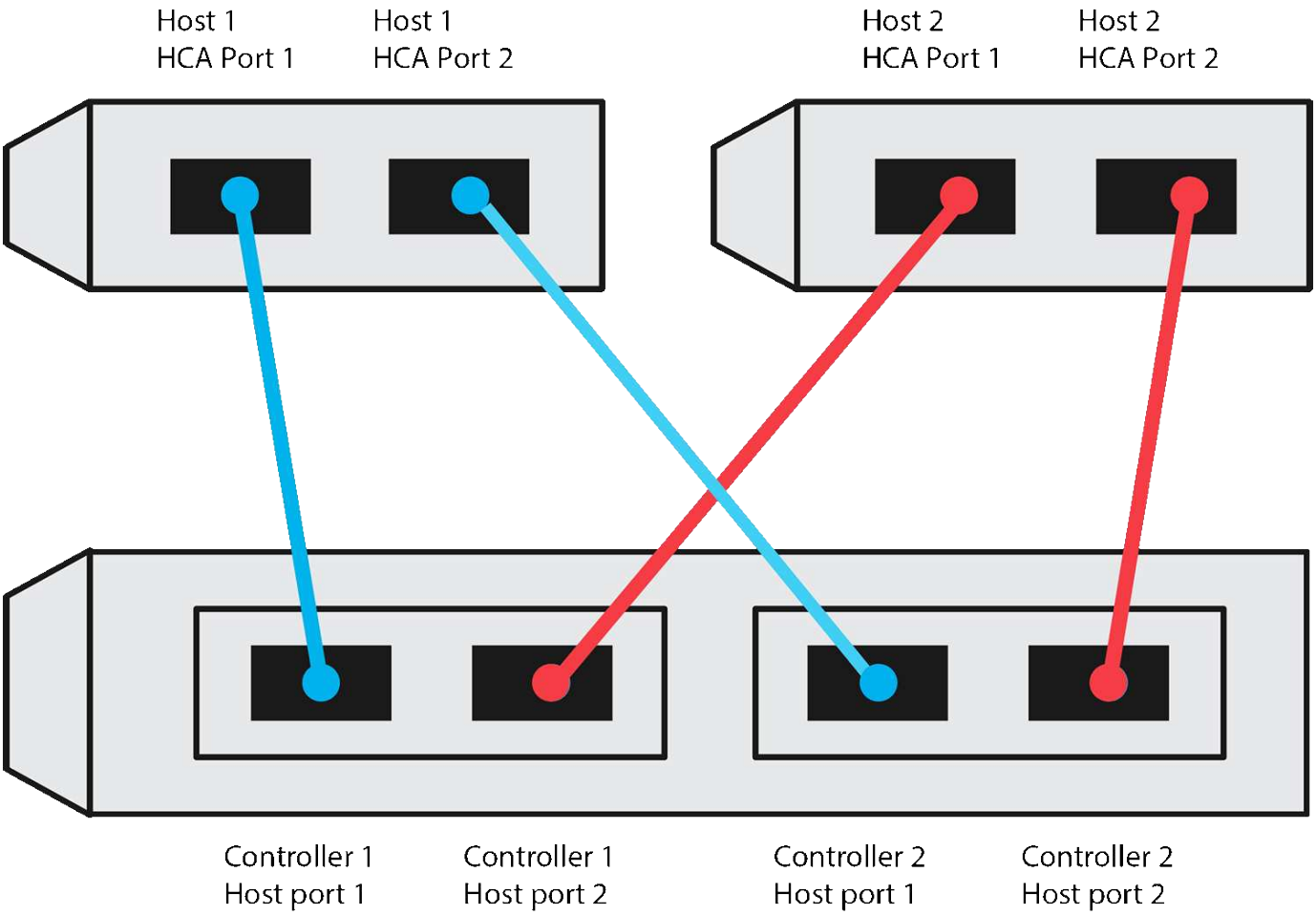
Locate and document the initiator NQN from each host. The NQN is typically found in the /etc/nvme/hostnqn file.

Callout No.	Host port connections	Host NQN
1	Host (initiator) 1	
n/a		
n/a		

Callout No.	Host port connections	Host NQN
n/a		
n/a		

Recommended configuration

In a direct connect topology, one or more hosts are directly connected to the subsystem. In the SANtricity OS 11.50 release, we support a single connection from each host to a subsystem controller, as shown below. In this configuration, one HCA (host channel adapter) port from each host should be on the same subnet as the E-Series controller port it is connected to, but on a different subnet from the other HCA port.



Target NQN

Document the target NQN for the storage array. You will use this information in Configure storage array NVMe over InfiniBand connections.

Find the Storage Array NQN name using SANtricity: **Storage Array** > **NVMe over Infiniband** > **Manage Settings**. This information might be necessary when you create NVMe over InfiniBand sessions from operating systems that do not support send targets discovery.

Callout No.	Array name	Target IQN
6	Array controller (target)	

Network configuration

Document the network configuration that will be used for the hosts and storage on the InfiniBand fabric. These instructions assume that two subnets will be used for full redundancy.

Your network administrator can provide the following information. You use this information in the topic, Configure storage array NVMe over InfiniBand connections.

Subnet A

Define the subnet to be used.

Network Address	Netmask

Document the NQNs to be used by the array ports and each host port.

Callout No.	Array controller (target) port connections	NQN
3	Switch	*not applicable*
5	Controller A, port 1	
4	Controller B, port 1	
2	Host 1, port 1	
	(Optional) Host 2, port 1	

Subnet B

Define the subnet to be used.

Network Address	Netmask

Document the IQNs to be used by the array ports and each host port.

Callout No.	Array controller (target) port connections	NQN
8	Switch	*not applicable*
10	Controller A, port 2	

Callout No.	Array controller (target) port connections	NQN
9	Controller B, port 2	
7	Host 1, port 2	
	(Optional) Host 2, port 2	

Mapping host name

The mapping host name is created during the workflow.

Mapping host name	
Host OS type	